University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering

# Bachelor's thesis

# Azure Machine Learning

Plzeň 2020 Irina Osovschi

**zde bude originál zadání**

# Declaration

I hereby declare that this bachelor's thesis is completely my own work and that I used only the cited sources.

Plzeň, 16th June 2020

Irina Osovschi

# Abstract

The purpose of this bachelor's thesis is to get familiar with a relatively new field of the modern world, machine learning. The thesis is divided into two parts - theoretical and practical. In the theoretical part, both the main machine learning approaches and the concept of Microsoft Azure tools are analyzed. The practical part contains an experiment built with the help of Azure tools. It consists of two models, each playing an important role in solving the problem of effective capacity planning of employees.



# Abstrakt

Cílem této bakalářské práce je seznámit se s relativně novým oborem moderního světa, strojovým učením. Bakalářská práce je rozdělena na teoretickou a praktickou část. V teoretické části jsou analyzovány hlavní přístupy strojového učení a koncept nástrojů Microsoft Azure. Praktická část obsahuje experimentální sestavení pomocí nástrojů Azure. Skládá se ze dvou modelů, z nichž každý hraje důležitou roli při řešení problému efektivního plánování kapacit zaměstnanců.

# Contents

# 1 Introduction

The aim of this work is to get started with machine learning using Microsoft Azure tools. The next chapter explains the connection between the human learning process, which is a common term for the reader, and the machine learning process, which is a modern term (Section 2.1). It contains the explanation of learning in terms of artificial intelligence and for a deeper understanding of it, it also covers some more general aspects, such as Behaviorism and Information Processing (Section 2.2).

Chapter 3 gets the reader familiar with the basic concepts and styles used in the machine learning field. It covers common algorithms used for each style and provides real-life examples. This chapter has a strong theoretical character and contains definitions and formulas.

Chapter 4 provides an overview of machine learning in Microsoft Azure tools. It explains the main concepts of working in Azure Cloud and provides the reader with a short guide of the software, that will be used later for the practical part, Machine Learning Studio (Section 4.2). This chapter also explains the process of machine learning itself (Section 4.3).

Chapter 5 represents the application of machine learning in practice. This chapter provides data analysis (Section 5.1 and 5.2), data pre-processing (Section 5.3) and the experiment built with the help of Microsoft Azure Studio software. Given a set of data that provides information about phone calls at CCA Group a.s. call center and historical dates, when the support was most needed, an experiment will be made. The experiment contains two models. The first model is a regression model and the second is an optimization model. Both of them were made to predict the optimal number of employees needed on a certain day.

# 2 Artificial Intelligence and Machine Learning

## 2.1 What is Machine Learning?

The learning ability is a vital characteristic of human intelligence. A few decades ago that was considered the only area where the term learning could be used. Today, we can firmly state and speak about artificial intelligence (AI) and its important characteristic - machine learning. The attribute artificial might awake different awareness. Speaking about intelligence it is hard to find an accurate definition. Is it about the amount of knowledge an individual possesses, or is it the capacity to react to our environment? Can we train our intelligence? Can we improve it? Scientists are still working on answering these questions.

With such different and yet not certain interpretations, it becomes difficult to define the term artificial intelligence. Nevertheless, I would like to quote and use in this work examples and historical definitions to characterize the field of AI. In 1955, John Mccarthy, one of the parents of AI, was the first to define the term of artificial intelligence as:

„The goal of AI is to develop machines that behave as though they were intelligent."

Now when we know what AI can be defined as, let's clarify the process of learning. Learning and methods used in training have always been an interesting field for psychologists to research in. In 1983, Harbert. A Simon, an American cognitive psychologist, writes an article where he defines learning as:

„A certain long-term change that the system produces to adapt to the environment and which leads to making the system finish the same or similar work next time more effectively."

The term Machine learning refers to the changes in systems that perform tasks associated with artificial intelligence. Such tasks involve prediction, planning, recognition, diagnosis, deduction, robot control, etc. The "changes" might be either improvement to already existing systems or building whole new systems. Nevertheless, forecasts or predictions from machine

learning make apps and devices smarter and represent an impressive added value for projects. [1]

## 2.2 Behaviorism and Information Processing

Behaviorism is a worldview that operates on a principle of stimulus-response. All behavior is caused by a force from outside. This force is called external. In other words, there is always a factor that causes a particular behavior.

The process of learning has a direct reference to the science of behavior. If the process is influenced by conditions, we call it learning by conditions. Let us provide a classic example from school. The classic example is the experiment made by the Russian scientist I.Pavlov in 1980. He predicted that if a bell is ringing every time before the dog is given food, in a while it will start to salivate just by hearing the bell. The more often this happens, the stronger the association becomes. This is called classical conditioning. This example illustrates what happens if an external factor keeps interacting with the system, in our case the dog is the system, the external force makes the system learn a new behavior.

In the middle of the '90s, behaviorism was slowly overtaken by information processing. The concepts are very similar, but it focuses on a more complex perception of operations. This approach was inspired initially from the information theory and the first computers that just made their appearance. The first mechanical computer was created by Charles Babbage in 1822, but the wide use of it as well as its features became popular in the 90'. In the theory of information processing, the stimulus is considered to be input, and the responses to be output. The most interesting and difficult part remains the process that converted the input to output. [4]

In terms of Machine Learning, the input represents the data set that we use as a learning source and the output is the goal that we want to achieve, in other words, the output varies based on what is the aim of the whole process. The process of converting the input into the output is known as a machine learning algorithm. It is not always easy to find the perfect algorithm, thus we often use more of them and then choose the one that fits better our requirements. The representation of what a machine learning system has learned from the training data is called a model. We will discuss more the Machine Learning process and other additional steps that are required for

creating a model in later chapters.

# 3 Styles of Learning

Machine Learning can be described by two different styles of learning. Generally speaking, the main two sub-fields of machine learning are supervised learning, also called predictive, and unsupervised, known as descriptive learning.

In supervised learning the focus is on accurate prediction obtained from finding a relation between the inputs $x$ and the outputs $y$ and forming input-output pairs:

$$D = (x^n, y^n), n = 1, \ldots, N \tag{3.1}$$

Here $D$ is called a training set for $N$ training examples.

Each training input $x_i$ is a $D$-dimensional vector of numbers. These are called features, attributes, or covariates. The form of the output, frequently can be found as a response variable, can be represented by anything, but most methods assume that $y_i$ is a categorical or nominal variable. When $y_i$ is categorical, the task is known as classification or pattern recognition, and when $y_i$ is nominal, we may also say real-value, the task is known as regression.

When speaking about unsupervised learning, the aim is to find how is our data described. This is a much less well-defined problem, since we do not know exactly what kind of patterns we are looking for and how different is the output, because we cannot compare our prediction with the $y_i$, unlike supervised learning does. In both cases, we are interested in methods to perform previously unknown data and behavior.

## 3.1 Supervised Learning

We will start with the most widely used in practice style of learning - supervised learning.

### 3.1.1 Definition

Given a set of data $D = (x^n, y^n), n = 1, \ldots, N$ the task is to learn the relationship between the input x and the output y:

$$f(x) = y \tag{3.2}$$

so next time when we are given a set of unseen $x^*$ we can use the function to predict the output $y^*$ as accurately as possible. The pair $(x^*, y^*)$ is not contained in the data set D, but we assume we used the same algorithm for finding it. To specify the abstract „as accurate as possible“ one defines a loss function $L(y_{pred}, y_{true})$ or a utility function $U$ (Section 3.2):

$$U = -L \tag{3.3}$$

In supervised learning our goal is to describe $y$ conditioned on knowing $x : p(y|x, D)$, where:

$$p(y|x) = \frac{p(y \cap x)}{p(x)}, p(x) > 0 \tag{3.4}$$

There should always be a „supervisor“, often called „teacher“, specifying the output $y$ for each input $x$ in available data D. [3]

### 3.1.2 Classification

Classification in machine learning and statistics is a supervised learning approach in which a system, using a classification algorithm, learns from the data given to it and makes new classifications. Classification is a process of typecasting a given set of data into classes. It can be realized in both structured or unstructured data. The process starts with predicting the class of given data inputs. The classification predictive modeling is the task of approximating the mapping function from input variables to discrete output variables. The main goal is to identify which class or category the new data belongs to.

