Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Telecommunications and Media Informatics

Predictive modeling of specialized support capabilities with Machine Learning

MSc Thesis

Author
Anna Réka Lotz

Academic Supervisor
Dr. Gábor Szűcs

Industrial Supervisor
Juan Carlos Arriaga
Cloudera, Inc.

December 13, 2021

DEPARTMENT HEAD

MSc Thesis Task Description

Anna Réka Lotz
candidate for MSc degree in Business Informatics

Predictive modeling of specialized support capabilities with machine learning

The aim of the thesis is to develop a system based on machine learning to optimize the support capabilities of technology companies. At these companies, customers turn to customer service with any problems. However, we distinguish several groups of customers, during which the solution of problems may differ from the average. These special clients could receive more attention and quicker solutions to critical problems.

The aim of the system to be developed is to determine, in the light of all the information available about clients and their issues, in which cases an average client will belong to a specialized group. This is important because the number of people working in customer service and the budget are determined based on this. The goal is to create a system that can use machine learning to estimate when customers will enter a particular group.

Tasks to be performed by the student will include:

- Study the structure of a database containing the customer data of the company.
- Identify the features relevant to the solution of the task.
- Implement database queries to collect these features.
- Analyze whether the features correlate with the target variable.
- Design machine learning algorithms and use it to predict the complexity of customer complaints.
- Compare the models for the specialized support groups.

Supervisor at the department: Dr. Gábor Szűcs, associate professor
External supervisor: Juan Carlos Arriaga, Cloudera, Inc.

Budapest, 5 October 2021

Dr. Pál Varga
head of department

Dept. of Telecommunications and Media Informatics
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics

H-1117 Budapest, Magyar tudósok krt. 2.
Phone: +36 1 463 2401 – Web: http://www.tmit.bme.hu
E-mail: pvarga@tmit.bme.hu

Contents

HALLGATÓI NYILATKOZAT

Alulírott Lotz Anna Réka, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2021. december 13.

Lotz Anna Réka
hallgató

Kivonat

Az adatok manapság mindenütt jelen vannak. Ezek elemzése és összegyűjtése segíthet olyan gépi predikciók létrehozásában, amelyek könnyíthetik az üzleti életet.

Dolgozatomban a Cloudera multinacionális cég ügyfélszolgálatának elemzését és az ügyfelek kategorizálásának predikciós analízisét mutatom be. A cégnek számos ügyfele van, akik nap mint nap a Cloudera termékeket veszik igénybe és bármiféle felmerülő problémával az ügyfélszolgálathoz fordulnak. Az ügyfelek kétféle speciális csomagra fizethetnek elő, amennyiben az általános csomagban nem kapják meg a kellő figyelmet, vagy egyéb okból kifolyólag dedikált embereket szeretnének, akik dolgozzanak a problémáikon. Attól függően kell a dolgozó emberek létszámát megbecsülni, hogy melyik csomagra hány ügyfél fizet elő. Szükséges-e új embereket felvenni vagy inkább képezni kellene a dolgozókat? Ezek mind olyan kérdések, melyeket befolyásol az, hogy hány ügyfél fordulhat problémával a csapathoz. Ha tudnánk a választ, az ügyfél is elégedettebb, hiszen elegendő ember foglalkozik az ügyein.

A projekt a CRISP-DM módszertan alapján épült fel, mely során a probléma és adatok mélyebb megértése után következett a gyakorlati munka adatgyűjtéssel, adattisztítással, modellépítéssel majd az eredmények értékelésével.

A projekt során kétféle gépi tanulás modell létrehozásával az volt a cél, hogy meg tudjuk becsülni a modellel, melyek azok az ügyfelek akik az adataik alapján jogosultak lennének az ügyfélszolgálati csoportváltásra. Különféle adattal kapcsolatos problémákat kellett ehhez megoldani, mint a hiányzó adatokat, valamint a kiegyensúlyozatlan osztályokat.

A dolgozat bemutatja a különböző modelleket, amelyek összehasonlítása után lett kiválasztva a legjobban teljesítő gépi tanuló modell. Ennek eredményeit felhasználva készült egy vizualizáció, mely segítségével az ügyfélszolgálat vezetői megtervezik a jövőbeli szükséges kapacitást.

Abstract

Data is everywhere nowadays. Collecting and analyzing data can help us discover hidden patterns in the data set, making it possible to create predictions that can make business life more manageable.

In my dissertation, I present an analysis of customer service of the multinational company called Cloudera and the predictive analysis of the customer categories. The company has many customers who use its products on a daily basis. Customers can pay for two different packages if they are not satisfied with the basic package and require more attention or would like to have dedicated people working on their problems. Depending on the number of customers subscribing to which package, the number of people working should be estimated. Is it necessary to hire new people or should the company train employee skills instead? Replying to this question is affected by how many customers can turn to the team with a problem. If we knew the answer, the customer would also be more satisfied, as there are enough people to deal with their issues.

The project was built on the basis of the CRISP-DM methodology, where a deep understanding of the business reasons and data was followed by practical work with data collection, data cleaning, model building, and then the evaluation of the results.

By creating two types of machine learning models, the aim of this project was to be able to estimate which customers would be eligible for the two support groups based on their data. Various data problems had to be solved such as missing data and unbalanced data sets.

The dissertation presents different models and after comparing these, the best-performing machine learning model was selected. Using the results of this, a visualization was created to help support leaders plan for future required capacity.

Chapter 1

Introduction

1.1 Concept

In today's world, it has become vital to make the huge amount of data usable for business and to create programs that help leaders to understand the trends and make business decisions. It is becoming increasingly common to plan the future with the help of machine learning algorithms. The purpose of my project is similar, its main goal was to support cost-effective business planning.

Multinational companies, who supply software, have a lot of customers who use their products every day. In case the customer runs into any problems related to these products, the support team of the company addresses the issues. The complexity can vary dramatically from simple cases requiring just a couple of minutes to resolve to very complex ones stretching over several months to get resolved. If the support team can not solve the case, they escalate it to engineering for help. These complex cases can remain open for a potentially long period of time. These situations can be very expensive and frustrating both for the customer and the company. From the company's perspective, they would like to have happy and satisfied clients, who remain loyal to the company and generate revenue. The customers wish attention and quick responses from the support and they want to have a solution for their problem. They will be satisfied if they get those. The company should ensure that customers can connect with the right person at the right time.

When a customer often has high-priority cases, would like to have dedicated people to deal with their case, does not wish to discuss the problem always with someone else, would like someone to know their infrastructure better, perhaps has even sensitive data, then often subscribes to a higher-level customer service package. When this happens, a smaller team will deal with his incoming tickets in the future. The number of these support team members depends on how many customers have subscribed to the special support offering.

The Cloudera software company helped me by providing the necessary data and software as well as industry knowledge to develop this idea in a real business environment. Cloudera is an enterprise data cloud company that offers a software platform for data engineering, data warehousing, machine learning, and analytics [1].

In a previous project, I created a model to forecast incoming support issues from corporate clients and based on these numbers create a visualization interface to estimate staff numbers. Depending on how many problems come in per day, we can estimate how many people need to solve issues from customers. This makes it easier for managers to plan their headcount to cover the ticket volume.

Then I expanded this model with the volume for the different support groups. This way, the leaders were able to see the incoming tickets and the projected volumes for the current customers in the groups. I also implemented the model, not just total projections but individual customer predictions. Thus, selecting one or multiple customers and looking at their ticket volumes is available. When selecting only one client, the predicted values are not so accurate because of the limited numer of issues.

Hence, there is a visualization with a machine learning background delivered for the leadership where they can look at the projected incoming numbers for the next year and plan the required headcount based on the time zones and teams. The challenge is that they can not see the customer changes, and can not prepare for a bigger amount of conversion. The goal is to help to determine which clients can be offered the special support package and how the new client's volume will increase the overall volume. If the company would know a likelihood for the candidates who possibly can be the next client in one of the special groups then sales could offer a deal for the selected companies. This way, the contacted customers could have a more successful journey at the Cloudera because the prediction would based on the customers' ticket activity, cluster details, and other information. There would be a reason why the delivered model shows that the candidates should change the support package. Thus, the salesman would have real data behind the new offer.

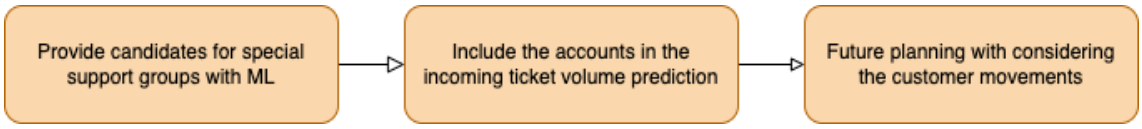


Figure 1.1: Relationship with the previous project

Figure 1.1 shows a flow chart with the phases of how this report is related to the previous work. This project's goal was to provide customer candidates for the specialized support groups. This includes data preparation, feature selection, and machine learning model building. The output of the models was a list of customer

names with a likelihood of whether they are eligible for the specialized support offering. The incoming case volume of these accounts was predicted and included in the visualization that allows the user to select the current specialized group's volume with additional candidates' volume.

During the project, I had to talk with multiple teams to understand how the process of changing support types works, what are the customer's motivation to extend the support package, what are the customer KPIs that the teams measure at Cloudera. I was working with ticket data earlier in the forecasting project but this was the first time that I had to be familiar with customer information and computer cluster details. Without a handy feature set, it takes a lot of time to gather all information, select the relevant features.

1.2 Project goals

The aim of the thesis was to develop a system based on machine learning to optimize the support capabilities of technology companies. At these companies, customers turn to customer service with any problems. However, we distinguish several groups of customers, where the solution of problems may differ from the average. These special clients could receive more attention and quicker solutions to critical problems.

The aim of the system to be developed was to determine all the information available about clients and their issues, in which cases an average client will belong to a specialized group. This is important because the number of people working in customer service and the budget are determined based on this. The goal is to create a system that can use machine learning to estimate when customers will enter a particular group.

1.3 Execution plan

Tasks to be performed will include:

- Understand the business reasons and processes
- Data Understanding
- Data Preparation, Feature selection
- Modeling and evaluating the results
- Deployment

At the beginning of the project, I studied the related literature, and I started to understand the business process and learnt how the company works. Thus, I had to learn how the support team works at Cloudera. In the Data Understanding phase, I have discovered the data that exists, and that could be important for the Machine Learning (ML) model. Then, I have analyzed, cleaned, and processed the data, created the feature set, and investigated the correlation between the features and the target variables. I created the final data set for the Machine Learning models. The next step was to find the suitable Machine Learning model for the prediction goals, to try several models, then compare them and select the best algorithm. After the Modeling phase, I evaluated the results and analyzed the evaluation metrics. Finally, in the deployment phase, I implemented a visualization that could help the managers in the future.

The report introduces Hadoop and Cloudera technologies including the used frameworks in Chapter 3. When a chapter presents the work using methods or techniques, the chapter starts with introducing these techniques with related literature and then describes the actual work.

1.4 Methodology

Data Mining is a part of Data Science that can help to analyze 'Big Data' and extract the relevant information. There are many types of Data Mining processes but the most popular ones are CRISP-DM, SEMMA and KDD. KDD stands for Knowledge Discovery Databases that refers to finding hidden knowledge in data and has nine steps. Sample, Explore, Modify, Model, and Access are the five stages of SEMMA as the name shows that is developed by SAS institute. SEMMA offers understanding, organization and maintenance of the data mining projects [2].

To carry out this project, the CRISP-DM methodology will be used, which stands for cross-industry process for data mining. This methodology provides an overview of the life cycle of a Data Mining project. It contains the phases of a project, their respective tasks, and the relationships between these tasks. It is known for being well-structured and described, as well as for its powerful practicality, its flexibility and its usefulness when using analytics to solve business problems.

The life cycle of a data mining project consists of six phases, shown in Figure 1.2. It is always necessary to move back and forth between the different phases. The result of each phase determines which phase, or particular task in a phase, should be performed next. The arrows indicate the most important and frequent dependencies between phases.

Figure 1.2: Cross-industry Standard Process for Data Mining (CRISP - DM)
Source: [3]

The outer circle of the figure symbolizes the cyclical nature of data mining itself. Data mining does not end once a solution is implemented. The lessons learnt during the process and from the implemented solution can trigger new, often more focused business questions. Subsequent data mining processes will benefit from the experiences of previous ones. The six phases of the CRISP-DM Process Model are described below [4]:

Business Understanding This initial phase focuses on understanding the project objectives and requirements from a business perspective, then turning this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives. The business understanding phase includes four main tasks: identifying business and data-mining goals, assessing the situation and producing the project plan.

Understanding the data The second phase begins with the initial data collection and becoming familiar with the data, identify quality problems and explore data to form a hypothesis for hidden information. The data-understanding phase includes four tasks: gathering, describing, exploring data and verifying data quality.

Data preparation This phase focuses on selection and preparation of the final data set. Data preparation tasks are likely to be performed multiple times and not in a prescribed order. Tasks include selecting tables, records, and attributes, as well as transforming and cleansing data for modeling tools. The data preparation phase includes five tasks: selecting, cleaning, constructing, integrating, and formatting data.

Modeling In this phase, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values. Typically, there are multiple techniques for the same type of data mining problem. Some techniques have specific requirements on the shape of the data. Therefore, it is often necessary to go back to the data preparation phase. The modeling phase includes four tasks: selecting modeling techniques, designing tests, building, and assessing models.

Evaluation At this stage of the project, a model (or models) has been built that appears to be of quality from the perspective of data analysis. Before proceeding with the final implementation of the model, it is important to thoroughly evaluate it and review the steps taken to create it, to ensure that the model achieves business objectives. A key goal is to determine if there are any major business issues that have not been sufficiently considered. At the end of this phase, a decision must be made about the use of the data mining results. The evaluation phase includes three tasks: evaluating the results, reviewing the process and determining next steps.

Deployment The final phase of CRISP-DM process focuses on determining the use of obtained knowledge and results. This phase also focuses on organizing, reporting and presenting the gained knowledge when needed. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process across the enterprise. In many cases, it is the customer, not the data analyst, who performs the implementation steps. However, even if the implementation effort will be carried out by the analyst, it is important for the customer to understand in advance what actions need to be taken to make actual use of the created models. The deployment phase includes four tasks: planning deployment, monitoring and maintenance, and reporting and reviewing final results.

At the end of the project, recommendations could be given on how the final model(s) can be implemented. However, this will be analyzed during the completion of this project. The deployment phase will be carried out during this project as the end-users are employees at Cloudera who will make decisions based on the results.

Chapter 2

Introduction to Cloudera Support

To understand the business it is really important to be familiar with the processes of support and how the support model looks like. In the following sections, I will introduce the support model at Cloudera, what are the support offerings and differences between them, and also introduce the headcount planner that I have been working on earlier. My current project will be merged into the projected case volume visualization from the previous project.

2.1 Support at Cloudera

Cloudera Support helps the customers to install, configure, optimize and run the environment for data processing and analysis. Technical experts are dedicated to resolving any technical issues that the user is facing [1].

Contrary to the support organization of other multinational companies, Cloudera support team does not distinguish the traditional four levels at Cloudera support team. The team consists of two levels: the frontline and the backline team. The members of both teams are engineers and are called Customer Operations Engineers. Based on their knowledge, they try to solve cases that are related to different components. Every frontline member knows the basics of 3-4 Hadoop components. If they can not solve an issue, they can escalate it to the backline team who knows much more about that software. Usually the members are good at one component but not every has a backline member. If the situation is more complicated or there is no backline member for that component then the support team escalates the case to the development team of the given software. When they identify and fix the problem, the Customer Operations Engineer can close the case. The solution of the problem could take several weeks because it is not easy to find the source of a problem, reading the log files could require a lot of time, and the communication is not always fluent.

2.1.1 Cloudera Support Offering

The customers can have three main types of support: the general, Premier and US Government Support. Cloudera Support pairs the team of experts with proactive and predictive support capabilities to enable the experience of more uptime, faster issue resolution and better performance.

Cloudera distinguishes customers whose cases are handled differently by other employees. The support offering are called Premier Support and US Secure Support.

Cloudera Premier Support goes beyond business-critical support by delivering enhanced services from a team of dedicated experts to ensure the clients' organization has optimal support to meet their business needs. The dedicated team of expert engineers provide proactive, high-value services based on a deep understanding of their business and Cloudera deployment. It is valuable for customers who create many tickets, need a solution quickly, and value the dedicated team.

When the cases are handled by the same 2-3 members of the team, the engineers get to know the specific business environment, infrastructure, and they do not need to ask the general questions over and over again. The familiarity with the customer's business help them to save a lot of time and they can jump into the current issue with knowing all the small details about the computer clusters, components, and settings.

All customers have access to a team of dedicated experts 24 hours a day 5 days a week based on their local business week. The support engineers are focused on supporting rapid case resolution, delivering proactive, high-value services based on their deep understanding of the customer's business, data driven workloads and deployment. The customers can be located anywhere worldwide, have 24x5 support and Cloudera should provide dedicated people to cover each time zone in their working hours. That is sometimes a challenge for the company especially during holidays when the managers should take into account the employees' national holidays.

The other support option is Cloudera Government Support for the US Public Sector customers. Cloudera offers three levels of support to give customers exactly what they need, when they need it. The most comprehensive support offering includes up to 50 on-site visits per year to ensure the customer's project is successful. All Government Support is US Based led by cleared US Citizens. The advantages are more uptime, faster issue resolution and better performance for the mission-critical applications.

The provided support is flexible: 24x7 hours of operation for the Severity 1 cases and 8x5 hours of operation for the lower priorities. The team members have a rich knowledge base on core technical topics, support for workflows and escalation process, ongoing health checks and air gap tooling for sensitive data.

The customer's region is Public Sector but not all of these customers have the Government Support. This support team is much smaller than the Premier team and has less customers.

Business Structures

To understand the data from customer side, it is important to be familiar with the architecture of the customer's business. In the Agile Organization structures two basic business operation categories are distinguished: lines of business and shared services. They manage the service units that deliver the enterprise products and services [5].

Line of Business (LOB) A general term that refers to a product or a set of related products that serve a particular customer transaction or business need. An enterprise will have one or more LOBs. An LOB delivers products or services to customers. The LOB will manage the top-level collaboration(s) that drive the life-cycle stages and directly or indirectly engage shared services to do the detailed work.

Shared Services Each shared service unit must report to an organization unit that is organizationally higher than all organization units that are recipients of its services. In the agile enterprise, most of the actual work of the LOBs should be delegated to shared services. Shared services are for example: finance and accounting, and human resource management.

At Cloudera, the line of businesses are maintained as individual accounts with unique IDs and they are linked to a 'parent' which is the logo account. For example, we can have accounts named *BME*, *BME-VIK*, *BME-GTK* and *BME-GPK*. BME is a university and *VIK*, *GTK* and *GPK* are one of the faculties of the university. Table 2.1 shows this example data set where *BME* would be the logo account and the faculties would be the line of businesses with unique IDs. Both type of accounts can create tickets and they have the same attributes in each data table.

How are the accounts moved?

At the beginning of the project, the main question was how the accounts are moved to the specific support groups. Are all the LOBs moved to Premier or US Secure Support? Or will the logo account pay for the dedicated experts?

With some exceptions (only three customers that have been with the company for long time), the conclusion was that only the LOBs that need the special type of support will be moved but not the logo accounts. It makes sense that if the customer

Table 2.1: Example Table for LOBs and Logos

Account Name	Account ID	Logo Account Name	Logo Account ID
BME	1		
BME-VIK	2	BME	1
BME-GTK	3	BME	1
BME-GPK	4	BME	1

has 5 different accounts that can create tickets but only one has high priority issues or only one has sensitive information then why to pay the extended support for each account.

The extended support package can be requested by the customer itself or a salesman can offer a deal for them. After reaching baselines in certain attributes (such as recording the amount spent at Cloudera or cluster details), the sales team will contact customers for a possible extension.

Currently, without a machine learning model and having knowledge of the likelihood of candidates, the team has some attributes to look at and make a proposal based on them. For Premier Support, they investigate the amount of money spent, the high technical ticket activities, and larger cluster footprints. For the Government Support, they look at the region to be 'Public Sector' but not the education-related industries.

2.1.2 Headcount Planning

All the created tickets are related to one or many Hadoop components. There are more than 30 components supported by the company. The components are distributed to four main groups called pillars based on their function (store, process, etc.). The support team members are working in these pillars and everyone knows 8-10 components, of which 2-3 components in more depth. Cloudera is a multinational company which means that the customers are from all over the world and they need help in any time zone. Therefore, the support team works in several shifts in 6 time zones. There are fewer team members in shift 1 than in shift 3 and fewer customers in that time zone. For headcount planning, it would be important to know how many support requests will come from customers in which pillar and shift.

A model was created to forecast incoming support issues from corporate clients and based on these numbers create a visualization interface to estimate staff numbers. Depending on how many problems come in per day, we can estimate how

many people need to solve issues from customers. This makes it easier for managers to plan their headcount to cover the ticket volume. In Figure 2.1, the orange line shows the actual while the blue shows the projected case volumes for each week. The predictions were broken down by pillars and shifts. Facebook's open-source Prophet tool was used as the time series forecasting model. It takes into account the seasonality and holidays which is why we can see the low peaks in the figure around Christmas.

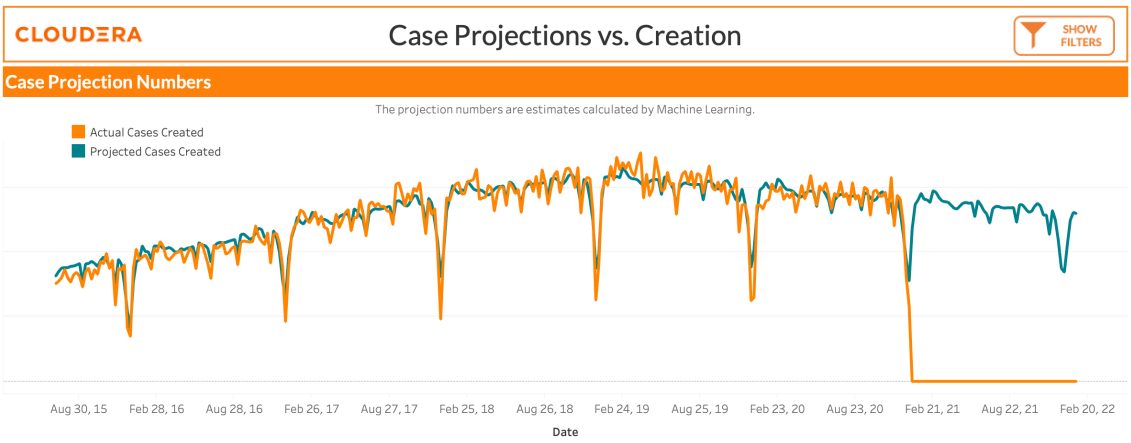


Figure 2.1: Forecasting Case Volume

Then this model was expanded with the volume for the different support groups. This way, the leaders were able to see the incoming tickets and the projected volumes for the current customers in the groups. The model was also implemented with not just total projections but individual customer predictions. Thus, selecting one or multiple customers and looking at their ticket volumes is available. When selecting only one client, the predicted values are not so accurate because of the limited data. The total volumes for each client in the Government Support can be seen in Figure 2.2. Comparing this figure to the 2.1, it shows that the support groups have significantly fewer tickets. The spikes are bigger while the first figure was smoother. The Premier team has a similar line graph with their case volume. Both are able to filter for an individual LOB customer name who is currently flagged with extended support.

2.2 Business Goals

The main goal of this project was to help the leaders with future planning by providing the customer names from the machine learning model output. With the candidates for extended support, the possible incoming ticket volume could be projected.

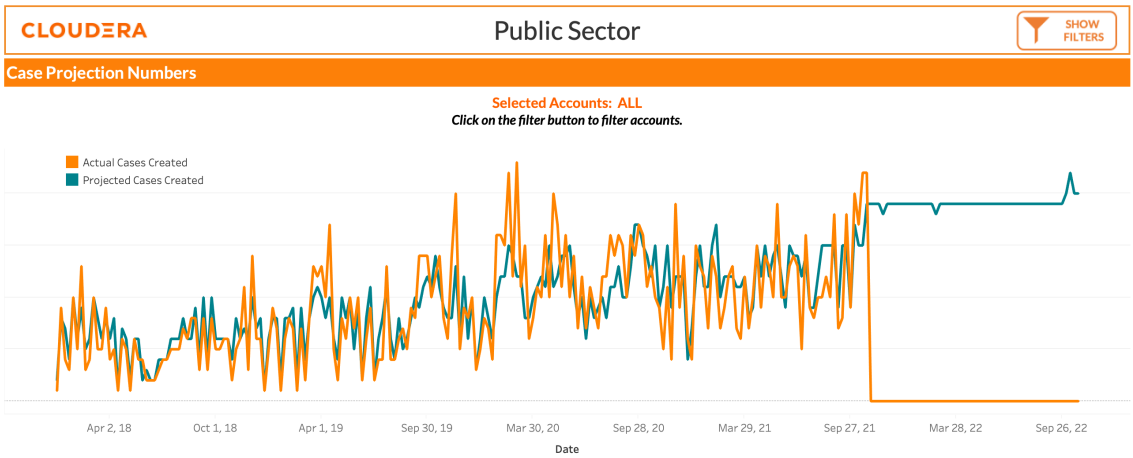


Figure 2.2: Forecasting Support Groups' Case Volume

The total case volumes are visible in Figure 2.2 and there is an additional filtering included for LOB customer names. With the predicted results of the model, the user would be able to select client names that are outside of extended support, and then compare the volumes with or without including the new clients.

For this, it would be great to know which customer accounts are eligible for extended support and what is the likelihood of their movement. The leadership would be able to plan required headcount more accurately with this feature. If they need to hire new people, they have the time for onboarding because they can calculate with the new customers in the group.

By knowing the candidates' likelihood the sales team's work would be easier and they could convince customers more easily based on their activity history. If the client is not satisfied with Cloudera Support it could be for example because of lack of attention, or too many people working on their cases. Obviously, it could be because of other reasons but the previously mentioned issues can be solved by extending the support package. This way the company would be able to reduce customer churn and they would have more happy, satisfied customers.

Chapter 3

Technical Background

3.1 Apache Hadoop

Apache Hadoop is an open source framework that is used to efficiently store and process large data sets ranging in size from gigabytes to petabytes of data. Instead of using one large computer to store and process the data, Hadoop allows clustering multiple computers to analyze massive data sets in parallel more quickly.

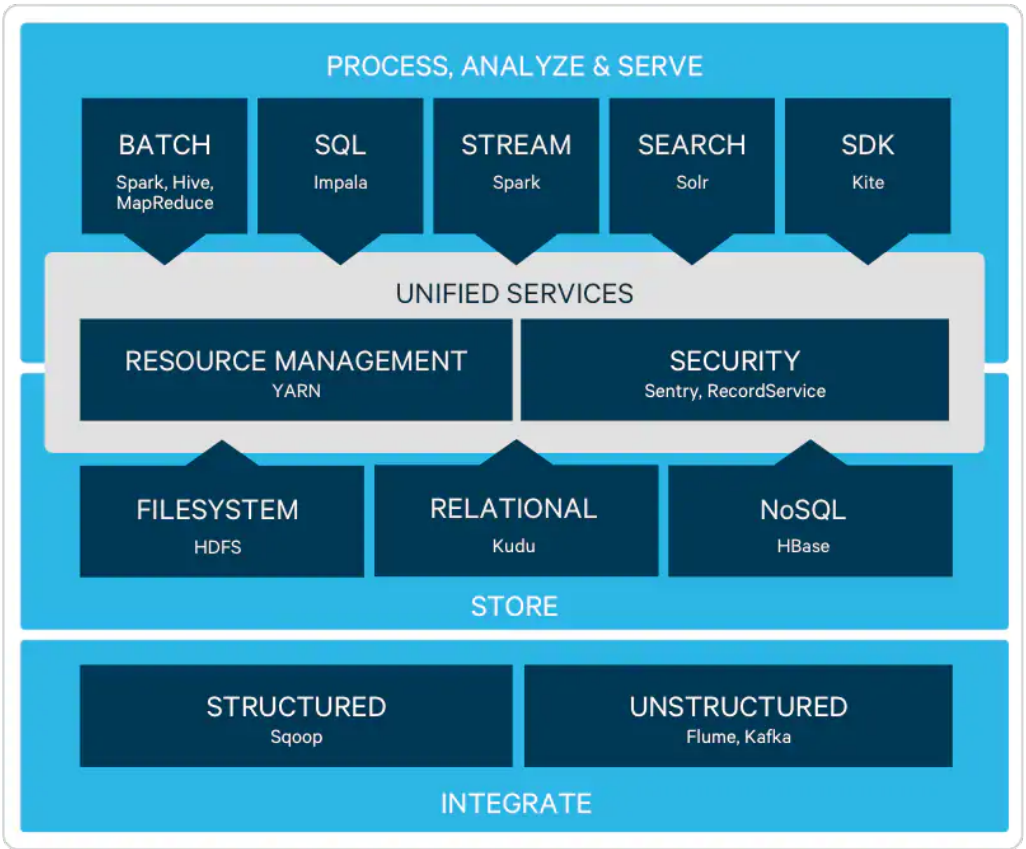


Figure 3.1: Apache Hadoop Ecosystem
Source: [1]

The Hadoop ecosystem, showed in Figure 3.1, describes the use cases and the components of the framework: integrate, store, process, analyze, and serve.

3.1.1 Cloudera and Hortonworks

There are many companies who provide Hadoop distributions and help to manage the Hadoop components. The most famous ones include Cloudera and Hortonworks. Hortonworks was founded in 2011 while Cloudera three years earlier in 2008. Both companies built their business on the open source thus freely available, Apache-licensed Hadoop framework and could rightly expect success as the open source system became a big data platform. However, due to the complexity of the system there was a need for companies to support corporate usage.

Cloudera created its own Hadoop distribution (CDH) and Cloudera Manager software. They sell the Manager software and also provide support and consulting services. In Figure 3.2, the Hadoop ecosystem is visible with the additional Cloudera products that can help with data management (Cloudera Navigator Encrypt) and operations (Cloudera Manager and Cloudera Director).

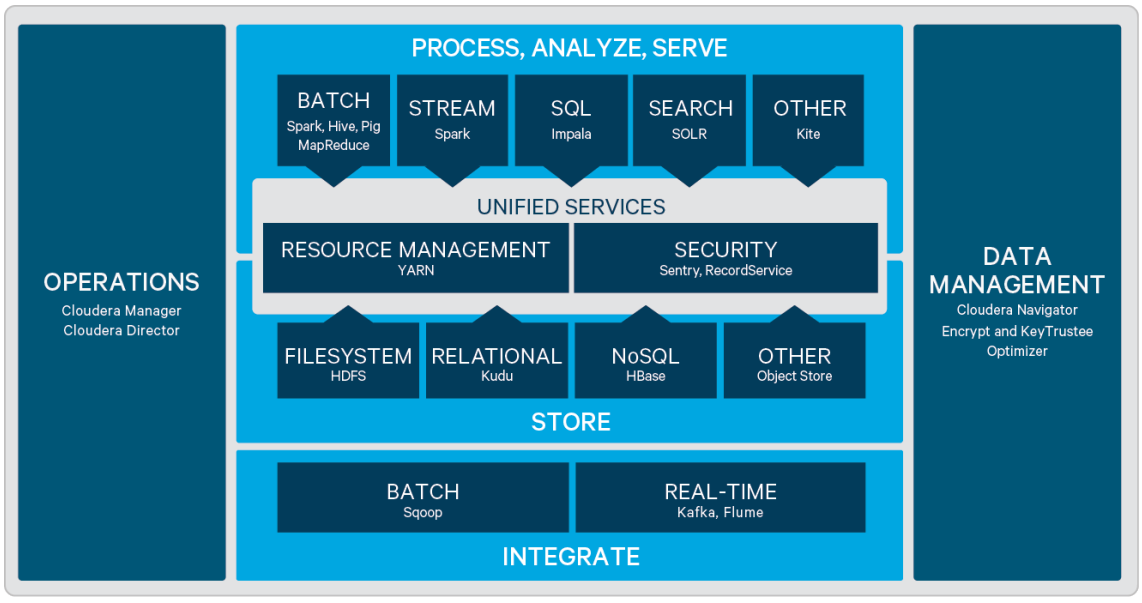


Figure 3.2: Cloudera Enterprise with Hadoop ecosystem
Source: [1]

Hortonworks' main product was the open source Hortonworks Data Platform (HDP) which can be used to store and analyze large amounts of data. They offered installation, configuration help, and expertise [6].