We want to learn more about the outputs $y \in \{1, ..., C\}$, where $C$ is the number of classes. If $C = 2$, we are talking about binary classification. This means that our output is either in the first class or in the second. There is no chance of it being in another category. In binary classification, we often assume $y \in \{0, 1\}$. If $C > 2$, this is called multi-class classification. The

models obtained using multi-class classification are called multiple output models. [2]

For example, a classification algorithm will learn to identify sports cars after being trained on a data set of images that are properly labeled with the features of different types of cars and some identifying characteristics. A feature is an individual measurable property of the phenomenon being observed. For a car, it can be shape, speed, color, and others. In *Figure 2.1* and *2.2* we can see the example of sports cars. Another real-world example would be email spam filtering, where the classes are spam $y = 1$ or valid email $y = 0$, and $x$ is the text itself.

Figure 3.1: Some training examples of cars. Some of them are sports cars.

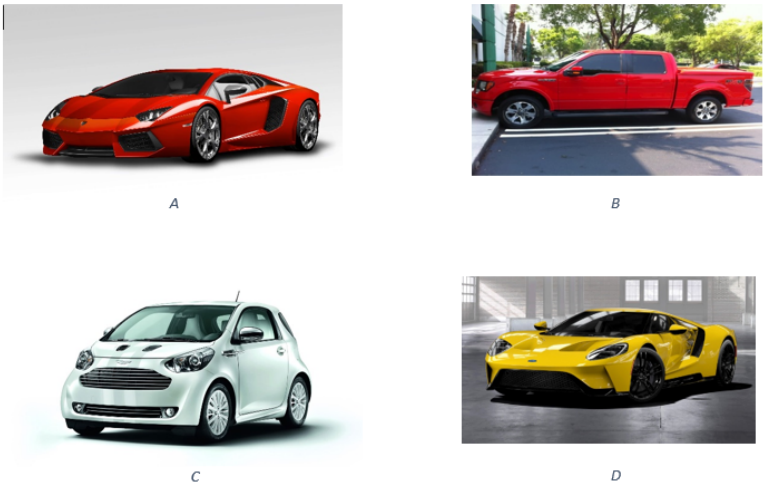



Figure 3.2: Representing the training data as as $N$ x $D$ matrix. Rows represent the feature vector $x_i$. The last column represents the label $y_i \in \{0, 1\}$, where 1 means sports car and 0 not sports car.

| *Model* | *Number of wheels* | *Colour* | *Speed* | *Label* |
|---|---|---|---|---|
| A | 4 | red | high | *1* |
| B | 4 | red | medium | *0* |
| C | 4 | white | low | *0* |
| D | 4 | yellow | high | *1* |

As mentioned before, classification is a supervised learning approach. There are many different machine learning algorithms for classification in machine learning.

### 3.1.3 Probabilistic prediction

It is important to distinguish prediction and classification. In many decision-making contexts, classification represents a premature decision. Sometimes the cost of the decision is higher than other times and in this case, it is always appropriate to also examine the problem from a probabilistic perspective. As it was mentioned above, it is important to denote the probability distribution over possible outputs, given the input vector $x$ and training set $D$ by $p(y|x, D)$. This means that the probability is conditional on the test input $x$ as well as on the training set $D$. When choosing between different models we will make this assumption explicit by writing $p(y|x, D, M)$, where $M$ denotes the model.

Given a probabilistic output, we can perform our "best guess" towards the real labels by using

$$y^* = f^*(\,x) \,= argmax(p(\,y = c|x, D)), c = \{1, .., C\}, \tag{3.5}$$

This corresponds to the most probable class label and is known as the mode of distribution $p(y|x, D)$ or MAP estimate. Using the most probable label gives us stronger and more plausible reasons for making the classification.

There are also cases when the value of $p(y|x, D)$ is far from 1, in this case, it is more relevant to say that we are uncertain and we cannot give a prediction because it could be wrong. This situation may occur when the risks of giving a wrong prediction are too high, in medicine for example. Here we rather say that we cannot predict and we either need more data to see if we can accomplish the prediction or find another way to solve the problem. [2]

### 3.1.4 Regression

Regression is a supervised learning method of modeling a target value based on independent predictors. This method is mostly used for forecasting and finding out the cause and effect relationship between variables. Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.

Regression is very similar to classification, the main difference is that the output is continuous. Many different models can be used. The simplest is linear regression. It tries to fit data with the best linear model which goes through the points. Most regression models propose that $Y_i$ is a function of $X_i$ and $\beta$, with $\varepsilon$ representing an additive error term that may stand in for un-modeled determinants of $Y_i$ [1]:

$$Y_i = f(X_i, \beta) + \varepsilon \tag{3.6}$$

Regression analysis is a form of predictive modeling technique that investigates the relationship between a dependent (target) and the independent variable (predictor). This technique is used for forecasting, time series modeling, and finding the causal effect relationship between the variables. Using regression is effective because it indicates the significant relationships between the dependent variable and the independent variable. It also indicates the strength of the impact of multiple independent variables on a dependent variable. An algorithm that is capable of learning a regression predictive model is called a regression algorithm. [2]

A real-world example where regression can be used is temperature prediction at any location inside a building, using weather data, time, sensors, etc.

There are different types of regression models. When choosing we want to pick the one that suits our data the best. Later in this work, we will make a Poisson regression model. Now we will examine the algorithm from the inside.

### Poisson Regression

Poisson regression is used to model count data, assuming that the label has a Poisson distribution. Which is perfectly suitable to predict the number of calls to a customer support center on a particular day.

As already mentioned before, Poisson regression is meant to be used in regression models that aim to predict numeric values, frequently counts. Hence, in machine learning, it is reasonable to choose this approach for our regression model if the values we are trying to predict satisfy the following conditions:

- The response variable has a Poisson distribution.

- The response variable is zero or positive. The algorithm will not succeed if it is used with negative labels.
- The inputs should be whole numbers because Poisson distribution is a discrete distribution.

For this algorithm, it is assumed that an unknown function, denoted $Y$, has a Poisson distribution and assumes the logarithm of its expected value can be modeled by a linear combination of unknown parameters. Poisson distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known constant mean rate and independently of the time since the last event.

A discrete random variable X is said to have a Poisson distribution with parameter $\lambda > 0$, if, for $k = 0, 1, 2, ...$, the probability mass function of X is given by:

$$f(k, \lambda) = P(X = k) = \frac{x^k e^{-\lambda}}{k!} \tag{3.7}$$

When given the instance $X$, for every $k = 0, 1, \ldots$ the module computes the probability that the value of the instance is $k$.

In terms of Poisson regression, if $x \in R^n$ is a vector of independent variables, then the model takes the form:

$$E(Y|X) = e^{\alpha+\beta' x} \tag{3.8}$$

$$E(Y|X) = e^{\Theta' x} \tag{3.9}$$

where $\Theta$ is simply $\alpha$ concatenated to $\beta$.

Given the set of training examples, the algorithm tries to find the optimal values for $\Theta$ by trying to maximize the logarithmic likelihood of the parameters(MLE). The likelihood of the parameters $\Theta$ is the probability that the training data was sampled from a distribution with these parameters. The prediction function outputs the expected value of that parameterized Poisson distribution. [7]

## 3.2 Utility and loss

The optimal prediction of $y$ depends on how costly making an error is. Here comes the loss function $U$. In forming a decision function $c(x^*)$ that will produce a class label, the predictive distribution $p(c|x^*)$. If $U(c^{true}, c^{pred})$

represents the utility of making a decision $c^{pred}$ when the real value is $c^{true}$, the expected utility for the decision function is:

$$U(c(x^*)) = \sum_{c^{true}} U(c^{true}, c(x^*)) \cdot p(c^{true}|x^*) \tag{3.10}$$

and the optimal decision function $c(x^*)$ is that which maximises the expected utility,

$$c(x^*) = argmax(U(c(x^*))) \tag{3.11}$$

Equivalently, we consider a loss $L(c^{true}, c(x))$, for which the expected loss with respect to $p(c, x)$ is known as the risk. The optimal decision function is then that which minimizes the risk to 0. [3]

## 3.3 Unsupervised Learning

When speaking about unsupervised learning we consider that we are just given the output, without any inputs. The aim here is to discover interesting structure in the data. Unlike supervised learning, we do not know what is the desired output from each input. Unsupervised learning is considered to be more animal and human learning alike and is also more applicable since it does not require a human expert to supervise it. Instead, we allow the model to work on its own to discover information.

### 3.3.1 Definition

Given a set for data $D = x^n, n = 1, \ldots, N$ in unsupervised data we aim to find a plausible description of the data. We want to model the underlying structure or distribution in the data $p(x)$ to learn more about it. The likelihood of the model to generate the data is a popular measure of the accuracy of the description.

It is called unsupervised learning because unlike supervised learning above there is no correct answer and there is no teacher. Algorithms are left to their devices to discover and present an interesting structure in the data. [3]

### 3.3.2 Clustering

Clustering is one of the main two types of unsupervised learning. The second type is association. Clustering deals with finding a pattern in unstructured data and split it into groups. These groups are often called clusters. Unlike supervised learning methods, the number of clusters can be chosen. One technique used for cluster creation is the probabilistic technique. It is based on probability distribution. A clustering algorithm identifies which cluster each output can be included in.

If data is grouped in such a way that every data belongs to one cluster only, we call this exclusive clustering. In this case where every single output defines a cluster we often use an algorithm called agglomerative clustering. It copes with iterative unions between the two closest clusters to reduce the number of clusters. [2]

Clustering itself is not one defined algorithm, but the general and complex problem that needs a solution. The solution can be obtained by various algorithms that are considerably different in terms of basic concepts of what should a cluster include and the logic of how to find them. This is why clustering is often formulated as a multi-objective optimization problem. The chosen algorithm depends a lot on every data set and more importantly on the general goal. Currently, there are known more than 100 published clustering algorithms. Cluster analysis requires a case to case approach, therefore it involves trials and failures till the best algorithm is found. Like many other types of learning, clustering algorithms require data pre-processing until the result achieves the desired properties. [7]

A real-life example where clustering is used in commerce and advertising. For instance, by using data about customers purchasing and their web behavior, an e-shop can identify one customer's shopping preferences group and use this categorization for advertising products that are more likely to be bought by them.

### 3.3.3 Association

Association is the second main type of unsupervised learning. It mainly consists in discovering interesting relationships between data in a large data set. These are called associations. This method is widely used in the market-basket analysis. The data used in this method typically consists of a matrix, where columns represent items and rows represent the time of transaction.

Many items are purchased together, so there is a correlation between them. The goal here is modeling purchasing patterns.

The task of association rule mining is defined as: If $I = \{i_1, i_2, .., i_n\}$ is a set of $n$ binary attributes called items and if $T = \{t_1, t_2, \ldots, t_m\}$ is a set of transactions, also known as database. Each transaction in $D$ has an individual transaction ID and contains a subset of the items in $I$. Each transaction is described by an association rule:

$$X \Rightarrow Y; X, Y \subseteq I \tag{3.12}$$

Confidence describes how often the rule has been found to be true. The confidence value of a rule, $X \Rightarrow Y$ , considering the set of transactions $T$, is the proportion of the transactions that contain $X$ which also contain $Y$.

$$conf(X \Rightarrow Y) = \frac{P(E_x \cap E_y)}{P(E_x)} \tag{3.13}$$

Here $E_x$ describes the event that transaction includes item $x$ and $E_y$, respectively, is the event that transaction includes item $y$.

The lift of a rule helps us in judging if $X$ and $Y$ are independent. It is defined as:

$$lift(X \Rightarrow Y) = \frac{P(E_x \cap E_y)}{P(E_x)(E_y)} \tag{3.14}$$

If the lift value is equal to 1, it implies that the events are independent of each other. When two events are independent of each other, it means that no rule that involves those events can be made.

If the lift value is $> 1$, that lets us know the degree to which those two occurrences are dependent on one another, which makes those rules potentially useful for predicting in the date set.

If the lift value is $< 1$, that lets us know the items represent substitutes for each other. In other words, this means that the presence of one item has a negative effect on the presence of other items and vice versa. [8]

A real-life example of association usage is movies association. It can be met in different streaming portals. The association rules are made based on a customer's movie view history. It makes it possible to predict which movies may be interesting for the customer.

## 3.4 Semi-supervised Learning

Problems where we have a large amount of input data (X) and only some of the data is labeled (Y) are called semi-supervised learning problems. As its name implies, semi-supervised learning falls between unsupervised learning and supervised learning. Unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

Unlabeled data is often cheap in obtaining, it doesn't require much effort, while labeled data is more expensive because it often requires human supervising. In machine learning, it is a common scenario. Semi-supervised learning is also considered the closest model for human learning.[3]

A good example is a photo archive where only some of the images are labeled, (e.g. dog, cat, person) and the majority are unlabeled. For solving these problems we use a combination of supervised and unsupervised learning methods.

## 3.5 Interacting with the environment

In many situations, a system needs to interact with the environment and therefore it should be able to behave in a different way than it was supposed initially. This interaction complicates the learning process, but at the same time once it is overcome it highly enriches the potential for learning.

Query (active) learning gives the ability to request more data from the environment. At the moment, when the agent recognizes that it is less confidently able to predict based on certain x, it requires more training data closed to the interval where x is located. In other words, it may be considered in an unsupervised context in which the system might request information in which p(x) is not informative.

Reinforcement learning is about inhabiting the environment in which it may take action. The side effects are that some actions may be efficient, while others are not. Based on the past and accumulated experiences, the system learns how to behave in a certain situation to maximize the probability of obtaining an effective and long-time lasting goal. Reinforcement learning has connections with control theory, Markov decision processes, and game theory. We will not discuss these terms further, but refer the reader

to the specialized text. [9, 10, 11]

# 4 Azure Machine Learning

Azure Machine Learning is a cloud-based environment used for training, deployment, automatizing, management, and tracking of machine learning models. One of the biggest advantages it gives to its users is the possibility to be used for any kind of machine learning. Living in the big data era, it is valuable to be able to use machine learning for our projects. Azure provides tools that can be used not only by professional data scientists.It means that with basic knowledge and understanding of probability and data processing even a non professional data scientist can create models.

## 4.1 What is Azure?

Azure is a cloud computing platform and an online portal that manages cloud services and resources provided by Microsoft and allows access to them. These services and resources include storing data and transforming it, depending on our requirements. To get access to these resources and services, all we need to have is an active internet connection and the ability to connect to the Azure portal.

It's free to start and follows a pay-per-use model, which means we pay only for the services we opt for. The portal requires having a Microsoft Azure account. Making an account is as easy and accessible as making a usual account in the space of the internet.

One of the major fields that Azure portal provides to its users is Azure Machine Learning. The Microsoft Azure Machine Learning offers an array of tools and services, including :

- Azure Machine Learning Workbench - an application that handles primary tasks for a machine learning project, including data import and preparation, model development, experiment management, and model deployment in multiple environments.
- Azure Machine Learning Experimentation Service - a service that helps connect Workbench, project management, access control, and version control.

- Azure Machine Learning Model Management - a service that helps manage, store, register, and process models.
- Microsoft Machine Learning Libraries for Apache Spark - a set of tools that combine Spark pipelines and machine learning tools.
- Visual Studio Code Tools for AI - a desktop editor used for writing scripts for machine learning experiments.
- Azure Machine Learning Studio:- a tool that helps create and deploy predictive analysis models. [18]

## 4.2 Machine Learning Studio

Azure Machine Learning Studio is a tool, provided by Microsoft, that is used to build, test, and deploy predictive analytic models. It helps in solving analytical problems that come with a set of data. By using it, we are offered the possibility to transform, analyze data, proceed with different data manipulation, and generate results.

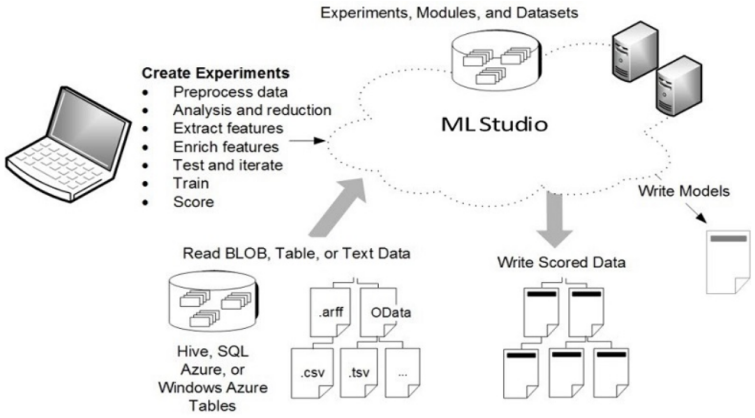
To develop a predictive analysis model, it is important to have data. The tool provides a large number of data sets or we can upload our own set from the local machine. If needed, the data can be transformed based on the experiment's requirements. The tool contains a set of modules to analyze data and generate results.