The two companies officially merged at the beginning of the year 2019. Both companies' approach are a little bit different. Hortonworks is famous for the purely open source software, where money is primarily in support, while Cloudera also has

paid-licensed products. The two portfolios complement each other well so they can come up with a much more comprehensive offering. Hortonworks brings end-to-end data management solutions, while Cloudera brings improvements to data storage as well as machine learning.

Cloudera Data Platform

Cloudera's two legacy Hadoop distributions are the Cloudera Distribution of Hadoop (CDH) and the Hortonworks Data Platform (HDP). After the merger, the first product was announced Cloudera Data Platform (CDP) that differs in big ways from those on-premise-oriented platforms, including the elimination of YARN in favor of Kubernetes for container management and a replacement of HDFS for public cloud object stores, including Amazon S3.

CDP is a big data platform for both IT and the business, Cloudera Data Platform (CDP) is:

- Simple to use and secure by design
- Manual and automated
- Open and extensible
- For data engineers and data scientists
- On premises and public cloud

Cloudera Data Platform provides various form factors: Public and Private Cloud. The newest Hybrid Cloud delivers all values of Private and Public Cloud. The difference between them is explained in the next section after defining what cloud computing is.

3.1.2 Cloud computing

Cloud computing is a branch of computing. We can distinguish several types of cloud-based services, the common feature is that the services are not operated on a specific hardware device but distributed on the service provider's devices hiding its operational details from the user. Services can be accessed by users over a network, for the public cloud over the Internet, for a private cloud over the local network or over the Internet. Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services [7].

There are three types of cloud computing service models:

Infrastructure as a Service (IaaS) The consumer has a control over storage, networking components, processing, accessing and monitoring computers and is able to develop applications.

Platform as a Service (PaaS) The cloud environment is hosted by the service provider and they deliver a framework for developers with cloud components. Compared to SaaS, applications could be created and modified in the PaaS model.

Software as a Service (SaaS) The consumer uses the provider's application that runs in a cloud environment (directly through the web browser). They are not able to manage the underlying cloud infrastructure.

The cloud infrastructure under cloud computing could be various. The models are not significantly different from each other as they use the same technology. Private cloud is expensive but secured in contrast to the public cloud. Hybrid cloud combines both so it could be a compromise for many companies.

Public Cloud The cloud infrastructure is publicly available and the users connect to the internet to reach their resources. They do not need to buy hardware, they pay based on usage to the cloud provider. The advantage of the public cloud is scalability, inexpensive, easy-to-use, but the disadvantage is that it is not the most secure. Google, Amazon, and Microsoft are some of the biggest public cloud providers in the world.

Private Cloud The cloud is hosted within the organization and only the employees can access the resources and data. This way the infrastructure's advantage is security, exclusivity, and privacy but the drawback is the cost.

Hybrid Cloud The infrastructure is a composition of two or more clouds (public and private). This way the consumers can have a secured, private cloud with confidential data and a cheaper, easier scalable public cloud for other computations.

3.2 Apache Impala

Apache Impala is an open source component of Hadoop that is able to process, analyze the data whether it is stored in HDFS, Apache HBase, or the Amazon S3. Impala is similar to Apache Hive as they share user interface, SQL syntax, ODBC driver, and metadata. With Hive, the user can run batch processing workloads while Impala provides fast, interactive SQL queries in real-time. Impala is able

to query big data and read widely-used file formats (e.g. Parquet, Avro, RCFile). It runs as a distributed service on the same machines as the other parts of the Hadoop infrastructure. One Impala daemon process is located in each node that is responsible for all aspects of query execution. The user interface where we can query data is Hue [8] [9].

Hue

Hue is a web-based interactive query editor that enables you to interact with data warehouses. Hue allows the user to browse the databases, explore the tables, import custom data. Two main editors (Hive and Impala) are available for querying but the consumer can also schedule repetitive jobs or create dashboards.

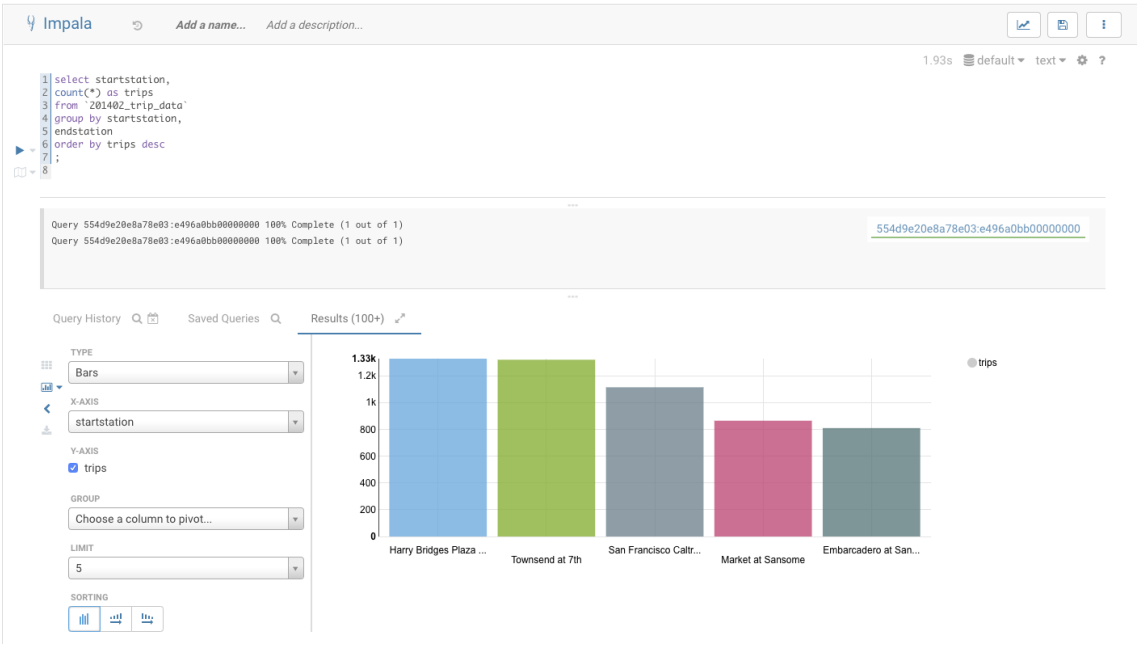


Figure 3.3: HUE user interface
Source: [1]

In Figure 3.3, an Impala query is shown with a chart visualization instead of a table result view. A basic graphic representation is included with bar, line or pie charts and with selectable axis. The table view results can be saved in CSV, Excel formats, copied to clipboard or exported to HDFS.

In Hue, I was able to query all the data that was needed for having enough information about the customer's behaviour. It provides access to every data source in the company, including accounts and support ticket details historically.

3.3 Data Science Workbench

The Cloudera Data Science Workbench (CDSW) is an enterprise data science platform built for machine learning projects. It allows the user to create projects in three different languages: R, Python, or Scala and run computations in Hadoop clusters. The data scientists can manage their analytics pipelines, including built-in scheduling, monitoring, and email alerting. The CDSW provides self-service access to data and also secure access to Apache Spark and Impala. It can handle every part of the data science with data collection, analysis, model building and visualization of the results. This collaborative platform has terminal access that makes it really easy to be up-to-date with the Github master branch. The user can schedule jobs with parameters and the workflow can run both in the public cloud and on-premises.

CDSW extends an existing CDH cluster, by running on gateway nodes and pushing distributed compute workloads to the cluster. CDSW requires and supports a single CDH cluster for its distributed compute. Data usually resides on the attached cluster. Two types of security are available the Kerberos authentication integrated via the cluster and external authentication via LDAP/SAML. The project files, internal postgresDB, and Livelog, are all stored persistently on the Master host [10].

The user interface in Figure 3.4 is directly on the web browser.

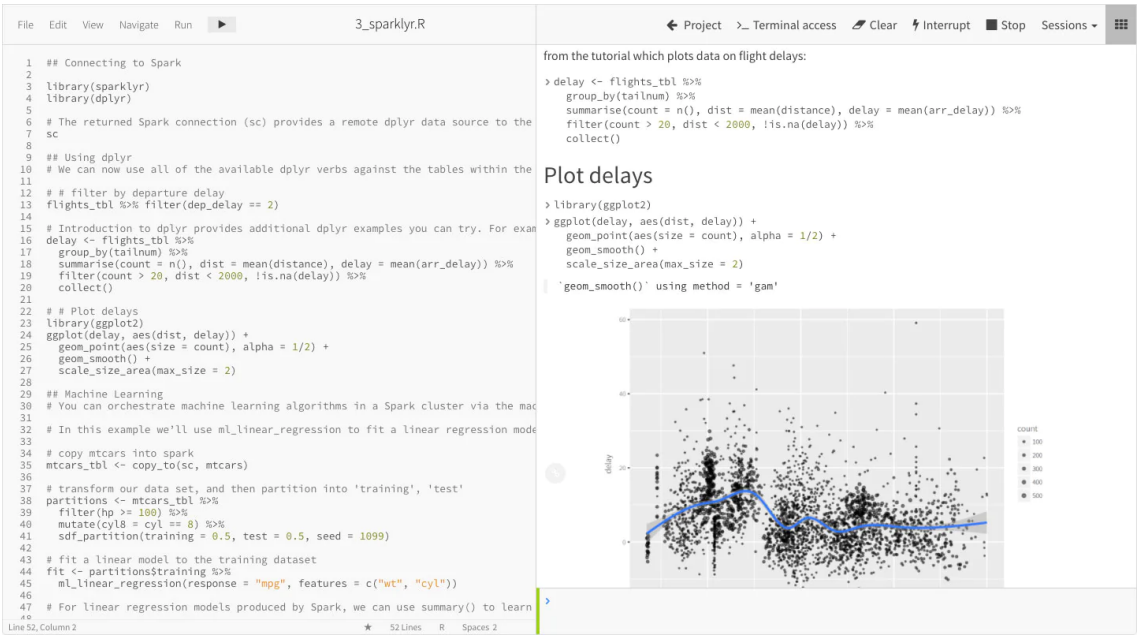


Figure 3.4: CDSW user interface
Source: [1]

For this project, the codes were written in Python and imported libraries were used for Machine Learning tools. I often made backups of my work and I pushed the changes to Github. Using CDSW was very convenient because the successful run

of some models took a long time and the programs were running on clusters which means that the work was not dependent on whether my laptop is falling asleep or running out of battery.

Python libraries

Running the CDSW with a Python 3 session enables the user to import any kind of library. The most important ones for this project are listed below.

impala.dbapi

The Impala DB API client can help connect the CDSW to Impala. It has a *connect()* function which has many parameters but I only used three of them with the following values: *host*= - defined the hostname, *use_ssl*=true - enabled SSL, *auth_mechanism*='GSSAPI' - for Kerberos authentication mechanism.

```
conn = connect(host = <hostname>, use_ssl = True,
    auth_mechanism = 'GSSAPI')
```

After connecting to Impala, I was able to select any tables in the HDFS filesystem. There are two options to read the query with the connection details. The first and shorter option is using Pandas *read_sql()* function that will result in a Pandas DataFrame:

```
results = pd.read_sql('''<query_details>''', conn)
```

The second option is using the Cursor object and its *execute()* function but to have the results in DataFrame, the *as_pandas()* function is necessary:

```
with conn.cursor() as cursor:
  cursor.execute( '''<query_details>''' )
  results=as_pandas(cursor)
```

numpy

Numpy is a module that support fast array operations.

pandas

Pandas is a module implementing DataFrame which is a two-dimensional tabular data structure. Pandas DataFrame consists of three principal components, the data itself, rows, and columns. All data coming from the company database were stored in DataFrames and I was able to easily format, and then clean the data in that format.

sklearn

Scikit-learn (sklearn in short) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and modeling including classification, regression, and clustering.

imblearn

Imbalanced-learn (imblearn in short) is a library for dealing with imbalanced data sets in Python. It provides tools for oversampling and undersampling the data.

matplotlib

Matplotlib has rich tools to visualize DataFrames as graphs including scatter plots, line plots, and bar plots.

3.4 Tableau

Tableau offers a highly interactive and intuitive visual-based data exploration experience for business users providing easy access, preparation, and analysis without having to write a single line of code. This experience is achieved through their 3 main products: an 'editing' tool and two sharing / collaboration platforms are available:

Tableau Desktop is an editing tool where you can connect to data and create views and dashboards to share with others.

Tableau Server is an on-premises solution that can be deployed through local hardware or a cloud-based provider.

Tableau Online is a 100% cloud-based platform which makes it easier to publish and share dashboards.

Many companies are launching software that creates beautiful dashboards compared to Tableau that has a strength that it also offers analytic - the details it makes stand out from other products. Its main goal is visual data analysis and the production of results that are easy to use and understand, flexible, and most importantly, fast. You can connect to almost any data source - plenty of built-in data connectors can be used both in memory and by a direct query for larger data sets [11].

I used Tableau Desktop for data exploration and also Tableau Servers for sharing data instead of creating extract locally. This tool is a space for visual data analysis. Most steps can be performed by drag & drop while continuously seeing the results of what has been done.

Chapter 4

Data Understanding

The data-understanding phase includes four tasks based on the CRISP-DM methodology: gathering, describing, exploring, and verifying data. In the first task, I started to gather all the available information and think about what could be relevant to this project. First, I had meetings with the leaders of the support teams to have a better understanding of the accessible data. They recommended some fields for consideration such as money spent at the company, and the size of the clusters. Then I tried to categorize the information into three main groups: customer information, computer cluster details, and support activity.

The mind map in Figure 4.1 shows the ideas for the possible features with the categories. For the support activity, the first few ideas were related to tickets and their details (owner, days to solve) but then I added the comment analysis (avg. comment per ticket or customer) and other Support KPI metrics that could be relevant. The second category is computer cluster information that includes technical data, the number of clusters and nodes, and cluster types. Thirdly, customer information covers the size of customer, time since client, sales area, industry and any other knowledge that the company stores about them.

4.1 Data Structure

Luckily, there are many data tables available to use and the issue was mostly to gather them together from the different sources. As visible in the mind map, I categorized the possible features into three main groups but there were much more underlying tables.

Cloudera uses a Customer Relationship Management (CRM) platform that manages interactions with customers and potential customers. This CRM system helps to build client relationships and streamline processes so they can increase sales, improve customer service, and increase profitability [12]. This platform is connected

Figure 4.1: Feature Ideas mind map

with the internal data warehouse thus I could easily query and explore the tables in Hue.

The main tables that were used can be seen in Figure 4.2 including the cases, comments, escalations, clusters, and the two account tables. This diagram is simplified as during gathering the cluster information I used four different tables. The account and cluster tables are connected based on the unique account_ids and the records in the comments and escalations tables belong to only one record in the case table.

I implemented database queries to collect the features. I wrote Impala queries to connect the tables and extract the information. I created Impala views in the data warehouse that I could read from CDSW with the *impala.dbapi* library in Python. In the queries, I used *WITH* statements and window functions (analytic functions) that took a long time to execute which is the reason why I decided to split the queries into multiple parts.

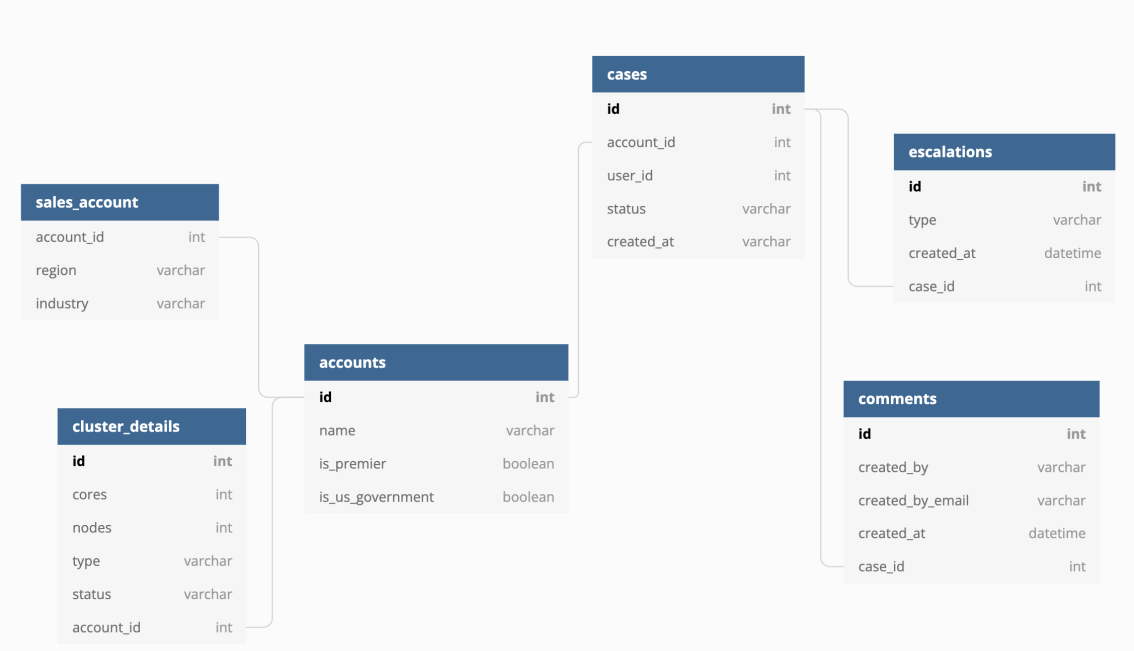


Figure 4.2: Database Relationship Diagram

Support Activity specific details

The support activities are stored in multiple data tables. The details related to tickets such as id, created, solved and closed dates, status, account id, severity, product and component can be found in the cases table. The cases table includes one record for each ticket and one case can have many comments and escalations. There is a creator email field in the comments table and the domain part shows to which company the people belong who wrote the note. If the end of the email address looks like '@cloudera.com' then they are employees otherwise clients. The escalation table has a type field that determines which kind of escalations we are talking about.