Generally, Machine Learning Studio contains an interactive and visual work-space, which makes it easy to build, test, and manipulate with the models. It is based on dragging and dropping different modules and connecting them. Depending on the module, we can manipulate with a different amount of ports. However, most of them are supposed to be already set as needed. At the same time, each module is under the user's control. It is possible to manipulate all the parameters of the module. Moreover, if the module itself supports a middle result of the data, the user has access to visualization in that certain module and step of the experiment. The main concepts visualization is *Figure 4.1* [16]

### 4.2.1 Machine Learning Studio experiment

An experiment consists of data sets and modules. To create a predictive analysis model we need to connect them in the most meaningful way.

Figure 4.1: Main concepts of Machine Learning Studio




### Data set

A data set is a data that has been uploaded to Machine Learning Studio so that it can be used in the modeling process. The software provides us two options: we can either use data set samples, this is mostly used for training and getting to know-how skills in the software, or upload a data set from our local machine. When uploading a data set, we can rename it, add a description, and choose the format in case the software didn't identify it.

### Modules

A module is an algorithm that can be performed on data. Azure Machine Learning Studio has several modules varying from statistical functions to training, scoring, and validation processes. During the experiment creation, we can choose from the list of modules available on the left of the canvas.

A module may have a set of parameters that we can use to configure the module's internal algorithm. Based on our project requirements, we can modify the parameters in that pane to personalize our model [16].

An experiment is considered to be valid and correctly made if it fulfills all of the following conditions:

- every experiment has at least one data set and one module
- data sets are connected only to modules
- modules can be connected either to data sets or to other modules
- all input ports should have a connection to data flow

- every module has all the required parameters set
- there are no errors during the compilation

### 4.2.2 Machine Learning Studio web service

Machine Learning Studio web service enables us to deploy the predictive analytic solution as a web service. It connects external applications with the Machine Learning Studio workflow scoring model in real-time. A call to a Machine Learning Studio web service returns prediction results to an external application. To make a call to a web service, we need to pass an API key that was created when we deployed the web service. To deploy the model all the next steps are required:

- Create a training experiment
- Convert it to a predictive experiment
- Deploy a web service

In other words, after creating and deploying the web service an API command becomes available. It is a command of post type that requires attributes before calling it. The type and number of attributes depend on the experiment itself. Once the azure web service is deployed, it is possible to send requests to the service and receive responses. [16]

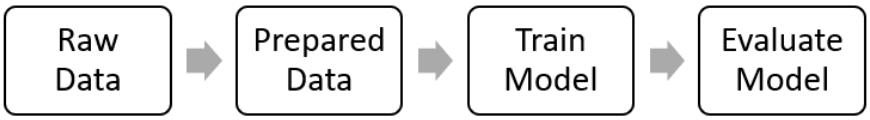
## 4.3 Machine Learning Process

The machine learning process is complex. It consists of 4 major parts. Before starting a machine learning process we need to have the adequate data sets collected. These sets are frequently called raw data. The next step is pre-processing. This step is highly important because data is not always clean. It may contain some missing values or values that are irrelevant to our experiment. Also at this point we divide our data into two parts, one will serve for working the algorithm and the second part will be used for testing the obtained model. After filtering the data, a Machine Learning Algorithm needs to be chosen. It is not always easy to find the perfect algorithm. Thus, we usually try different algorithms on the same data, and then we compare the outputs to find the one that satisfies best our needs. The next step is building the model by validating and testing the model. This is a process that requires more iterations. The final step is deploying the model, after proceeding this step our model will be able to predict. *Figure 4.2* is a visual

representation of the process.

Azure Machine Learning Studio is an important resource in terms of speeding up the whole machine learning process. It provides a set of implemented tools that help realize all the steps mentioned above with a higher efficiency.

Figure 4.2: The Machine Learning process.



# 5 Application of Machine Learning in practice

As it was mentioned before, the main aim of this work is to create a model for the effective capacity planning of employees using the principles of machine learning. First of all, data analysis will be executed, following all the required steps. The goal is to create and deploy a predictable model that will be able to tell us the optimal number of employees working at calling support center in CCA Group a.s. company. The problem becomes more complex due to the time when software distribution occurs. By software distribution time, we understand the time when the company is releasing a new product version, thus the requirement for customer support increases. The result should be a model, which will be capable to tell us the optimal number of employees needed in a specific interval of time so that the company's productivity and the added value for its customers is at it's maximum. For achieving all the goals mentioned before, we will use Azure Machine Learning Studio.

## 5.1 The process of distribution

Distribution time is the time when the final product gets to the customer. It is considered to be part of the software creating process. The process itself consists of more steps and distribution happens to be the final one. For a deeper understanding of our problem, it is necessary to briefly pass through all the main phases.

The first and basic phase is developing the product. It can be either a brand new software made to fulfill customer demand, or it can be a new version or update of an existing product. Normally, a software product consists of more than one system and application. In this work, we will use the general term software product, but we should keep in mind that it is a complex term that consists of many other central and local systems.

During the development, the product is being partly tested by a testing team, but the main work for this team comes once the product is in its final developing state. Here starts the testing. It can take a different amount of time and it consists of finding bugs in both front-end and back-end develop-

ment of the system. Depending on how many bugs were found and on how quickly the developing team reacts to solving these problems, the testing period may take a different amount of time.

Once the product is tested and ready to be released, the software distribution time comes. It is that time when the product reaches its final customer. It includes the time the customer needs to install and launch the software. Normally, it takes 1-2 days and is followed by a higher demand for customer support. Thus, the distribution process can be defined as a time, described by software release, which causes a higher demand for customer support.

When a customer requires technical support the reasons may be either technical, which means that there are difficulties during the installation, or there are certain problems in the application itself. Whether it is the first type of problem or the second, the call center employees provide help and support. In practice, it happens a lot that all the phone lines are busy and the customer has to postpone his call to another time. We will dive deep into investigating historical data regarding employee attendance and call reports in later paragraphs. A second way of reporting a problem is by using e-mails or leaving a report. We will not cover help-desk reporting in this work.

The product may be distributed to a large company with thousands of employees as well as to a couple of people. In both cases the number of phone calls is similar. This phenomenon is explained in the following way. If we speak about a big company with a lot of people that use the software $x$, there is always somebody responsible for solving the technical problems, in other words, not every person is likely to call and ask for support, but rather the people that are responsible for this. This means that for training our model there is no difference between the customers that receive the product.

The product reaches the final customer together with documentation. The documentation also contains a know-how chapter, that describes all the steps for installing the software. In other words, the customers are responsible for the accuracy of the installation.

## 5.2 Data used for learning

For accomplishing the goal of this work we will use a supervised learning approach. The output of the learning depends on more inputs. We will take into consideration those that directly affect the final result. CCA Group a.s. offered us a set of raw data that tracks employees' activity at work as well as important dates such as distribution dates.

### 5.2.1 Distribution dates

Given a set of data, which contains information about dates, hours, and activities in CCA Group a.s., our primary goal is to find out when did the distribution process occur in previous dates and find if there is any relation between them.

Data contains information about distribution dates starting from the year 2013 till the beginning of the actual year, 2020. After a deeper analysis of it, a clear pattern was observed. The distribution process occurs 4 times per year, with an average interval of 3 months. Starting from 2013, distribution starts at the beginning of a season: March, June, September, and December. Then it lasts from 2 to 3 days. The exact date varies from year to year, but the difference is small, just a couple of days. It never happens to be more than one distribution at the same time, so there is no chance of collision between them.

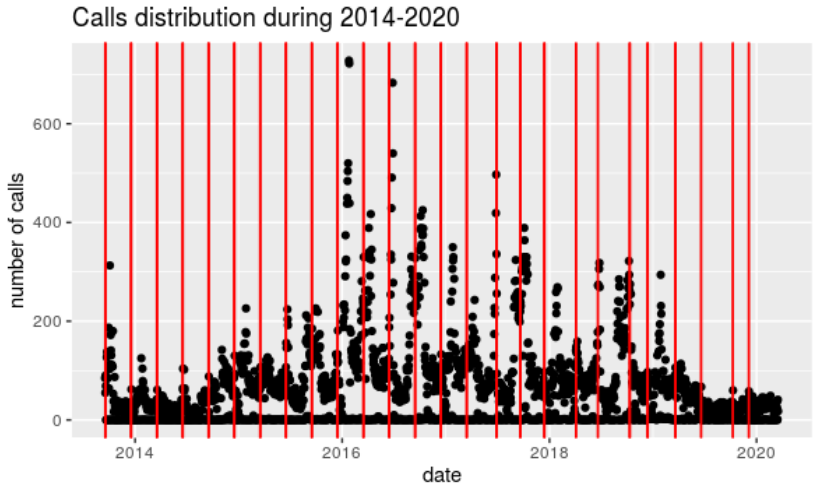
In 2018 a late distribution occurred twice. The date and month changed from March to early April and from September to early October. Considering this deviation from the usual behavior, the pattern is still valid, this means there were still 4 distribution dates in this year with an average interval of 3 months.