Customer specific details

The company keeps customer information in multiple tables. The most important is the one coming from the CRM system with the attributes stored in the platform along with a historical table. Each row represents one client in the account table and has a pair in the *sales_ account* table. The sales table is more accurate because that is used for customer communication and when they need any specific details the sales team search in that table.

Cluster specific details

Regarding the clusters, the number of nodes and cores are the most relevant knowledge. The type of the clusters can be various based on products and components. The data warehouse contains multiple cluster related tables. I used the *current_assets* that stores the legacy cluster informations, the *cdsw_clusters*, and *cdp_clusters*. It is valuable to know whether the customer is upgrading from the legacy products (CDH, HDP) to the new CDP or still using the old platforms.

4.2 Describing Data

I would like to introduce the available data and present some visualizations. In most of the figures, the blue trend line shows the attributes right before the customer moved to either Premier or Government Support.

4.2.1 Numbers of Cases

Creating cases is the most important activity in support and many metrics are related to tickets. The number of created cases, closed cases, the time until a solution or closure, close rate - these are all possible features in a period of time with a right measure such as average, count. An example can be seen in Figure 4.3 with an increasing number and huge peak in the number of created cases. One month later this customer moved to Premier. It is not as easily visible with other clients but eventually, the model will decide which feature is relevant.

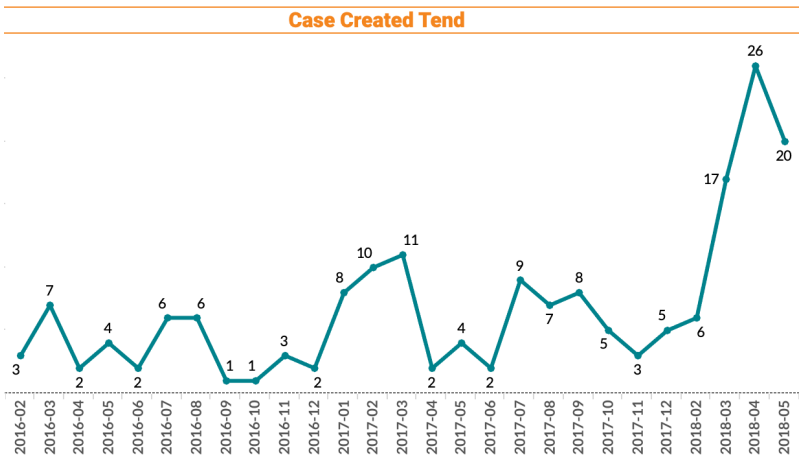


Figure 4.3: Created Cases Trend - Example

Each case possesses a unique id and a number. The case number consists of six numbers while the id is a string with random characters. Either one of them is included in each ticket-related data table.

The ticket life cycle has different timestamps: when was the case created, solved, and closed. A timestamp is visible when there is a handover between employees. The initial feature list contains the average days open and avg time until the solution in days. Besides the timestamps, each case is opened in a product that could be also relevant so I added to the list the different types and the case numbers in those products.

4.2.2 Ticket Severity

Based on the priority, the team has different rules for handling the issues. With severe cases comes more attention from the support team as they want to solve the issue as soon as possible and get the production environment running again. The severity of the tickets has four levels:

Level 1 This is the highest priority when there is a major error in the product that severely impacts the customer's use for production purposes such as loss of data, production system is down or severely impacted such that routine operation is impossible.

Level 2 The second highest priority shows that the system is working but in a limited capacity. This includes a problem that is causing significant impact to portions of the customer's business operations and productivity, or where the software product is exposed to potential loss or interruption of service.

Level 3 A medium-to-low impact error that involves partial and/or non-critical loss of functionality for production and/or development purposes, such as a problem that impairs some operations but allows the customer's operations to continue to function.

Level 4 An S4 case is a low priority request for information or feature request where there is no impact to customer's business operations.

In Tableau, I was looking at the number of created Level 1 and 2 tickets but for the model, I included all ticket severities. In the Created S1 Cases trend line graph in Figure 4.4, the difference between the peak and the average is not as significant as in the case trend example but having nine highest priority cases means that there were some serious issues going on at the client's side.

There are three types of severity in the case-related tables. The initial severity shows the priority that has been set during creating the ticket mostly by the customer, the highest severity is looking at the highest priority during the life cycle of

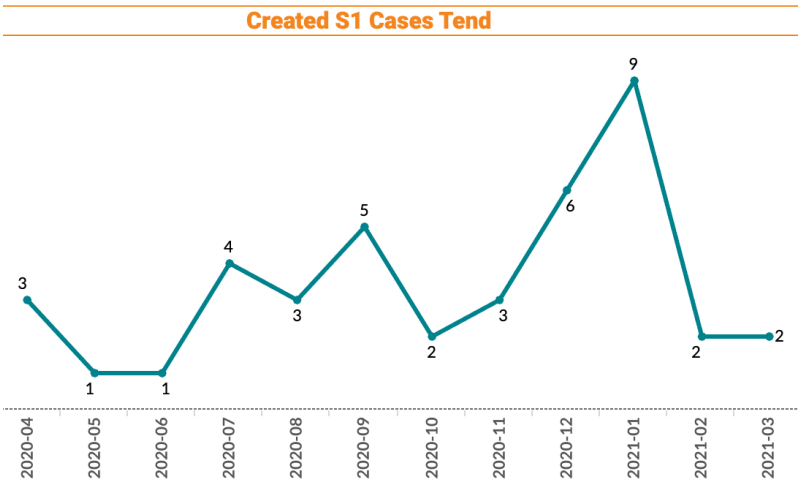


Figure 4.4: Created S1 Cases Trend - Example

the case and the third type is the currently active severity. When a Level 1 Priority case is handled and after the severe damage is solved the owner decreases the priority.

4.2.3 Case Owner

The leaders of the support groups told me based on their experience that the number of different ticket owners often has an impact on the support offering change. During gathering data, I tried to understand what could really have an influence so I visualized the features in Tableau.

In Figure 4.5 and 4.6 are two examples that shows the number of different ticket owners every month for two example customers who are currently in Premier Support. In the second example these numbers increase dramatically from 1 in each month to 11 and 10 monthly.

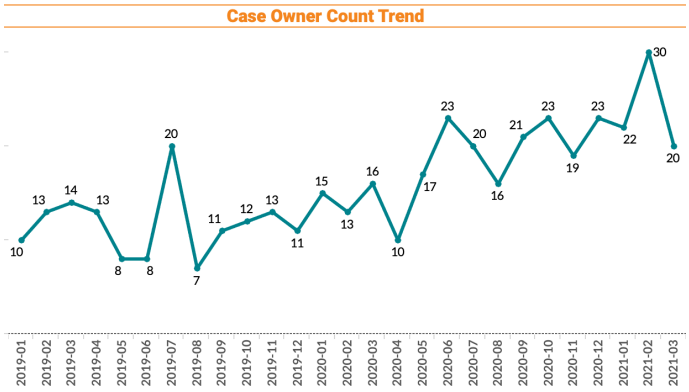


Figure 4.5: Number of Case Owners Trend - Example 1

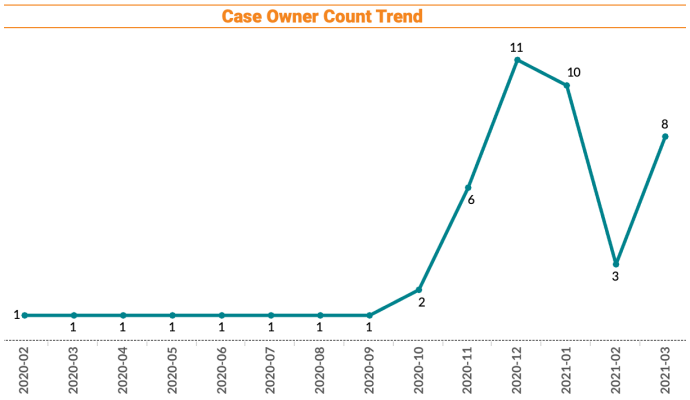


Figure 4.6: Number of Case Owners Trend - Example 2

In addition to the number of different case handlers, I also included the skills of these members in the components of the tickets. There are four levels: Beginner, Intermediate, Advanced, and the highest is SME (Subject Matter Expert). If the case is solved by an SME then probably the expert has more knowledge in that component and the solution will be delivered sooner. This can result in having more satisfied and happy customers.

4.2.4 Case Comments

Besides the activity in creating and closing the cases, it would be understandable to investigate the comments from both sides. Each people has a style for communication and it has an effect on the ticket life cycle. From the Cloudera side, the support members write comments on the cases as an update or ask the customer for information. It can happen that there is a handover between the team members for example when it is a high priority case and it is the end of the working hours of the first employee. So this way the first will fill in the second case solver who will continue to work on a solution. Usually, there is nobody else in the conversation with the customer unless there is a management or customer escalation. From the client side, one of the employees creates the ticket and keeps contact with Cloudera. This person could change over time but the reasons behind this are unknown to us. It can also happen that the people from the customer side go on holidays and there is no conversation during that time frame so the solving date often increases in these situations.

Based on the domain part of the email address of the comment creator it can be determined which side (Cloudera or customer) is the comment from. The number of comments for a case could change for each ticket depending on the severity, the customer response activities, and also the related files. Each time a new file is attached to a case there is a new public comment created by a bot user.

4.2.5 Escalations

There are four types of escalations at Cloudera:

- **Backline**: Not all components have backline members but if there is any then the support members can ask help from them before going to engineering.
- **Engineering**: When the support team can not solve a case, they escalate it to engineering for help. The customer is in contact with the support team while the support members are communicating with engineering. These complex cases can remain open for a potentially longer period and sometimes because of the slow communication.
- **Management**: The Management escalation are usually created on behalf of the customer. It happens when the client has a non-technical problem, a non-break/fix issue, when they are dissatisfied or the case is about a known product limitation.
- **Customer**: This is the only type of escalation that is created by the customer. They have an option to push a button when they would like to have more attention.

4.2.6 Checking Account Health with Z-Scores

The company created a data-driven abnormality detection program that utilizes the concept of Z-scores to understand a stable state of an account for identified categories and compares it to the current state to determine if abnormalities exist in that category.

Z-Score is a statistical measure that describes the position of a raw score in terms of its distance from the mean, when measured in standard deviation units.

$$z = \frac{x - \mu}{\sigma} \tag{4.1}$$

Equation 4.1 shows the calculation for z-scores where x stands for the raw score, μ the mean and σ the standard deviation.

In general, a Z-Score at or near 0 indicates the account is in its stable state. As the Z-Score increases above 0, the higher it goes the more abnormal the current activity is. If the Z-Score is going negative, it indicates that the abnormality is phasing out and the account is either entering a new normal state or returning completely to the original normal state.

For example, if a customer has created 4 support tickets over the course of a month (1 on average a week), and suddenly creates 8 cases in the last week, this is

abnormal and may warrant review. In this case, the Z-Score of this category will be high and can indicate an area of concern.

The primary objective of this program is to provide an early indication of an account that is having an abnormal experience and give directed guidance to what is driving that experience.

There is a Total Z-Score value that summarizes the categories below which I included in the initial feature list. The following list highlights the categories, with a small description of why the category was chosen.

Case Volume - Increased volume may indicate activity that is going astray (new installation, upgrade, use case deployment).

S1 Volume - Specific increases to S1 cases may indicate general instability within an environment that needs additional attention.

Case Age - Increased age in cases may indicate issues with the product, which require engineering and cause delays that are not customer driven.

Continuous S1 Volume - Identifies if cases are in an S1 state for a continuous 24, and 48 hour mark. As this event is rare; this has a significant impact on the Z-Score.

Upgrade Tagged Cases - As our customers are required to move from legacy lines to new CDP products, any upgrade related issues should be treated as high priority

Customer Escalations - Increase in escalation volume can indicate pressure at the customer site, or a major incident that may not be concentrated on a single ticket.

4.2.7 Net Promoter Score

The company measures the satisfaction of its customers and I think it is valuable to include the Net Promoter Score (NPS) when looking at the data. The NPS is an index ranging from -100 to 100 that measures the willingness of customers to recommend a company's products or services to others. It is used as a proxy for gauging the customer's overall satisfaction with a company's product or service and the customer's loyalty to the brand. Sadly not all the clients are responding to the survey so there are many missing records for NPS score.

4.2.8 Account size

It is a challenge to measure the size of a customer. It can be determined based on the number of tickets, or the number of cluster nodes, cores, and the amount of money that they are spending. The spent amount would not be the best for modeling as a 1 dollar difference could confuse the model although those customers are within the same order of magnitude. The best approach would be to categorize the size of the clients. The company keeps track of an account's journey stage which represents the progression of the account over time, including the steps by which they progress from prospects to customers. We distinguish six different types individually for each account (both the LOB and the logo accounts).

4.2.9 Sales information

In the sales account table, there is information about the customer's geography and sales territory. For the US Government Support group, the candidates should have 'Public Sector' as the sales region but I also included this field for the Premier model. The other requirement was to exclude educational industries. Hence, from the sales table, the following fields were imported: sales region, industry, account type, and account segmentation. The last two attributes were suggested by one of the team leaders. These are all categorical features that need to be encoded for the modeling.

4.2.10 Entitlement - Opportunity

Entitlements represent the units of customer support with the provided level and type. Based on the entitlements, it can be decided whether a customer is eligible for the requested service or not . Opportunities are past sales or pending deals managed by the sales team. Both belong to accounts and in theory, without an opportunity, there would be no entitlement but they are not well-documented in the system and there could be some missing records. That is the reason why I would investigate entitlements and opportunities and if needed, combine these two fields.

As for entitlements, I looked at the current active entitlements, the total number during the customer's journey, and the months that passed since the first entitlement. The last attribute shows the time since the client is working with the company.

I split the opportunities based on the involved products and created boolean variables for example, whether they have legacy opportunities or CDP opportunities.

4.2.11 Clusters

A cluster means a set of hosts running independent services. Each client uses clusters on which they run the software but the exact number of machines is unknown unless the client sends diagnostic bundles or shares this information. This is why the company does not know every detail about the consumer's environment. The computers in the cluster are called nodes and the processor CPU-cores are referred to as 'cores' in the feature names. I separated the production and non-production clusters and counted the totals for clusters and their nodes. There is a table available with the CDSW cluster technical information. Besides the specifications of the environment, the cloud usage for each account is also tracked.

4.3 Data Exploration

After identifying the available data and determining the initial features, the next phase is to explore the quality and the possible steps for data preparation.

For checking the statistics of each attribute, I needed to gather the data together. There were four views created in Impala: cases, escalations, accounts and case comments information. For the view of the cases, I gathered the number of cases created and closed, the number of cases with severity types, the average days open, and days to solve in different time periods (last 3, 6, and 12 months) along with the total case counts in the existing products. I combined the count of escalation types with the distinct case owners into the second view. The case comment table is much bigger than the other tables since each ticket belongs to 1 to 4000 rows. The fourth view is for account details including sales region, industry, account type, segmentation, number of clusters, nodes and cores, number of entitlements, and the booleans whether they have opportunities.

If I would like to start to investigate the data in the different tables, I would realize that there are many missing values. But for business reasons, I am not interested in customers who have not created any tickets in the last few months as they are not active. Hence, the four tables were inner merged based on the account id. This way only the customers with created cases in the last year were kept. This leaves 418 Premier possible accounts and 83 US Government Support candidates.

There were no missing values for the number of created cases, account details but the cases are often not escalated thus for the Premier group 343 rows, for the US 78 records are missing. In practice, this results in them not having any escalations so we could fill these attributes with zeros. This could be the appropriate action for the cluster details and units consumed. In Figure 4.7 are visible all the attributes that may have Null or NaN values. 296 records are not filled for the active subscription

which is a boolean variable so if it is empty we can conclude that they do not own any entitlements. In the escalation attributes, 293 to 343 rows are missing.

cust_esc_per_case_last_3_months	343	current_journey_stage	0
backl_esc_per_case_last_3_months	343	current_logo_journey_stage	0
mgmt_esc_per_case_last_3_months	343	active_entitlements	0
eng_esc_per_case_last_3_months	343	total_entitlements_ever	0
cust_esc_per_case_last_6_months	334	months_since_entitlement	1
backl_esc_per_case_last_6_months	334	nps_score	121
mgmt_esc_per_case_last_6_months	334	zs_total	0
eng_esc_per_case_last_6_months	334	num_contact	55
cust_esc_per_case_last_12_months	293	prod_clusters	0
backl_esc_per_case_last_12_months	293	non_prod_clusters	0
mgmt_esc_per_case_last_12_months	293	prod_cluster_nodes	0
eng_esc_per_case_last_12_months	293	non_prod_cluster_nodes	0
avg_cldr_comments_last_3_months	0	cdp_total_units_consumed	373
avg_cust_comments_last_3_months	0	total_cdsw_cluster_cores	376
avg_cldr_comments_last_6_months	0	total_cdsw_cluster_nodes	376
avg_cust_comments_last_6_months	0	has_active_cdp_subscription	296
avg_cldr_comments_last_12_months	0	has_cdp_opportunity	0
avg_cust_comments_last_12_months	0	has_legacy_opportunity	0

Figure 4.7: Number of missing values in Premier

Figure 4.8 describes the statistics for NPS Scores and total Z-Scores. The left table shows the Premier Support accounts and the candidates while the right displays the US Government Support customers and its candidates. One difference is that while the NPS Score's minimum in the left table is -100, it is 0 in the right which means that the customers are not as satisfied in the left group as they are in the right group. Many records are missing in the NPS Score compared to the Z-Scores field probably because it is a newly introduced metric and not all the customers are willing to fill out the survey. It can be included in the Premier model but most likely I did not use it as a feature in the US Government model.

	nps_score	zs_total
count	297.000000	418.000000
mean	74.958552	2.472276
std	39.378958	8.356658
min	-100.000000	-3.275512
25%	66.666667	0.000000
50%	97.058824	0.000000
75%	100.000000	1.683914
max	100.000000	110.211027

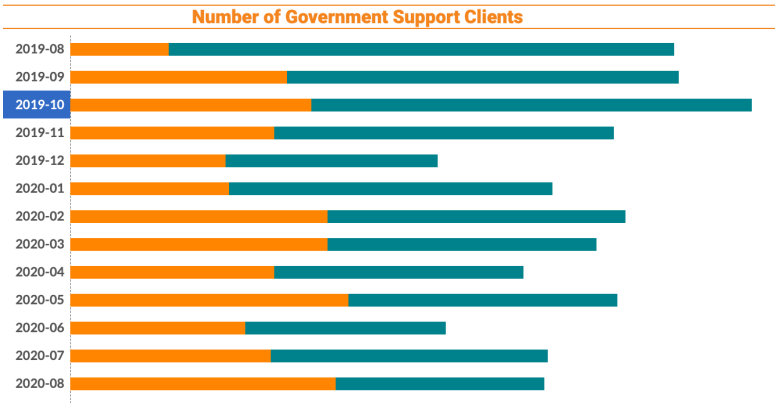
	nps_score	zs_total
count	2.000000	83.000000
mean	50.000000	0.904581
std	70.710678	4.267893
min	0.000000	-0.924980
25%	25.000000	0.000000
50%	50.000000	0.000000
75%	75.000000	0.000000
max	100.000000	35.795717

Figure 4.8: Net Promoter Score and Z-Score exploration

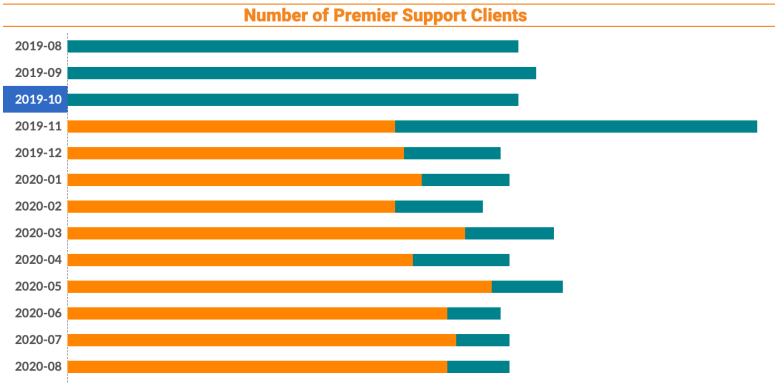
4.4 Challenges

4.4.1 Missing history

The first conspicuous issue occurred while checking the first records in Premier support. It was fascinating that most of the clients were moved to Premier in the same month but this issue was not visible for the Government support clients. In Figure 4.9 both bar charts have the same date frame and the orange bars show the active client number in each month. While 4.9a has some active customers in October 2019, 4.9b has no active customers in Premier. October 2019 was the first month with Premier Support records but in the next month (November 2019) this number has highly increased.



(a) US Government



(b) Premier

Figure 4.9: The number of clients in the support groups

The reason behind this issue revealed soon itself after understanding what happened when Cloudera merged with Hortonworks. The official merger happened in February 2019 but the two companies' data migration to have historical data from both sides happened later only in November 2019. The name 'Premier' came from the legacy Hortonworks while legacy Cloudera had a different name for the dedicated support team offering.

Thus, before November 2019, Premier Support did not exist so a different way was needed to track who and when were moved to Premier Support. Luckily, the company has records of sales activity and the support renewals or changes can be found in the databases. It was much harder to extract the start date this way.

4.4.2 Balanced - imbalanced data set

Knowing if we deal with balanced or imbalanced data in the Machine Learning world is crucial as the whole project can be useless without this information. This project is a classification problem when the goal is to decide whether an account is eligible for the specialized support group. For unsupervised learning problems, it would not cause any issue.

When a classification has 99% accuracy the user concludes that the build was successful, the model is well-trained, and can determine the right classes. However, if the data set is imbalanced and from 100 records only 1 is in the A class and the other 99 are in the B, the model will predict all records to be in the B class which will lead to a 99% accuracy.

In the US Government Support group, 58% of the records are currently flagged with the special support and 42% of the accounts are candidates. This data set can be interpreted as balanced. In contrast, the candidates in the Premier data set are the majority class with 92% and the currently active Premier accounts are only 8% of the records leading to imbalanced data.

Solutions

There are many solutions in the related literature for dealing with imbalanced data sets [13][14]. The following techniques

1. Changing the weight of each class can be another solution for classification models.

2. Threshold adjusting is a manual way for balancing.

3. Resampling techniques can also help to balance the data set:

    - Undersampling means that eliminating some records from the majority class can lead to information loss.
    - Oversampling means generating new records for the minority class that may cause the classifier to overfit.

Before creating the models for the Premier data set it is necessary to use any of the above solutions to balance the data. Otherwise, a great accuracy model can be achieved without being useful in business life.

Chapter 5

Data Preparation

The goal of the Data Preparation phase is to refine the available data and prepare it for modeling. This step is the most important sincse without the right data, building a good model is not easy thus it can require a long time.

5.1 Transform data

5.1.1 Missing data

Handling missing values has a huge impact on the robustness of future models. Many machine learning algorithms do not support null values hence every data analyst should work with empty rows.

There are many ways for dealing with missing values. The first method is deleting every row that has nulls but this could cause information loss. Another technique is to fill in the empty records with some values such as mean or median. When the attribute is categorical the user could create a new category for the missing values.

In the Premier and US Government data sets, there were some missing records as can be seen in Figure 4.7. Fortunately, the company database is well-maintained and not filled with messy data thus the reason behind the Null or NaN values was the non-existence. When a case is not escalated, it does not have any number of escalations. When a customer is not using CDP products, they will not consume any cloud units. With these fields, the method was to fill them with zeros. There was one customer without months since entitlement but after merging the accounts with the cases table that row disappeared.

The exceptions are the number of contacts and the NPS Score. The NPS Score was filled with the mean which is 75 for the Premier data set and the field was dropped for the US Government as there were only two records so the majority was missing. The rows with an empty number of contacts were dropped because that

would indicate that the customer is not contacting the support. If that is the issue it is not relevant from the business perspective.

5.1.2 Categorical variables

In the feature set, there were ten categorical variables where the binary variables have two values: false or true. These can be used as numerical attributes where 0 represents false and 1 means true. The only step was to convert the boolean type to integer. The non-binary features such as account type, segmentation, sales region, and industry are strings thus they should be converted to numerical before using them in the machine learning models.

Encoding means that the categorical variables will be represented with a number. This step is essential as many algorithms are not able to handle other value types than numbers. The most widely used techniques for categorical variable encoding are Ordinal Coding and One Hot Coding. Ordinal coding does not create new columns, it changes the existing values by assigning an integer to each category. The drawback of using ordinal coding it could mislead the model with a non-existing order between the categories. The other popular technique is One Hot Coding where a new column is created for each category as a binary variable where 0 represents the absence and 1 shows the presence of the category. This method can greatly increase the feature set size depending on the number of distinct values [15].

Account type, account segmentation, and sales region all have four distinct categories and the industry contains 12 sub-categories. For the Premier group, all values are included but the initial requirement for US Secure candidates was to only investigate the Public Sector sales region and the potential customers should not work in the educational industry. After eliminating the missing values, merging the tables, and encoding the variables there were 24 new columns added to the Premier data and 13 new features were created for the US Secure group. The original attributes can be dropped and a result is a data set with only numerical variables.

5.1.3 Numerical variables

After eliminating the missing records from the numerical variables, the next important task is to determine whether scaling is necessary. Some machine learning algorithms are insensitive to feature scaling such as decision trees but the performance of other models depends significantly on the records. These models are mostly either calculating distances between data points or they are using gradient descent optimization techniques.

Scaling techniques

There are multiple scaling methods such as normalization and standardization. Using them before building machine learning models can help improve the performance [16].

Normalization (Min-Max scaling) is a scaling technique where the rescaled feature values range from 0 to 1. Equation 5.1 describes the formula for normalization using the minimum and maximum values of the feature.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{5.1}$$

Standardization (or Z-score normalization) is the process of rescaling the values to center them around the mean with a unit standard deviation. This technique can be helpful when the data follows a Gaussian distribution. The formula for standardization is the same as the Z-Score calculation shown in equation 4.1. The standardized value is simply the raw score minus the population mean, divided by the population standard deviation.

Both methods are available in Python after importing the scikit-learn preprocessing package. MinMaxScaler and StandardScaler were fit on the training set and for the validation and test data, the transform function was used.

5.1.4 Balancing

Contrary to the US Government data, the Premier set was imbalanced. Building a model on the Premier data set could create an illusion of a well-performing model. For handling imbalanced data sets, there is a library called imbalanced-learn available in Python. This package includes approaches for resampling techniques. For undersampling the RandomUnderSampler and for oversampling the SMOTE algorithm was tested.

SMOTE stands for Synthetic Minority Over-sampling Technique and it has been designed to generate new samples for the minority class. This method creates 'synthetic' examples instead of duplicating records. These samples are generated in the following way:

1. Take the difference between the feature vector sample and its nearest neighbor
2. Multiply this by a random number between 0 and 1
3. Add the result to the current feature vector sample

These steps cause the selection of a random point between two specific features.

The *RandomUnderSample* method removes samples randomly from the majority class until the minority reaches the desired rate between the minority and the majority class [17].

Both techniques were tried out on the Premier data set and then each model was built on two types of data.

5.2 Feature Analysis

The Feature Analysis phase includes analyzing the correlation between the variables and the target variable and selecting the relevant features for the models. The model performance could increase significantly with the right feature set.

5.2.1 Feature selection techniques

Usually running the machine learning models with the initial feature set does not result in the best model performance. Selecting the relevant features for the current problem and discarding the irrelevant ones can help to build a useful model. High dimensionally models have a long training time and risk for overfitting. Choosing the right feature selection method is not an easy task and it is always important to check the effectiveness of the selected variables. The following three main categories are distinguished based on the relationship between the feature selection and the machine learning algorithm [18] [19].

- **Filter methods**: Filter methods use univariate statistics to pick the features as a preprocessing step before training any model.
- **Wrapper methods**: These methods involve machine learning algorithms and try to select variables by evaluating all combinations of features for better performance.
- **Embedded methods**: Embedded methods take into account the model's feature importance during training.

Figure 5.1 describes the above-mentioned methods with their advantages, disadvantages, and a few examples.

Information Gain, Variance Inflation Factor, Chi-Square Test, and Recursive Feature Elimination techniques will be introduced from filter and wrapper methods as these were tested during this project.

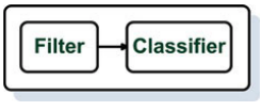
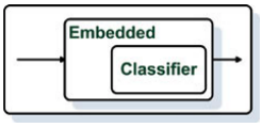
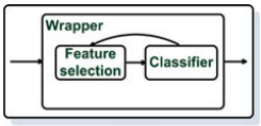
Method	Advantages	Disadvantages	Examples
Filter			
	Independence of the classifier Lower computational cost than wrappers Fast Good generalization ability	No interaction with the classifier	Consistency-based CFS INTERACT ReliefF $\mathcal{M}_d$ Information Gain mRMR
Embedded			
	Interaction with the classifier Lower computational cost than wrappers Captures feature dependencies	Classifier-dependent selection	FS-Percepton SVM-RFE
Wrapper			
	Interaction with the classifier Captures feature dependencies	Computationally expensive Risk of overfitting Classifier-dependent selection	Wrapper-C4.5 Wrapper SVM

Figure 5.1: Feature selection techniques
Source: [18]

Information Gain

Information Gain is a univariate feature selection technique. Mutual information measures the amount of information shared by two variables and therefore, the dependence of one variable on another. If X and Y are two independent variables, they do not share any information. With machine learning, maximizing the information would be the goal as there is a dependence between the features and the target. [20]

Variance Inflation Factor

Variance Inflation Factor (VIF) is a statistical method to check whether the given data set has multicollinearity. This means that there are multiple highly correlated variables and they contain similar information. These features can cause unreliable models and weak performance. The calculation formula is shown in equation 5.2 where R_i^2 is the coefficient of determination for the i^{th} variable [21].

$$VIF_i = \frac{1}{1 - R_i^2} \tag{5.2}$$

In general, a VIF_i value greater than 10 shows that there is multicollinearity in the data set.

Chi-Square Test

Another feature selection method, the Chi-square test, is used to provide information on whether a significant relationship exists between two categorical features in the data set. It is a statistical test with two hypotheses:

- Null Hypothesis: There is no relationship between the two variables
- Alternate Hypothesis: There is a relationship between the two variables

The formula for chi-square is provided in equation 5.3 where O represents the observed data and E means expected data [22].

$$\chi^2 = \sum \frac{(O - E)^2}{E} \tag{5.3}$$

The Chi-Square Test as a statistical test uses the chi-square table to calculate the p-value. The significance level is usually chosen to be 5%. When the p-value test result is lower than the significance level (0.05), the null hypothesis can be rejected thus there is a relationship between the variables. The other use case is to check the chi-square scores without p-values. The higher the chi-square value the higher the relationship between the variables. The model results can be improved by selecting the features with high chi-square values.

Recursive Feature Elimination

The Recursive Feature Elimination (RFE) is a wrapper method that performs feature selection by iteratively training a model and eliminating the least important features. It has a parameter for the desired number of features and the algorithm is running until it reaches this value. The estimator provides information about the feature importance by assigning weights to the variables.

5.2.2 Feature selection

Categorical features

The impactful categorical features were selected with the help of the Chi-Square Test. Table 5.1 shows the remaining categorical variables for both groups where the only common attribute is the Named account segmentation.