On the first day of distribution clients' service is busier from 6:00 to 21:00, and the second day has a busy line from 6:00 to 8:00. As CCA Group a.s. employees affirm, the first day seems to be always fully busy with reports and calls, then the demand for customer's support declines.

### 5.2.2 Human resources

Not every employee is responsible for providing customer support. Given a data set of employees code and their job position, we chose those who provide

Figure 5.1: Graphical representation of calls distribution in 2013-2020.




customer support. The employee $x$ may be a developer, but also provides technical support when a call or email comes. At present, 46 people are responsible for support and the usual length of a workday is 8 hours.

### 5.2.3 Registered phone calls

Data that contains information about registered phone calls at the customer support center represents the most valuable source for our experiment. It gives us information about the date, start time, end time, and if any calls were missed. It seems to be difficult to predict this information in the future, but when analyzed carefully, we can recognize a pattern that is repeated regularly after a certain time. In the period of software distribution, people give phone calls with a higher frequency than in periods when no new updates arrive, which directly affects the employee's capacity needed. In *Figure 5.1* we visualize the whole data set.

We know by now from historical data in the period 2013-2020 that the phone line is busier on the first day, as the calls are coming from 6:00 until 21:00, and less busy the following day calls coming from 6:00 until 8:00. There are also exceptions when customers call later, but we will not consider these exceptions as an important factor that could change the output because when this phenomenon happens, there is no need for a higher costumers support.

Another abnormal fact that happens to the frequency of the calls is that close to the release date there is a big increase in the number of calls. This

may occur because before the release day the new product is sent to a testing organization. This means that the call frequency gets higher because of dealing with the testing organizations.

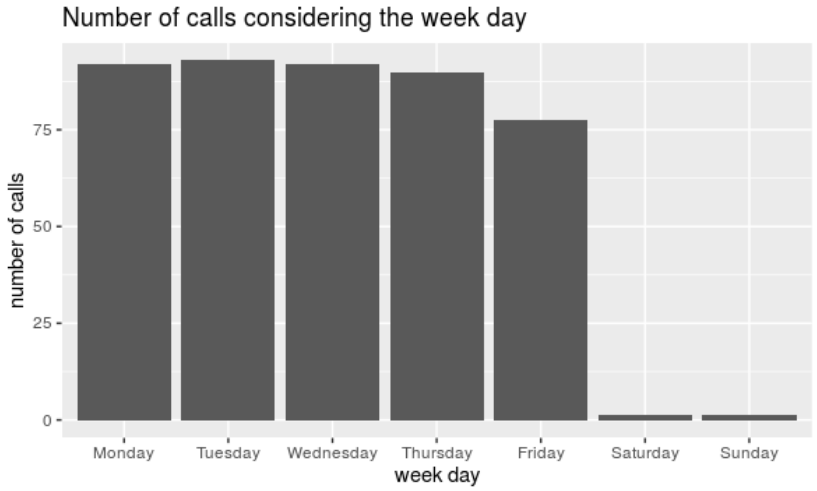
The time needed to solve a customer problem varies. It can be influenced by the complexity of the report and the level of customer knowledge in the IT field. But we will not consider this variation when training the model for two reasons. The first reason is that we do not have enough data for stating the exact length of a call and second is that it does not have a major impact on our prediction.

## 5.3 Data pre-processing

Before uploading the source file to Machine Learning Studio it is necessary to achieve an appropriate format. This means that the first step is data pre-processing. For drawing the graphs, the R programming language was used. All the scrips are attached as sources.

First of all, it is necessary to make days categorization. We will examine every day of the week from Monday to Sunday based on the number of calls per day. Of course, all the results represent an overall average. In *Figure 5.2*, we can see that the days Monday, Tuesday, Wednesday, and Thursday represent one category. This means that there is no difference from a practical and demanding perspective for our future model between these days. Friday represents another category and weekend days, Saturday and Sunday, a different category. From now we will consider 3 categories of days. This information will be used later for making our regression model.

Figure 5.2: Days categorization graph

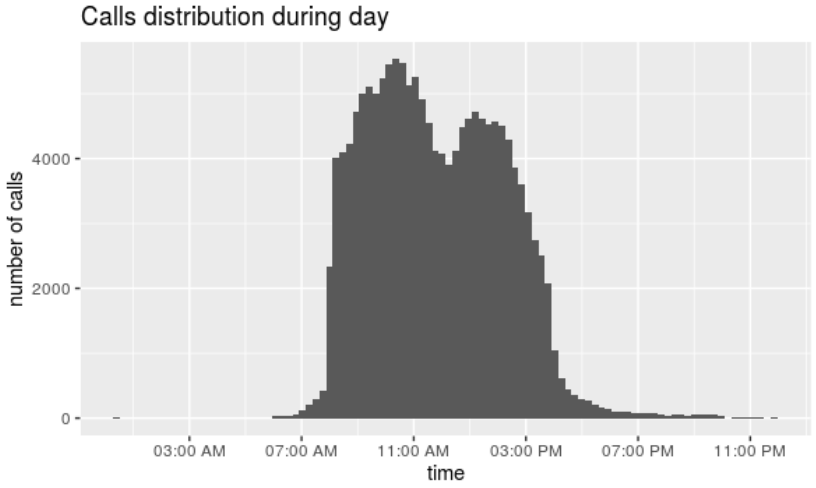



For making our future model more precise we will also take into consideration holidays. A list of all the days off staring from 2013 until 2020 was made and included as another raw data set. All of the following are considered holidays: Christmas, New Year, and Easter. Holidays will be included in the third category, it means that call frequency during holidays is equal to calls frequency during the weekend days.

In *Figure 5.3*, we can see the average calling intensity during a day. As expected, it is almost uniform for the whole day. There is a small gap in the

middle of the day. This gap may be caused by launch break. Because the gap is not considered big, we assume that the distribution of calls during a day is uniform.

Figure 5.3: Calls distribution graph




Another important label that will serve as a day attribute in our experiment is the number of days since release. We already know that software releasing is happening on average every 3 months and we know that more calls are coming in the next few days after the release. Within one week after the release the number of calls per day tends to decrease. But it is important to take into consideration that it is not a constantly decreasing function. After some time, the number of calls stop decreasing and stay at a normal rate. Therefore, we will categorize days by the number of days passed since released from 1 to 20. After 20 days all the days will have similar behavior. With these being said, another call's feature is the number of days after release, and it can have a value from 1 to 20.

Now when we have identified the important features of a call, we can label our data by these features and we can upload the data set to Machine Learning Studio. *Figure 5.4* is the visualization of the resulted data set.