The first approach was to analyze whether the Chi-Square Test null hypothesis can be rejected. This can be checked in a for cycle showed in Figure 5.2.

The hypothesis test for the US Government group found two features that have a relationship with the target. The target variable is the flag whether they are currently in the US Goverment support. These were the *is_first_class_customer*

Table 5.1: Categorical features

Feature Name	Premier	US Gov
is first class customer?		✓
has legacy opportunity?	✓	
account segment - Strategic		✓
account segment - Named	✓	✓
account segment - Commercial	✓	
industry - Banking...	✓	
industry - Telecom...	✓	
infustry - Manufacturing..	✓	

```
categorical_columns = X_cat.columns
chi2_check = []
for i in categorical_columns:
    if chi2_contingency(pd.crosstab( y_cat, X_cat[i]) )[1] < 0.05:
        chi2_check.append('Reject Null Hypothesis')
    else:
        chi2_check.append('Fail to Reject Null Hypothesis')

result = pd.DataFrame(data = [categorical_columns, chi2_check]).T
result.columns = ['Feature Name', 'Hypothesis']
```

Figure 5.2: Chi-Square Test in Python

boolean and one of the account segmentation encoded variables. The two null hypothesis rejections are shown in Figure 5.3.

The other calculation was created with the *SelectKBest* library that allows the user to set the score function to chi-square and it has fit and transform functions. When printing the top 10 scores and feature names out, the first two features with the highest numbers were the same as in Figure 5.3.

The difference is that while the hypothesis test would drop each variable where the p-value does not reach the significance level, looking at the scores can help to identify the ranking between the features. Figure 5.4 shows the top 10 features based on the chi-square value. There is a bigger gap between the first two variables and the third score. As the last feature selection technique will be based on a machine learning algorithm, more categorical features were kept than the few suggested by the hypothesis test.

Numerical features

In the feature set, there were more than 80 numerical variables. Eliminating the non-important features could help decrease multicollinearity and increase performance. The first feature selection was carried out using the Mutual Information

	Feature Name	Hypothesis
0	is_ibm	Fail to Reject Null Hypothesis
1	is_on_cal	Fail to Reject Null Hypothesis
2	is_first_class_customer	Reject Null Hypothesis
3	has_cdp_opportunity	Fail to Reject Null Hypothesis
4	has_legacy_opportunity	Fail to Reject Null Hypothesis
5	has_active_cdp_subscription	Fail to Reject Null Hypothesis
6	account_type_Customer	Fail to Reject Null Hypothesis
7	account_type_Support Only Account	Fail to Reject Null Hypothesis
8	industry_Banking, Finance & Insurance	Fail to Reject Null Hypothesis
9	industry_Bus, Legal, Consulting & Misc Services	Fail to Reject Null Hypothesis
10	industry_HealthCare, Pharma & Biotech	Fail to Reject Null Hypothesis
11	industry_Manufacturing & Automotive	Fail to Reject Null Hypothesis
12	industry_Natural Resources: Energy, Utilities,...	Fail to Reject Null Hypothesis
13	industry_Telecom, Media & Broadcasting	Fail to Reject Null Hypothesis
14	industry_US Government (Federal)	Fail to Reject Null Hypothesis
15	industry_US Government (State & Local)	Fail to Reject Null Hypothesis
16	account_segment_Commercial	Fail to Reject Null Hypothesis
17	account_segment_Named	Fail to Reject Null Hypothesis
18	account_segment_Strategic	Reject Null Hypothesis

Figure 5.3: Chi-Square Test for US Government group

Features	Score
is_first_class_customer	9.388547
account_segment_Strategic	6.577047
account_segment_Named	1.557862
account_segment_Commercial	1.520023
industry_HealthCare, Pharma & Biotech	1.490651
industry_Natural Resources: Energy, Utilities,...	1.424242
industry_Telecom, Media & Broadcasting	1.424242
industry_Bus, Legal, Consulting & Misc Services	0.702128
industry_Manufacturing & Automotive	0.702128
industry_US Government (Federal)	0.637945

Figure 5.4: Chi-Square Scores for US Government group

Gain filter method where 30 features were kept with the highest scores. Using the *mutual_info_classif* library in Python, as can be seen in Figure 5.5, made it easier to calculate the information score for each variable and then only keep the 30 largest scores. The most informative feature was the *months_since_entitlement* as visible in Figure 5.6.

The second feature selection technique was calculating the VIF value for each attribute. There is a *variance_inflation_factor()* in Python, shown in Figure 5.7, that allows the user to easily calculate the VIF value for each attribute.

The features with the highest scores were eliminated iteratively to reduce multi-collinearity. For example, in the initial data set the number of cases created in the last three months, six months, and twelve months were included. The VIF value for these variables was high so new features were created such as created cases in the

```
#----------------------------
#Mutual Information Gain (MI)
#----------------------------
importances = mutual_info_classif(X_num, y_num)
feat_importances = pd.Series(importances, X_num.columns).sort_values(ascending = False)
feat_importances.nlargest(30).plot(kind = 'barh',color = 'teal')
top30features=feat_importances.nlargest(30).keys()
```

Figure 5.5: Mutual Information Gain in Python

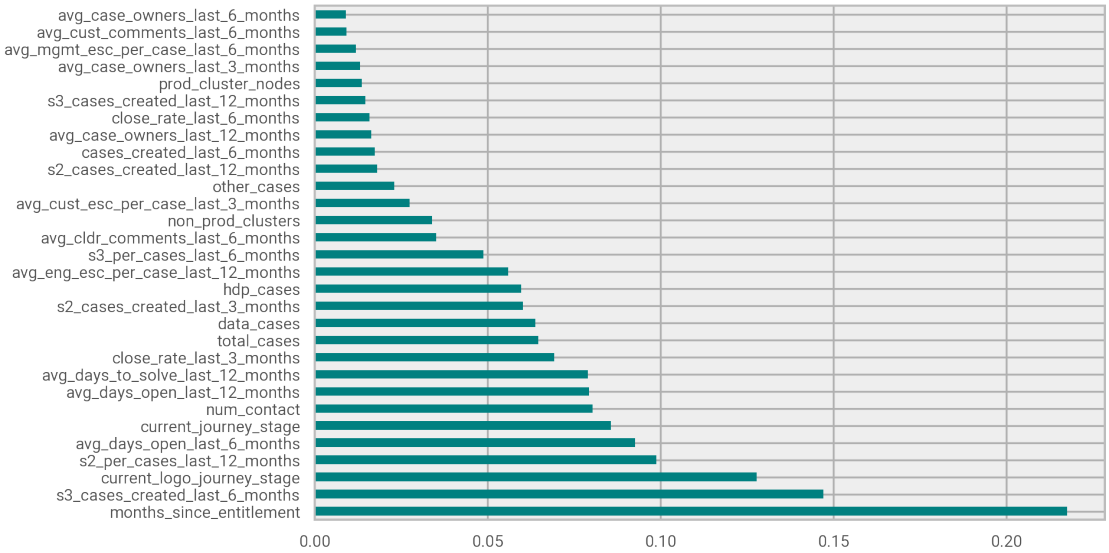


Figure 5.6: Mutual Information Gain - top 30 features

last three months compared to the previous six months, the number of S1 cases per case, and close rate for the given time period. These new features had lower VIF values.

The Recursive Feature Elimination was tried out on different models. The feature set in the best-performing model was considered the final feature set. RFE is one of the scikit-learn feature selection methods and has a *fit()* function to recursively select the relevant features for a chosen model, as shown in Figure 5.8. It has a parameter that allows determining the desired number of features. 10, 15, and 20 features were tested during this project. Only the algorithms that have *coef_* or *feature_importances_* attributes can be used with RFE.

```
#------------------------------
#Variance Inflation Factor (VIF)
#------------------------------
vif_info = pd.DataFrame()
vif_info['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif_info['Column'] = X.columns
vif_info.sort_values('VIF', ascending=False)
```

Figure 5.7: Variance Inflation Factor in Python

```
#-----------------------------------
#Recursive Feature Elimination (RFE)
#-----------------------------------
rfe_selector = RFE(estimator=LogisticRegression(), n_features_to_select = 15)
rfe_selector = RFE(estimator=RandomForestClassifier(), n_features_to_select = 15)
rfe_selector = RFE(estimator=GradientBoostingClassifier(), n_features_to_select = 15)
rfe_selector.fit(X,y_num)
X.columns[rfe_selector.get_support()]
```

Figure 5.8: Recursive Feature Elimination in Python

Feature set

Table 5.2: Final feature sets

Feature Name	Type	Premier	US Gov
Support Activity			
# S1 cases created (12 months)	N		✓
# S2 cases created (12 months)	N	✓	✓
# S3 cases created (6 months)	N	✓	✓
# S3 cases created (12 months)	N		✓
# S2 cases / # cases (6 months)	N	✓	
# S2 cases / # cases (12 months)	N	✓	✓
# S3 cases / # cases (6 months)	N	✓	✓
Close rate (12 months)	N		✓
Avg. Days to Solve (12 months)	N	✓	✓
Avg. Days Open (6 months)	N	✓	
Avg. Case owners (12 months)	N		✓
Customer information			
Months since entitlement	N	✓	✓
Current account journey stage	N		✓
Current logo account journey stage	N	✓	✓
Number of contacts	N	✓	
Number of total cases	N	✓	✓
Number of other type cases	N	✓	
Has legacy opportunity?	C	✓	
Is first class customer?	C		✓
Account Segment - Named	C	✓	✓
Industry - Banking...	C	✓	
Industry - Telecom...	C	✓	
Industry - Manufacturing..	C	✓	
Cluster information			
# Non-production Cluster Nodes	N	✓	✓

After selecting the variables that led to the best-performing models the final list of features is provided in Table 5.2 for both the Premier and US Government groups. The feature type can be either categorical (C) or numerical (N).

For the Premier group, the correlation between the final numerical features and target variable can be seen in Figure 5.9. This shows that the following variables are highly correlated with each other and they are not the target variable:

- *s2_cases_created_last_12_months* with *s3_cases_created_last_6_months*
- *s2_per_cases_last_6_months* with *s2_per_cases_last_12_months*
- *avg_days_to_solve_last_12_months* with *avg_days_open_last_6_months*
- *s2_cases_created_last_12_months* with *total_cases*
- *s3_cases_created_last_6_months* with *total_cases*

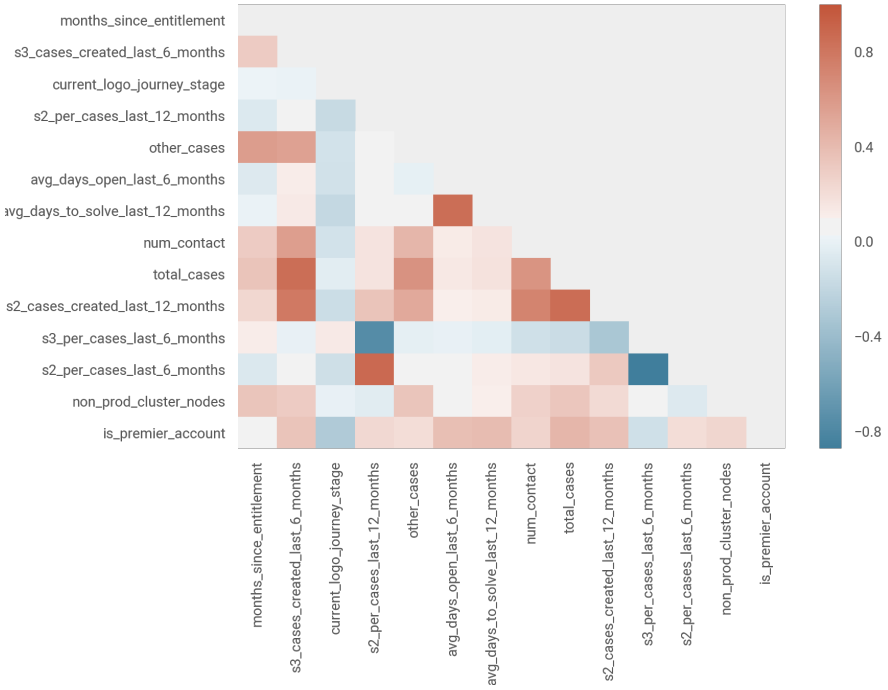


Figure 5.9: Correlation between final numerical features and target (is_premier_account) variable - Premier

Usually, when the correlation between two numerical variables is greater than 0.7, there is a chance for multicollinearity. The VIF value is used for checking whether the data set has multicollinearity. In Figure 5.10 are the VIF values for the final features and every score is under 10 thus the feature set will remain even if there is a correlation between some variables.

	VIF	Column
11	8.328046	s2_per_cases_last_6_months
9	7.301469	s2_cases_created_last_12_months
8	6.971673	total_cases
3	4.818004	s2_per_cases_last_12_months
10	4.738012	s3_per_cases_last_6_months
1	4.548068	s3_cases_created_last_6_months
6	3.944061	avg_days_to_solve_last_12_months
5	3.810972	avg_days_open_last_6_months
4	2.388840	other_cases
7	2.327753	num_contact
0	1.653070	months_since_entitlement
12	1.262287	non_prod_cluster_nodes
2	1.175665	current_logo_journey_stage

Figure 5.10: VIF values of the final numerical features - Premier

	VIF	Column
11	8.887562	s3_cases_created_last_12_months
1	7.975683	s3_cases_created_last_6_months
13	6.188577	avg_case_owners_last_12_months
7	5.551778	s2_cases_created_last_12_months
9	3.483018	total_cases
3	3.322289	s2_per_cases_last_12_months
8	3.253638	s3_per_cases_last_6_months
4	2.910127	avg_days_open_last_6_months
10	2.872614	avg_days_to_solve_last_12_months
12	2.048471	s1_cases_created_last_12_months
14	1.976393	current_journey_stage
2	1.966562	current_logo_journey_stage
6	1.283325	non_prod_clusters
5	1.231598	close_rate_last_12_months
0	1.227282	months_since_entitlement

Figure 5.11: VIF values of the final numerical features - US Government

Figure 5.11 shows the variance inflation factors for the selected attributes for the US Government and every score is below 10.

In Figure 5.12 is the heatmap for these final features. The following features are highly correlated on the figure:

- *s3_cases_created_last_12_months* with *s3_cases_created_last_6_months*
- *avg_days_to_solve_last_12_months* with *avg_days_open_last_6_months*
- *s2_cases_created_last_12_months* with *avg_case_owners_last_12_months*

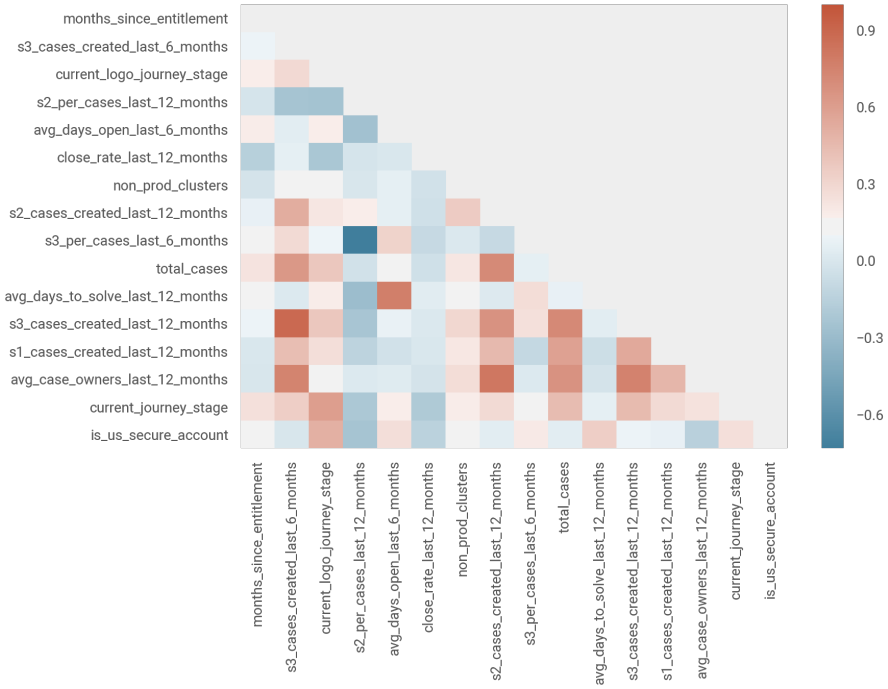


Figure 5.12: Correlation between final numerical features and target (is_us_secure_account) variable - US Government

Chapter 6

Modeling

6.1 Machine Learning techniques

For this project, multiple supervised learning models were tried out. The target variable is a boolean whether the account is eligible for the specialized support groups. Predicting the complexity of customer complaints means forecasting the customers who could move to a higher support category.

The solution requires a classification algorithm that can decide whether the accounts belong to class 0 or class 1. These classifiers were used to build models: Random Forest Classifier, Gradient Boosting Classifier, Support Vector Machine, and Logistic Regression. All of these four classifiers are widely known. Two of them are ensemble methods based on decision trees with bagging or boosting techniques.

Decision Tree

A decision tree uses a tree-like graph decision. Take the decision tree as a logical function. The input to the function is the subject or all attributes of the situation, and the output is a 'yes' or 'no' decision value. In Figure 6.1 we can observe the general structure with example questions from the feature set where *eligible* shows the 'yes' desicion and *not eligible* represents the 'no' output.

In the decision tree, each tree node corresponds to a property test, each leaf node corresponds to a logical value, and each branch corresponds to a possible value of the test attribute [23].

6.1.1 Random Forest Classifier

Random Forest is an easy-to-use, flexible, simple Machine Learning algorithm that gives often successful prediction [25]. The formula was developed by Leo Breiman. As the base components are tree-structured predictors, and since each of these is

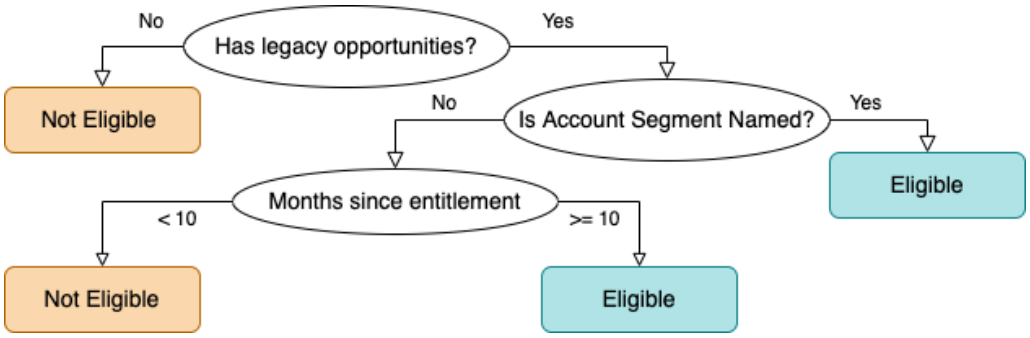


Figure 6.1: A decision tree structure example

constructed using an injection of randomness, the method is called 'random forest'. The algorithm builds multiple decision trees and then aggregates the votes from the decision trees to get the final vote. We can observe the structure of the algorithm in Figure 6.2.

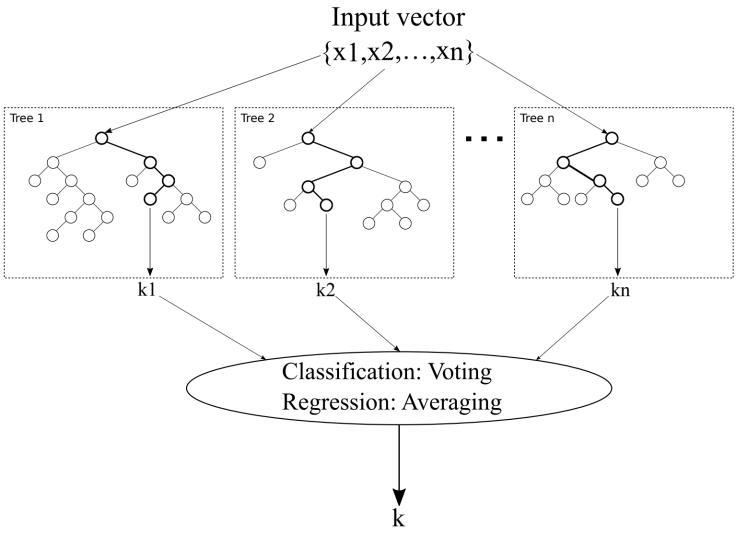


Figure 6.2: Random Forest algorithm
Source: [24]

The algorithm steps are the follows:

1. Draw n bootstrap samples from the original data. The maximum number of samples can be determined by the max_samples parameter.

2. For each of the bootstrap samples, grow a classification tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample m of the predictors and choose the best separation based on those variables.

3. Predict new data by aggregating the predictions of n trees majority votes for classification [26].

The performance of the model can be optimized with hyperparameter tuning, which means that parameters are set before training. Scikit-learn implements sensible default hyperparameters, but these are not always optimal for the problem. Determining the best collection of hyperparameters is a complex, time-consuming task. The most important settings are the following:

- *n_estimators*, the number of trees in the forest (ideally, the more trees there are, the better the performance is);
- *max_depth*, maximum depth of the trees (max. number of levels in each decision tree);
- *min_samples_split*, the minimum number of samples required to split an internal node;
- *min_samples_leaf*, the minimum number of samples required to be at a leaf node.

6.1.2 Gradient Boosting Classifier

Gradient Boosting algorithm is a widely-used machine learning algorithm, due to its efficiency, accuracy, and interpretability.

The random forests rely on simple averaging of models, which is called bagging. The boosting methods are based on a different strategy. The main idea is to add new models to the ensemble sequentially. At each particular iteration, a new weak, base-learner model is trained concerning the previous error of the whole group [27]. If the error function is the classic squared-error loss, the learning procedure would result in consecutive error-fitting. The researcher can choose the loss function.

The algorithm steps are the follows:

1. Calculate the predicted probability distribution based on the current model
2. Calculate the difference between the true probability distribution and the predicted probability distribution. (The goal is to minimize the total loss and for each data point, we wish the predicted probability distribution to match the true probability distribution as closely as possible)
3. Construct a decision tree
4. Predict the target label using all of the trees within the ensemble
5. Repeat until the number of iterations matches the number specified by the hyperparameter (i.e. number of estimators)

6. Once trained, use all of the trees in the ensemble to make a final prediction as to the value of the target variable [28].

To optimize the Gradient Boosting algorithm, hyperparameter tuning can be applied. It is important to choose the parameters to avoid overfitting. The following listing shows the most important parameters of Gradient Boosting:

- *learning_rate* shrinks the contribution of each tree by *learning_rate*. There is a trade-off between *learning_rate* and *n_estimators*.
- *n_estimators,* the number of trees in the forest;
- *max_depth,* maximum depth of the tree (max. number of levels in each decision tree).

6.1.3 Logistic Regression

Logistic Regression despite the name is a classification method with probabilities between 0 and 1. The algorithm is based on the idea of modeling the odds of belonging to class 1 using an exponential function in Equation 6.1 Instead of fitting a straight line or hyperplane, the logistic regression model uses the logistic function. The output of the linear equation is between 0 and 1 as shown in Figure 6.3 [29].

$$f(x) = \frac{1}{1 + e^{-x}} \tag{6.1}$$

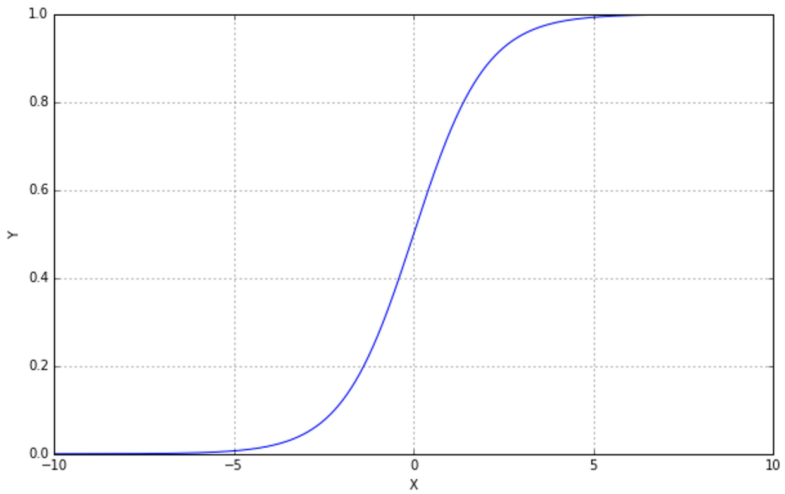


Figure 6.3: Logistic Regression - Sigmoid function
Source: [20]

There are three types of logistic regression based on the target variable: Binary Logistic Regression where there are only two possible outcomes, Multinomial

Logistic Regression with multiple categories, and Ordinal Logistic Regression with multiple categories with ordering.

The following parameters can help with hyperparameter tuning the Logistic Regression model.

- *solvers* which determines the underlying optimization technique such as 'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'.
- *penalty* specifies the norm of penalty that could be 'none': no penalty is added; L2 or L1 penalty term, or both with the 'elasticnet'.
- *C parameter* represents the inverse of regularization strength, the smaller values specify stronger regularization.

6.1.4 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm that finds a hyperplane that creates a boundary between the types of data. Most of the algorithms try to reduce the input space to a lower-dimensional one but the SVM maps the input feature space to a much higher dimensional one [24].

The idea is to map the input space to a higher dimensional one where the separation of the different groups can be performed by a linear method, shown in Figure 6.4.

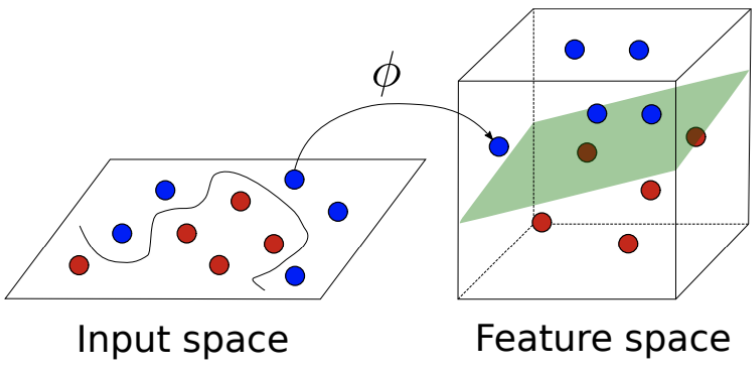


Figure 6.4: Support Vector Machine
Source: [24]

Algorithmically, support vector machines build optimal separating boundaries between data sets by solving a constrained quadratic optimization problem [30]. C-Support Vector Classification (SVC) is the package based on the SVM package to handle classification problems with a regularization parameter.

The following parameters are the most important ones:

- *kernel* function tells what is the similarity between data points in the feature space. Possible values are 'linear', 'poly','rbf', 'sigmoid'
- *gamma* value how far the influence of an example reaches. Possible values are 'rbf', 'poly' and 'sigmoid'.
- *C parameter* represents the inverse of regularization strength, the smaller values specify stronger regularization [31].

6.2 Model building

Building the machine learning algorithm always starts with defining the train set and test set. For hyperparameter tuning a validation set is also required. By definition, the train set is the data on which the model is trained. The test data is used for the final evaluation of the model. The validation set can help to tune the model and set the parameters for better performance. There are many ways to create these sets of data. The first step is to split the data into train and test set as shown in Figure 6.5. The train and validation split can be performed with k-fold cross-validation that will split the data to train and validation set k times and build the model in each iteration.

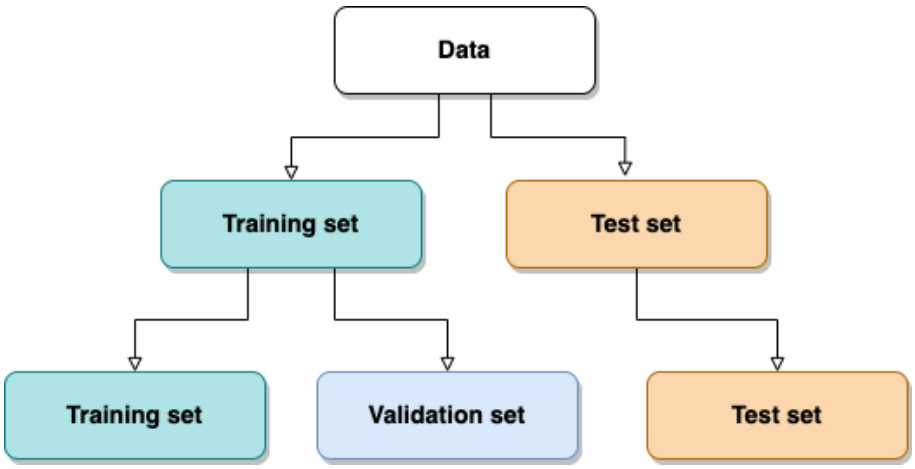


Figure 6.5: Train, validation, and test sets

There are daily snapshots of the customers' current state thus the training and validation set was different than in general. For the test and train data, a snapshot date '2020-12-01' was used while for the validation the date '2021-05-01' was applied. The train-test and validation were transformed separately and in the model building phase, the train-test was split. During data preparation, the train-test set was fit

and transformed with *StandardScaler()* and *MinMaxScaler()* then the validation set was scaled with the *transform()* function.

Scikit-learn has a *train_test_split()* function in Python that can randomly split the data set by a given percentage. During this project, the test and train size rate was 2:8.

Instead of using any kind of built-in cross-validation Python package, the hyperparameter tuning was performed manually. The cross-validation libraries are splitting the train data into train and validation sets based on the number of splits. However, they do not support the separate train and validation sets as an input. The built-in packages have a number for the number of divided group. For example, with 3-fold cross-validation, the data is divided into three parts and in the first iteration, the model is trained on part1 and part2 then evaluated on part3.

Figure 6.6 shows the hyperparameter tuning for the Gradient Boosting Classifier with the *n_estimators*, *learning_rate*, *max_depth* possible parameter values. During each iteration, a new parameter combination is investigated, and the accuracies of the train and validation set are saved to a list. Finally, the results are compared and the model is built with the best parameters and evaluated on the test set.

```
train_acc = []
valid_acc = []
param1 =[]
param2 =[]
param3 =[]
n_est_range=[5,50,100,150,200,250,400,500]
max_depth_range=[1,3,5,7,9]
learning_range = [0.01,0.1,1,10,100]
for para1 in n_est_range:
  for para2 in max_depth_range:
    for para3 in learning_range:
      log_reg = GradientBoostingClassifier(max_depth=para2,n_estimators=para1,
                                           learning_rate=para3,random_state=42)
      log_reg.fit(X_train, y_train)
      param1.append(para1)
      param2.append(para2)
      param3.append(para3)
      train_acc.append(accuracy_score(y_train, log_reg.predict(X_train)))
      valid_acc.append(accuracy_score(y_valid, log_reg.predict(X_valid)))
```

Figure 6.6: Gradient Boosting Hyperparameters

The same code snippet was used to perform the hyperparameter tuning for the other three types of models. The Random Forest Classifier was tuned with the *max_depth* , *n_estimators*, *min_samples_split*, *min_samples_leaf* shown in Figure 6.7.

Figure 6.8 shows the parameter ranges for the Logistic Regression with the *solvers*, *penalty*, and *C* values. There is some limitation as the choice of the solver is based on the penalty. 'l2' is supported by any kind of solver.

```
max_depth_range= [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None]
n_estimators_range= [50,75,100, 125,150, 200, 250, 400, 600,]
min_samples_split_range= [2, 5, 10]
min_samples_leaf_range= [1, 2, 4]
```

Figure 6.7: Random Forest Classifier Hyperparameters

```
solvers_range = ['newton-cg', 'lbfgs', 'liblinear']
penalty = ['l2']
c_values_range = [100, 10, 1.0, 0.1, 0.01]
```

Figure 6.8: Logistic Regression Hyperparameter

The SVC model parameters are the C value, *gamma* and *kernel*. The range with the investigated values can be seen in Figure 6.9.

```
C_range=[0.1, 1, 10, 100, 1000]
gamma_range=[1, 0.1, 0.01, 0.001, 0.0001]
kernel_range = ['linear', 'poly', 'rbf', 'sigmoid']
```

Figure 6.9: SVC Hyperparameters

Chapter 7

Evaluation

7.1 Evaluation metrics

To compare the model results with each other, some metrics are needed.

Confusion Matrix

A confusion matrix, also known as an error matrix, can help to evaluate the models more easily. For binary classification, it is a 2x2 matrix, shown in Figure 7.1. It summarizes the number of correct and incorrect predictions by counting the assessments in each group.