Figure 5.4: Data set visualization in Machine Learning Studio

Num_calls prediction [Predictive Exp.] › daily_calls.csv › dataset

rows 2374 columns 10

view as

| date | num_calls | weekday | month | is_release | is_holiday | days_since_release | days_since_release_cap | release_decay | day_cat |
|---|---|---|---|---|---|---|---|---|---|
| 2013-09-17T00:00:00 | 85 | Úterý | září | 1 | 0 | 0 | 0 | 1 | Working days w/o Friday |
| 2013-09-18T00:00:00 | 55 | Středa | září | 1 | 0 | 0 | 0 | 1 | Working days w/o Friday |
| 2013-09-19T00:00:00 | 90 | Čtvrtek | září | 1 | 0 | 0 | 0 | 1 | Working days w/o Friday |
| 2013-09-20T00:00:00 | 69 | Pátek | září | 0 | 0 | 1 | 1 | 0.9 | Friday |
| 2013-09-21T00:00:00 | 0 | Sobota | září | 0 | 0 | 2 | 2 | 0.81 | Weekend / Holiday |
| 2013-09-22T00:00:00 | 0 | Neděle | září | 0 | 0 | 3 | 3 | 0.729 | Weekend / Holiday |
| 2013-09-23T00:00:00 | 127 | Pondělí | září | 0 | 0 | 4 | 4 | 0.6561 | Working days w/o Friday |
| 2013-09-24T00:00:00 | 138 | Úterý | září | 0 | 0 | 5 | 5 | 0.59049 | Working days w/o Friday |
| 2013-09-25T00:00:00 | 84 | Středa | září | 0 | 0 | 6 | 6 | 0.531441 | Working days w/o Friday |
| 2013-09-26T00:00:00 | 130 | Čtvrtek | září | 0 | 0 | 7 | 7 | 0.478297 | Working days w/o Friday |
| 2013-09-27T00:00:00 | 85 | Pátek | září | 0 | 0 | 8 | 8 | 0.430467 | Friday |
| 2013-09-28T00:00:00 | 0 | Sobota | září | 0 | 0 | 9 | 9 | 0.38742 | Weekend / Holiday |
| 2013-09-29T00:00:00 | 0 | Neděle | září | 0 | 0 | 10 | 10 | 0.348678 | Weekend / Holiday |
| 2013-09-30T00:00:00 | 187 | Pondělí | září | 0 | 0 | 11 | 11 | 0.313811 | Working days w/o Friday |
| 2013-10-01T00:00:00 | 64 | Úterý | října | 0 | 0 | 12 | 12 | 0.28243 | Working days w/o Friday |
| 2013-10-02T00:00:00 | 112 | Středa | října | 0 | 0 | 13 | 13 | 0.254187 | Working days w/o Friday |
| 2013-10-03T00:00:00 | 103 | Čtvrtek | října | 0 | 0 | 14 | 14 | 0.228768 | Working days w/o Friday |

## 5.4 Regression Model

### 5.4.1 Training model

The whole experiment is based on a regression model. Making a regression model in Machine learning Studio requires a few steps. First of all, we need to upload a data set. In previous chapters, we talked about the consistency of this data set. Then it is needed to make some data manipulations. From the list f modules available to the left of the canvas we choose columns selector module. It is necessary to pick the relevant columns. When choosing we pick only the columns that affect the capacity utilization of the employees and are important for training our model. Relevant columns are days categories, number of days since release, number of calls, and month. All the other columns will not affect the regressing model.

The next step is splitting the data it two parts. We choose the split data module and fill in the required parameters. With a 0.7 fraction, we split the whole data set. This means 70% is used for training the model and the other 30% is used for scoring the model. The split data experiment module has two output ports. The output from the first port goes for training and the second one for scoring.

Choosing the machine learning algorithm is one of the most important steps. Machine learning studio offers a bunch of different algorithms, both for supervised and unsupervised learning. For our experiment, we choose

supervised learning and regression algorithms. Because we deal with count data the only meaningful regression algorithm that we can opt for is Poisson regression. Poisson regression represents another module that we choose and drag into our experiment. After choosing it, it requires to either fill in or leave the default values of the regression parameters. As mentioned earlier, we use regression because we are trying to predict a continuous variable.

Now with the help of the chosen regression algorithm and 70% of data we train the model. Training a model requires setting the label column which is at the same time the result that we are interested in. In our case, it is important to find out the number of calls. Therefore we choose the column that indicates the number of calls.

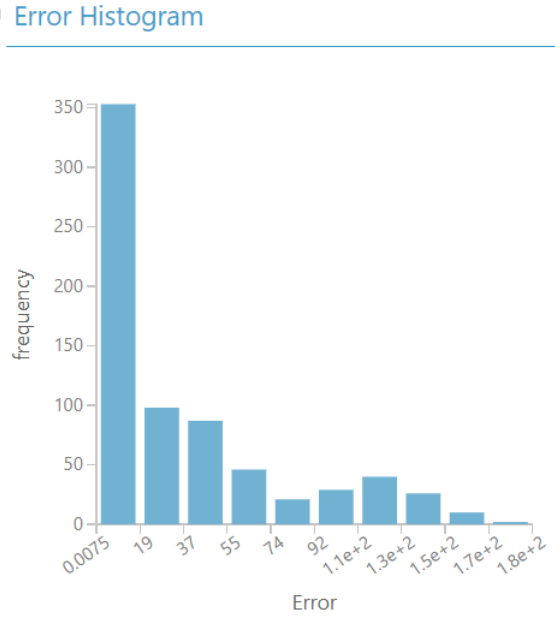
Once the model is trained, it is time to score it. For scoring the model we use the rest 30% of the data that is left and the trained result itself. A table that visualizes the scoring is in *Figure 5.5*. The first column *num_calls* represents the actual number of calls registered and the column *Scored Lables* represents the predicted value by the earlier trained model. The predicted values are close to the real values. This is a sign that the algorithm was chosen right.

Figure 5.5: Scored labels visualization in Machine Learning Studio

| num_calls | month | days_since_release_cap | day_cat | Scored Labels |
|---|---|---|---|---|
| 136 | dubna | 1 | Friday | 126.198024 |
| 7 | dubna | 2 | Weekend / Holiday | 1.691649 |
| 6 | dubna | 3 | Weekend / Holiday | 1.669836 |
| 159 | dubna | 4 | Working days w/o Friday | 146.192371 |
| 135 | dubna | 5 | Working days w/o Friday | 144.307284 |
| 122 | dubna | 6 | Working days w/o Friday | 142.446497 |
| 101 | dubna | 7 | Working days w/o Friday | 140.609728 |
| 116 | dubna | 8 | Friday | 115.238471 |
| 2 | dubna | 9 | Weekend / Holiday | 1.544739 |
| 0 | dubna | 10 | Weekend / Holiday | 1.52482 |
| 102 | dubna | 11 | Working days w/o Friday | 133.496426 |
| 78 | dubna | 12 | Working days w/o Friday | 131.775058 |

The last step in building the model is evaluating. After evaluation, we can state the errors. The resulted mean absolute error (MAE) is 37.0498, the Root Mean Squared Error (RMAE) is 56.759342. Error histogram visualization is in *Figure 5.6*. Overall, 353 scored items, that represent 50% out of the total amount of data provided for testing, has an error value in over 0.0075 and below 19. The final training experiment can be seen in *Figure 5.7*.

Figure 5.6: Error histogram visualization in Machine Learning Studio

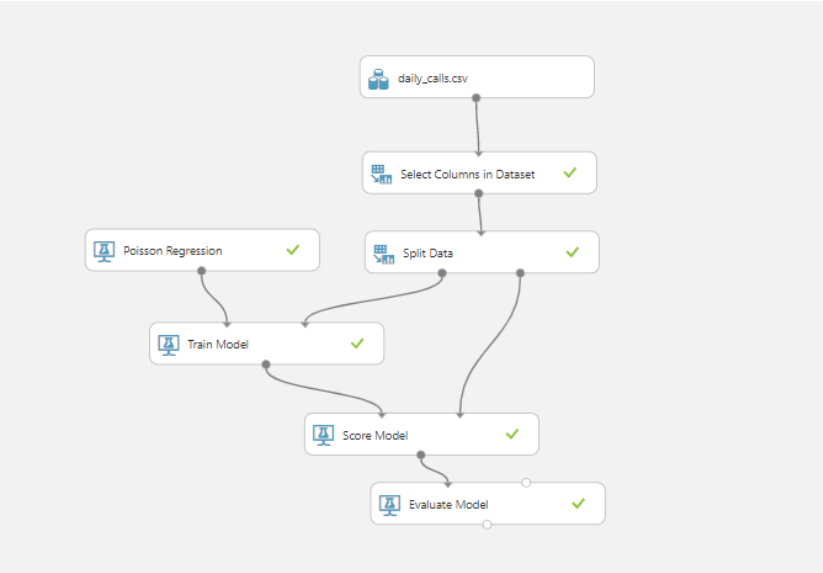



### 5.4.2 Predictive model

For creating a predictive model we need to deploy Azure web services. When we convert the training experiment to a predictive experiment, some of the modules are no longer needed. We have already used a couple of modules to train the model, these modules are replaced with a single module that contains the model we trained. This new module is saved in the Trained Models section of the module palette.

Also, we need to adjust the input and output modules. The input module represents the parameters of the future request and outputs the final result. The input data provided through the web service will now pass directly into the Score Model module without any pre-processing. As a web service output, we consider the number of calls. It means that we need to launch a

Figure 5.7: Training experiment in Machine Learning Studio




new module called column selector and choose the scored labels column.

Now that the predictive experiment has been prepared (*Figure 5.8*), we can deploy it as a new Azure web service. Using the web service, a user can send data to the model and the model will return its predictions.

### 5.4.3 Web Service Deployment

Azure Machine Learning Studio enables us to build and test a predictive analytic solution. Then we can deploy the solution as a web service. Azure provides two types of web services: Request-Response service and Batch Execution service. For our aim, we use a Request-Response service. A call to web service returns predictions. To make a call, all we need to do is to pass an API key. This key is created in the moment of web service deployment.

Once we have the predictable experiment we can deploy the web service just by clicking this option in the tools panel. At this point, we can get predictions by testing the experiment. For testing, we can either enter input values from the list view mode or upload as an input a CSV file containing the values. To upload CSV files an upgraded account is required. For our learning purposes, we will keep with the simple list view mode testing as it is for free (*Figure 5.9*).

Figure 5.8: Predictive experiment in Machine Learning Studio

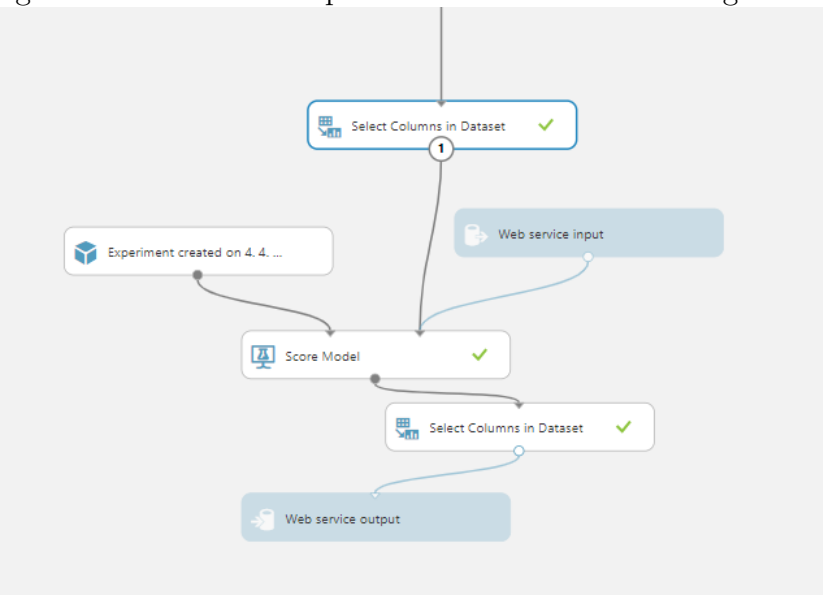
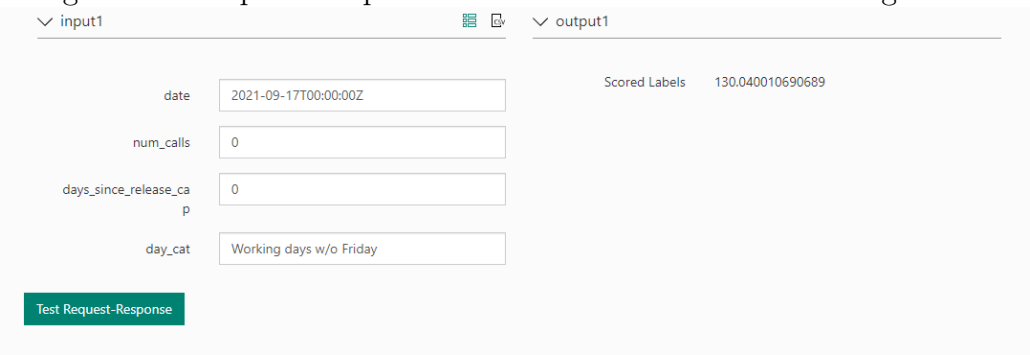



Figure 5.9: Request-Response web service in Machine Learning Studio



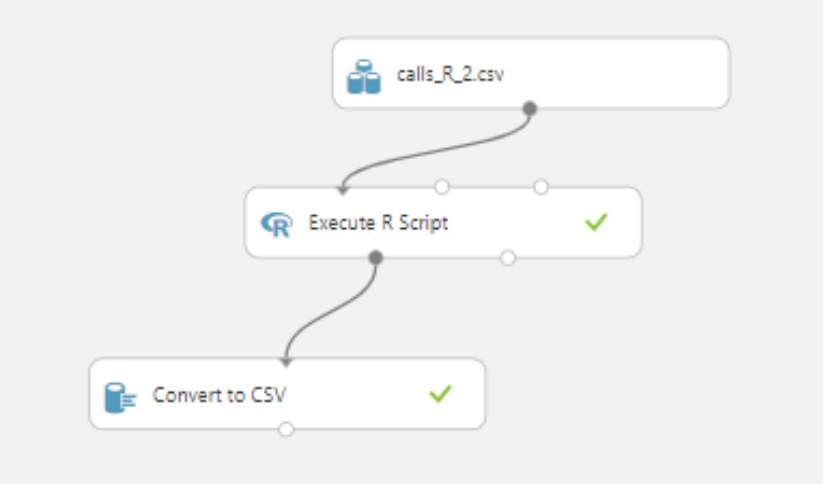

## 5.5 Optimization Model

Knowing the number of calls per day is not enough from a practical point of view. We need to know the optimal number of employees that will be able to answer these calls.

Another feature of Machine Learning Studio gives us the ability to upload our own algorithms to the experiment. The algorithm used for our optimization model is written in the R programming language. The software provides an Execute R Script module. All the scripts are included as source files.

This algorithm was written to create a table that contains the optimal

Figure 5.10: Optimization model in Machine Learning Studio




numbers of employees based on the number of calls. The table contains two columns, the first column represents the number of calls and the second represents the number of employees needed. The input remains the same data set resulted from pre-processing that we used in the regression model. The output will be converted in CSV format so that the user can download it.

The main idea of this algorithm consists of calculating the optimal number of call dispatchers. Every time a start time of the call is encountered, value 1 is added to the number of dispatchers currently being active. Respectively, every time an end of the call is encountered value -1 is added. For example, if call number 1 starts at 9:00, +1 is added, then comes the call number 2 that starts at 9:02, again +1 is added. Now the number of dispatchers that are currently active is equal to 2. If the call number 1 ends, the algorithm will free one dispatcher by adding the value -1. In this way, we will keep track of the optimal number of dispatchers that would be able to handle the phone calls. We will generate 1000 simulations. The resulted table is represented in *Figure 5.11* and *5.12*.

The resulted model in Machine Learning Studio can be seen in *Figure 5.10*.

## 5.6 Evaluation of the model implementation

Using Azure Machine Learning tools, we managed to build an experiment that fulfills all the goals that we set at the beginning. After scoring the

Figure 5.11: Optimal number of employees based on number of calls. *num_calls* stands for number of calls. *q* stands for number of emlpoyees.

| num_calls | q | num_calls | q | num_calls | q | num_calls | q | num_calls | q |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 28 | 3 | 55 | 4 | 82 | 5 | 109 | 6 |
| 2 | 1 | 29 | 3 | 56 | 4 | 83 | 5 | 110 | 6 |
| 3 | 1 | 30 | 3 | 57 | 4 | 84 | 5 | 111 | 6 |
| 4 | 1 | 31 | 3 | 58 | 4 | 85 | 5 | 112 | 6 |
| 5 | 2 | 32 | 3 | 59 | 4 | 86 | 5 | 113 | 6 |
| 6 | 2 | 33 | 3 | 60 | 4 | 87 | 5 | 114 | 6 |
| 7 | 2 | 34 | 3 | 61 | 4 | 88 | 5 | 115 | 6 |
| 8 | 2 | 35 | 3 | 62 | 4 | 89 | 5 | 116 | 6 |
| 9 | 2 | 36 | 3 | 63 | 4 | 90 | 5 | 117 | 6 |
| 10 | 2 | 37 | 3 | 64 | 4 | 91 | 5 | 118 | 6 |
| 11 | 2 | 38 | 3 | 65 | 5 | 92 | 5 | 119 | 6 |
| 12 | 2 | 39 | 3 | 66 | 5 | 93 | 5 | 120 | 6 |
| 13 | 2 | 40 | 4 | 67 | 5 | 94 | 5 | 121 | 6 |
| 14 | 2 | 41 | 4 | 68 | 5 | 95 | 6 | 122 | 6 |
| 15 | 2 | 42 | 4 | 69 | 5 | 96 | 6 | 123 | 6 |
| 16 | 2 | 43 | 4 | 70 | 5 | 97 | 6 | 124 | 6 |
| 17 | 3 | 44 | 4 | 71 | 5 | 98 | 6 | 125 | 6 |
| 18 | 3 | 45 | 4 | 72 | 5 | 99 | 6 | 126 | 6 |
| 19 | 3 | 46 | 4 | 73 | 5 | 100 | 6 | 127 | 6 |
| 20 | 3 | 47 | 4 | 74 | 5 | 101 | 6 | 128 | 6 |
| 21 | 3 | 48 | 4 | 75 | 5 | 102 | 6 | 129 | 6 |
| 22 | 3 | 49 | 4 | 76 | 5 | 103 | 6 | 130 | 7 |
| 23 | 3 | 50 | 4 | 77 | 5 | 104 | 6 | 131 | 7 |
| 24 | 3 | 51 | 4 | 78 | 5 | 105 | 6 | 132 | 7 |
| 25 | 3 | 52 | 4 | 79 | 5 | 106 | 6 | 133 | 7 |
| 26 | 3 | 53 | 4 | 80 | 5 | 107 | 6 | 134 | 7 |
| 27 | 3 | 54 | 4 | 81 | 5 | 108 | 6 | 135 | 7 |