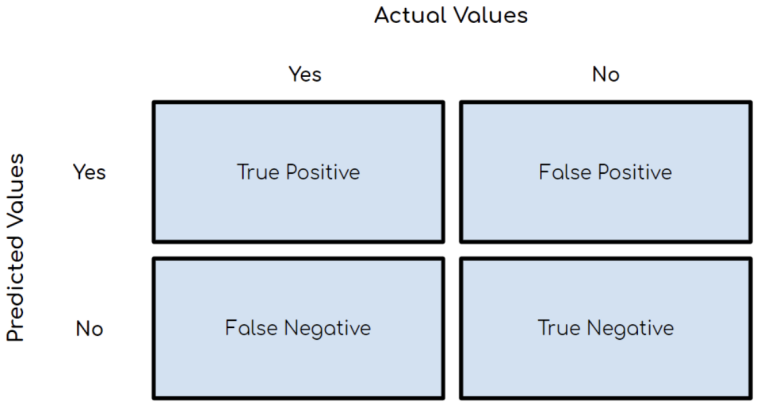


Figure 7.1: Confusion Matrix
Source: [32]

The accuracy, precision, recall, and F1 score can all be expressed from the number of true positive (tp), false positive (fp), the total number of positive (p) results, and the number of true negative (tn), false negative (fn) and the total number of negative (n) results [32].

Accuracy

Accuracy is the proportion of predictions that the model classified correctly. It is defined by Equation 7.1.

$$accuracy = \frac{tp + tn}{p + n} \tag{7.1}$$

Precision

Precision the proportion of relevant instances among the retrieved instances. It shows the correctly identified proportion as shown in Equation 7.2.

$$precision = \frac{tp}{tp + fp} \tag{7.2}$$

Recall

Recall is also known as the true positive rate (TPR). It represents the proportion of the total amount of relevant instances that were actually retrieved. Equation 7.3 shows how recall is expressed.

$$recall = \frac{tp}{tp + fn} \tag{7.3}$$

ROC Curve

The ROC curve (receiver operating characteristic curve) is a graphical method for showing a ratio between the true positives and the false positives. The Area Under the ROC Curve (AUC) provides another way for evaluating which model is better [33]. If the model is perfect, then the AUC is one, and the ROC curve jumps to one right at zero, while a bad-performing model (a no-skill predictor) leads to a diagonal ROC.

7.2 Comparison

Table 7.1 shows the best accuracies for the models and groups after applying the hyperparameter tuning.

The Premier group was divided into oversampled and undersampled models. The best-performing model was Gradient Boosting Classifier with SMOTE applied on the data sets. The validation accuracy was 91% and the test accuracy was 93.8%. These accuracies were reached with the following parameters: the number of estimators = 50, maximum depth = 5, and the learning rate = 1.0. For the undersampled data, the best model was Random Forest Classifier with 83.3% accuracy. Comparing the

Premier models, this 83.3% is the fifth best model. All the oversampled models performed better than the undersampled.

For the US Government group, the best-performing model was a Gradient Boosting Classifier trained on data, that was scaled with *StandardScaler()*, with 90.3% test accuracy and 91% validation accuracy. There were two different parameter sets that performed with the same accuracies. The number of estimators was either 100 or 150, with a maximum depth and learning rate of 1. The second best model had normalized training data and the test accuracy was 87.5%.

Table 7.1: Accuracy of the best-performing models

Model Name	**Validation accuracy**	**Test accuracy**
Premier with Oversampling		
Gradient Boosting Classifier	**0.910**	**0.938**
Random Forest Classifier	0.866	0.910
Logistic Regression	0.883	0.896
Support Vector Machine	0.881	0.881
Premier with Undersampling		
Gradient Boosting Classifier	0.883	0.667
Random Forest Classifier	**0.833**	**0.833**
Logistic Regression	0.733	0.750
Support Vector Machine	0.850	0.667
US Government with Standardization		
Gradient Boosting Classifier	**0.903**	**0.910**
Random Forest Classifier	0.872	0.813
Logistic Regression	0.769	0.625
Support Vector Machine	0.808	0.688
US Government with Normalization		
Gradient Boosting Classifier	**0.872**	**0.875**
Random Forest Classifier	0.833	0.813
Logistic Regression	0.731	0.625
Support Vector Machine	0.846	0.838

The ROC Curve for the Premier Gradient Boosting Classifier can be seen in Figure 7.2. The orange curve on the chart is really close to the perfect model. The area under this ROC curve is 0.983.

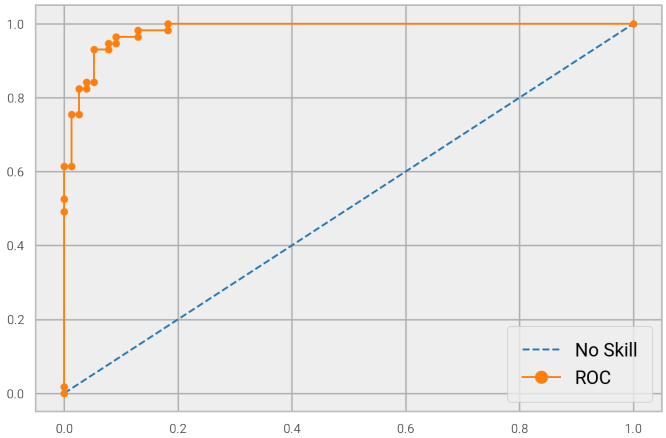


Figure 7.2: ROC curve - Premier

Although the US Government ROC curve is not as good as the best model for Premier it is tending towards the upper left corner in Figure 7.3. The area under this ROC curve is 0.931.

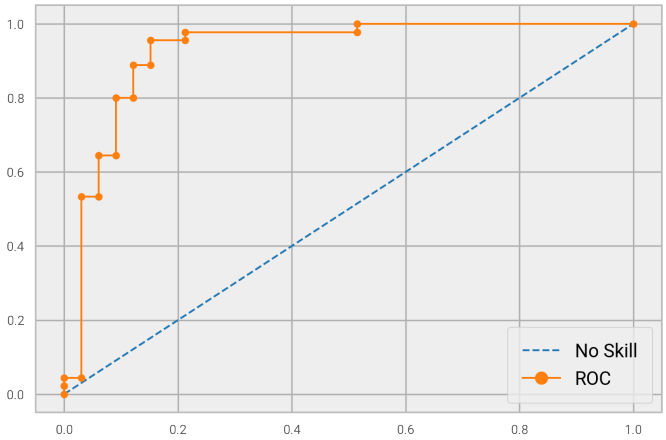


Figure 7.3: ROC curve - US Government

Chapter 8

Deployment

In the Modeling chapter, two well-performing models were built: one for the Premier group and their candidates and the other for the US Government and potential joiners. There are multiple ways to deliver the results to the stakeholders.

Besides the train-test and validation data sets, prediction data was also created and transformed the same way as the other two tables. The snapshot date for the prediction table was '2021-11-01'. The main idea is to predict the customers with the current state to look who can be eligible for a support extension. If an accounts' probability is above 70% and it is not currently flagged with either of the support group the account id should be saved for later.

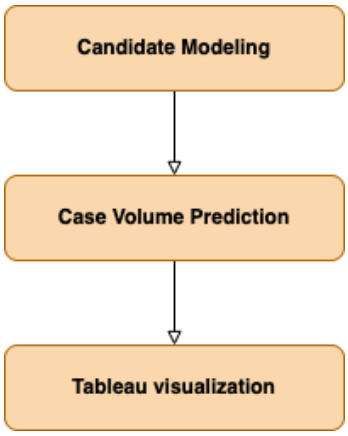


Figure 8.1: Deployment steps

The flow chart of the steps for deployment is visible in Figure 8.1. The first task is to run the models for both groups and save the account names or ids into a table where the likelihood reaches a baseline (currently 70%).

In the second step is the case volume prediction performed. The forecasting model looks at the total numbers, the product level, and also account level volumes. The improvement is to predict volumes for customers who are not currently flagged with special support.

The third task is to visualize the numbers for potential accounts. The challenge was to show the total numbers for currently flagged accounts and have an opportunity to select individual accounts with their additional volumes. The solution was to restructure the underlying table to have a *type* field that indicates whether this is candidate modeling. By selecting type 'Premier' or 'US Government' the account list will only show the currently flagged names. By selecting type 'Non-Premier' or 'Non-US Government' the account list will include the clients without the flags. A total number is calculated and added for the non-flagged types thus the stakeholders can plan the future by looking at the current number with additional volumes. A realization can be seen in Figure 8.2

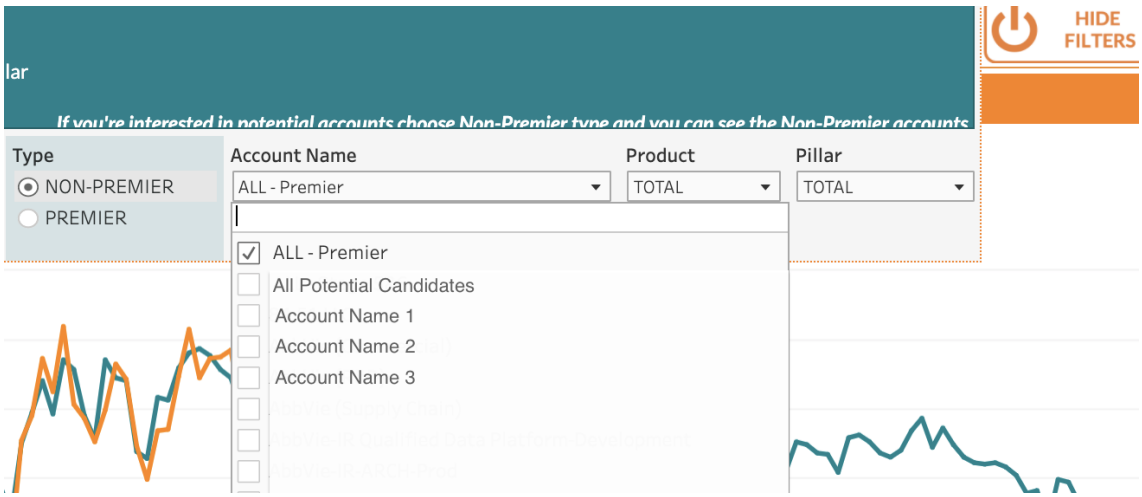


Figure 8.2: Tableau visualization - Account selection

A future development idea is to include the probabilities from the models' *predict_proba()* functions in a way in the visualization. This could be a new table with the account names linked with their likelihood or dividing and coloring the customer names based on their probability. The baseline for including clients in the volume prediction can be also changed.

Chapter 9

Summary

In my research, I presented a procedure based on data processing, analysis, and machine learning. I was looking for an approach to two similar but different classification problems to tel whether a customer is a candidate for the Premier or US Government support extension. The availability and the size of the data were diverse. Eight different models were built for each problem. By selecting the best-performing for each support group, I have two predictive models to decide whether a customer is eligible for the extended support groups and the sales should contact them. For these models, I had to go through all the machine learning steps.

I collected many important features for the machine learning models and after analysis, I created a clean and accurate data set which is essential for robust prediction. I learned that not all data set is ready for model building thus scaling and balancing are significant. I tried out two approaches for both problems: normalization and standardization for scaling, and the undersampling and oversampling for balancing the data. I asked the support leaders for their opinion about potential features and validated the results that I experienced with them.

For the machine learning problems, I tried out Random Forest Classifier, Gradient Boosting Classifier, Logistic Regression, and C-Support Vector Classification. These supervised learning algorithms are used for classification problems and are flexible with hyperparameter tuning opportunities. For both support groups, the Gradient Boosting algorithm performed better than the others. These predictions offer solutions to business situations as well because the companies could plan the future based on the possible account movements. My models could make a big impact on the company's headcount planning processes with the Tableau visualization.

I presented that these concepts are feasible, we can make correct predictions with the two models. One future improvement idea is to better visualize the likelihood of the customers.

Bibliography

[1] "Cloudera." *https://www.cloudera.com/.* Accessed: 2021-10-22.

[2] U. Shafique and H. Qaiser, "A comparative study of data mining process models (kdd, crisp-dm and semma)," *International Journal of Innovation and Scientific Research*, vol. 12, no. 1, pp. 217–222, 2014.

[3] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth, *et al.*, "CRISP-DM 1.0: Step-by-step data mining guide," *SPSS inc*, vol. 9, p. 13, 2000.

[4] R. Wirth and J. Hipp, "CRISP-DM: Towards a standard process model for data mining," in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, vol. 1, Springer-Verlag London, UK, 2000.

[5] F. A. Cummins, *Building the agile enterprise: With capabilities, collaborations and values.* Morgan Kaufmann, 2016.

[6] A. Erraissi, A. Belangour, and A. Tragha, "A big data hadoop building blocks comparative study," *International Journal of Computer Trends and Technology. Accessed June*, vol. 18, 2017.

[7] S. Goyal, "Public vs private vs hybrid vs community-cloud computing: a critical review," *International Journal of Computer Network and Information Security*, vol. 6, no. 3, p. 20, 2014.

[8] "Apache Impala." *https://impala.apache.org/.* Accessed: 2021-11-03.

[9] M. Bittorf, T. Bobrovytsky, C. Erickson, M. G. D. Hecht, M. Kuff, D. K. A. Leblang, N. Robinson, D. R. S. Rus, J. Wanderman, and M. M. Yoder, "Impala: A modern, open-source sql engine for hadoop," in *Proceedings of the 7th biennial conference on innovative data systems research*, 2015.

[10] "Cloudera Data Science Workbench Documentation." *https://docs.cloudera.com/cdsw/1.9.2/index.html.* Accessed: 2021-10-22.

[11] "Tableau." *https://www.tableau.com/*. Accessed: 2021-11-02.

[12] "What is CRM?." *https://www.salesforce.com/uk/learning-centre/crm/what-is-crm/*. Accessed: 2021-11-10.

[13] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.

[14] A. L. Duca, "How to balance a dataset in Python." *https://towardsdatascience.com/how-to-balance-a-dataset-in-python-36dff9d12704*, 2021. Accessed: 2021-11-14.

[15] K. Potdar, T. S. Pardawala, and C. D. Pai, "A comparative study of categorical variable encoding techniques for neural network classifiers," *International journal of computer applications*, vol. 175, no. 4, pp. 7–9, 2017.

[16] A. Bhandari, "Feature scaling for machine learning: Understanding the difference between normalization vs. standardization." *https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/*, 2020. Accessed: 2021-11-17.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[18] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowledge and information systems*, vol. 34, no. 3, pp. 483–519, 2013.

[19] A. Gupta, "Feature Selection Techniques in Machine Learning." *https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/*, 2021. Accessed: 2021-11-17.

[20] G. Bonaccorso, *Machine Learning Algorithms - Second Edititon*. Packt Publishing, 2018.

[21] C. Jiehong, S. Jun, Y. Kunshan, X. Min, and C. Yan, "A variable selection method based on mutual information and variance inflation factor," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, p. 120652, 2021.

[22] S. A. Ramcharan Kakarla, Sundar Krishnan, *Applied Data Science Using PySpark: Learn the End-to-End Predictive Model-Building Cycle*. Apress, 2020.

[23] Q.-Y. Dai, C.-p. Zhang, and H. Wu, "Research of decision tree classification algorithm in data mining," *International Journal of Database Theory and Application*, vol. 9, no. 5, pp. 1–8, 2016.

[24] U. Saralegui, *Occupancy Estimation and People Flow Prediction in Smart Environments*. PhD thesis, 09 2017.

[25] N. Donges, "A complete guide to the random forest algorithm," *Built In*, vol. 16, 2019.

[26] A. Liaw, M. Wiener, *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[27] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear estimation and classification*, pp. 149–171, Springer, 2003.

[28] C. Li, "A gentle introduction to gradient boosting." *http://www. ccs. neu. edu/home/vip/teach/MLcourse/4_ boosting/slides/gradient_ boosting. pdf*, 2016. Accessed: 2021-11-24.

[29] "Scikit-learn documentation about logistic regression." *https://scikit-learn.org/stable/modules/linear_ model.html/logistic-regression* . Accessed: 2021-11-24.

[30] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.

[31] A. Bora, "Introduction to support vector machines (svm)." *https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/* . Accessed: 2021-11-24.

[32] T. Shin, "Understanding the confusion matrix and how to implement it in python." *https://towardsdatascience.com/understanding-the-confusion-matrix-and-how-to-implement-it-in-python-319202e0fe4d*. Accessed: 2021-12-01.

[33] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve.," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[34] M. R. Segal, "Machine learning benchmarks and random forest regression," 2004.

List of Figures