Figure 5.12: Optimal number of employees based on number of calls. *num_calls* stands for number of calls. *q* stands for number of emlpoyees.

| num_calls | q | num_calls | q | num_calls | q | num_calls | q | num_calls | q |
|---|---|---|---|---|---|---|---|---|---|
| 136 | 7 | 163 | 7 | 190 | 8 | 217 | 9 | 244 | 9 |
| 137 | 7 | 164 | 7 | 191 | 8 | 218 | 9 | 245 | 9 |
| 138 | 7 | 165 | 7 | 192 | 8 | 219 | 9 | 246 | 9 |
| 139 | 7 | 166 | 7 | 193 | 8 | 220 | 9 | 247 | 9 |
| 140 | 7 | 167 | 7 | 194 | 8 | 221 | 9 | 248 | 9 |
| 141 | 7 | 168 | 7 | 195 | 8 | 222 | 9 | 249 | 9 |
| 142 | 7 | 169 | 7 | 196 | 8 | 223 | 9 | 250 | 10 |
| 143 | 7 | 170 | 8 | 197 | 8 | 224 | 9 | | |
| 144 | 7 | 171 | 8 | 198 | 8 | 225 | 9 | | |
| 145 | 7 | 172 | 8 | 199 | 8 | 226 | 9 | | |
| 146 | 7 | 173 | 8 | 200 | 8 | 227 | 9 | | |
| 147 | 7 | 174 | 8 | 201 | 8 | 228 | 9 | | |
| 148 | 7 | 175 | 8 | 202 | 8 | 229 | 9 | | |
| 149 | 7 | 176 | 8 | 203 | 8 | 230 | 9 | | |
| 150 | 7 | 177 | 8 | 204 | 8 | 231 | 9 | | |
| 151 | 7 | 178 | 8 | 205 | 8 | 232 | 9 | | |
| 152 | 7 | 179 | 8 | 206 | 8 | 233 | 9 | | |
| 153 | 7 | 180 | 8 | 207 | 8 | 234 | 9 | | |
| 154 | 7 | 181 | 8 | 208 | 8 | 235 | 9 | | |
| 155 | 7 | 182 | 8 | 209 | 8 | 236 | 9 | | |
| 156 | 7 | 183 | 8 | 210 | 9 | 237 | 9 | | |
| 157 | 7 | 184 | 8 | 211 | 9 | 238 | 9 | | |
| 158 | 7 | 185 | 8 | 212 | 9 | 239 | 9 | | |
| 159 | 7 | 186 | 8 | 213 | 9 | 240 | 9 | | |
| 160 | 7 | 187 | 8 | 214 | 9 | 241 | 9 | | |
| 161 | 7 | 188 | 8 | 215 | 9 | 242 | 9 | | |
| 162 | 7 | 189 | 8 | 216 | 9 | 243 | 9 | | |

regression model we see that the predictions are not perfectly accurate, but close to the real values. No better regression algorithm that Poisson regression could be found because the task itself is based on count variables that occur in a time interval, which involves using Poisson analysis regression.

The MAE indicates 37.0498 and RMAE is 56.759342, this means that we can likely obtain a prediction that could be by a few dozens wrong, but if we look at the table in *Figure* 5.11 and 5.12, we see that this could be solved by adding just a couple more employees. As the number of calls rises the consequences of making a not accurate prediction get lower.

# 6 Conclusion

This thesis work analyzed the problem of effective capacity planning of employees using machine learning principles. The first chapters provided an introduction to a relatively new field of the modern world, machine learning. We explained what is the interconnection between both human intelligence and machine learning, a branch of artificial intelligence.

Later the concepts of the main and most important machine learning approaches were studied. The focus was made on styles of learning and where they can be applied. Then we got familiar with Azure machine learning and the tools that it provides to its users. From the whole array of tools, Azure Machine Learning Studio was chosen.

The problem of effective capacity planning of employees was solved by building two models in Machine Learning Studio. The first model represents a predictive analysis model that predicts the number of calls in the day that we are interested in, while the second model provides a table that tells us what is the optimal number of employees depending on the number of calls.

The resulted models can be improved by making them public services so that they can be used daily by CCA Group a.s. Another improvement that can be made is continuously tracking the calls and adding data to the input so that the model will be trained with more precise and up to date data.

# 7 List of Abbreviations

- AI - Artificial Intelligence
- MLE - Maximum likelihood estimation
  * Maximum likelihood estimation is a method of estimating the parameters of a probability distribution by maximizing a likelihood function. A likelihood function measures the goodness of fit of a statistical model to a sample of data for given values of the unknown parameters
- MAP - Maximum a Posteriori estimate
  * Maximum a Posteriori estimation is a regularization of MLE.

# 8 List of Figures

# 9 List of source files

- *instructions.pdf* - instructions for launching the models
- *optimization_table.xlsx* - resulted optimization table
- *daily_calls.csv* - date set used in experiment creation
- *script.R* - source code used in "Execute R Script" module
- *data.R* - code for creating daily_calls.csv data set
- *pre_processing_plots.Rmd* - code for drawing graphs used in data pre-processing

# Bibliography


[1] Nils J. Nillson. *Introduction to Machine Learning.* Stanford University, 1998.

[2] Kevin P.Murphy. *Machine Learning. A Probabilistic Perspective.* Cambridge, Massachusetts Institute of Technology, London, 2012, p. 1-55

[3] David Braber. *Bayesian Reasoning and Machine Learning.* Cambridge University Press, New York, 2012, p. 305-322

[4] Francis Heylighen. *Cognitive Systems.* Universiteit Brussel, Brussel, 2014

[5] Oded Maimon, Lior Rokach. *Data mining and knowledge discovery handbook.* Tel-Aviv University, Israel, 2005

[6] Wikipedia, the free encyclopedia. Poisson regression [2020-04-23]: `https://en.wikipedia.org/wiki/Poisson_regression`

[7] Wikipedia, the free encyclopedia. Cluster analysis [2020-04-23]: `https://en.wikipedia.org/wiki/Cluster_analysis`

[8] Wikipedia, the free encyclopedia. Association rule learning [2020-04-23]: `https://en.wikipedia.org/wiki/Association_rule_learning`

[9] Wikipedia, the free encyclopedia. Markov decision process [online:2020-04-24]: `https://en.wikipedia.org/wiki/Markov_decision_process`

[10] Wikipedia, the free encyclopedia. Game theory [2020-04-24]: `https://en.wikipedia.org/wiki/Game_theory`

[11] Wikipedia, the free encyclopedia. Control theory [2020-04-24]: `https://en.wikipedia.org/wiki/Control_theory`

[12] Wikipedia, the free encyclopedia. MAP estimation [2020-04-25]: `https://en.wikipedia.org/wiki/Maximum_a_posteriori_estimation`

[13] Wikipedia, the free encyclopedia. ML estimation [2020-04-25]: `https://en.wikipedia.org/wiki/Maximum_likelihood_estimation`

[14] Microsoft Machine Learning Services [2020-06-10] `https://azure.microsoft.com/cs-cz/services/machine_learning/`

[15] Microsoft Azure Portal [source link]
https://portal.azure.com/?l=en.en-ushome

[16] Microsoft Azure Machine Learning Studio [source link]
https://studio.azureml.net/

[17] Microsoft Azure Machine Learning Studio documentation [2020-06-10]:
https://docs.microsoft.com/en-us/azure/machine-learning/studio/

[18] Ahmed Khemiri. Web article "Introduction to Microsoft Azure Machine Learning", February 2019 [2020-06-10]:
https://medium.com/