OULUN AMMATTIKORKEAKOULU

Petri Leppänen

# PREDICTIVE MODELLING TO CALCULATE THE NEED FOR OFFICE SPACE

# PREDICTIVE MODELLING TO CALCULATE THE NEED FOR OFFICE SPACE

Petri Leppänen
Master Thesis
Spring 2024
Degree Programme in Data Analytics and Project Management
Oulu University of Applied Sciences

**ABSTRACT**

Oulu University of Applied Sciences
Degree Programme in Data Analytics and Project Management

---

Author: Petri Leppänen
Title of the thesis: PREDICTIVE MODELLING TO CALCULATE THE NEED FOR OFFICE SPACE
Thesis examiner(s): Ville Majava
Spring 2024 Pages: 30

---

Remote work or hybrid work after the pandemic is the new default in people's workplaces and many have been making good home offices themselves. This is why increasingly many offices are empty most of the time. This kind of prediction can be a good tool to help seek almost the right size offices and with presence sensors and reserved spaces; there can be enough space for different type of work. Meeting rooms, quiet rooms or areas, and ad hoc spaces just to sit down and print out papers and move to a meeting are the kinds of places employees need when they come to the office. For example, lounge and café type of spaces are also good for small talk with persons outside of your team. The purpose of an office space has changed after the pandemic and the company needs to be ready to act based on different kinds of variables and predictions.

In this thesis, the focus was on investigating the possibility of making some kind of prediction that can be used to anticipate office sizes in the future. My employer asked me to see if there is some kind of library or model to predict our office places in the future. This project-based thesis aimed to make the first iteration of a prediction that can be used and modified according to company needs. The commissioning company provided all the data needed for this thesis work.

The results show that the prediction met the requirements of the commission. With more data, the prediction will be more accurate. Prophet may not be the best for doing predictions and therefore the suggestion is to further investigate different libraries for different kinds of needs. Longer periods of data will get better prediction results and also adjusting variables that are used to make predictions would be ones that need more investigating.

---

Keywords: Python, Machine Learning, Predict, Data Science, Data Visualization, Data Normalize, Plotly, Pandas, Prophet, TensorFlow, PyTorch, Power BI, Dash

# CONTENTS

## ABBREVIATIONS

| | |
|---|---|
| CSV | Comma Separated Values |
| HR | Human Resources |
| CSS | Cascading Style Sheets |
| HTML | HyperText Markup Language |
| SQL | Structured Query Language |
| GDPR | General Data Protection Regulation |

# 1 INTRODUCTION

The goal of this thesis was to make a prediction model that can be used to predict the need for the office space. Past and current time need to give baseline for the prediction model so that in the future there can be more accurate office sizes for company to use.

The programming language is Python. To make a prediction model, Pandas, Plotly and Prophet libraries are used. To show data I use mainly dash at this point. Data will be collected and enriched from the company's programs and public data. Selecting this model is about understanding of how much employees are using office space now and in the future. The company gets more data, which facilitates decision-making.

Base data is collected from software called Worksense and more data is collected from our HR program Sympa. Worksense program gives usage information and HR program gives employee count of that office. Right amount of different variables will lead to good result in prediction. Variables can be for example public holidays, overall development of the area, my employer's interest in the current location growing and universities line of studies in areas. Too many variables or too few variables can lead to a different kind of result and therefore I need to test and select the best one that suits us. The task is to first make a simple model that makes predicting linear and then adds more variables in the future. Prophet is used to educate the prediction model with our data to make better and more accurate results.

# 2 BACKGROUND STUDIES

Various topics were studied to develop a prediction model for decision-making regarding office space, Python as main programming language. Python is an easily approachable programming language and used in many places. Python runs on different operating systems, and it is freely available.

Use of the Python code for cleaning or merging data is one thing that needs to be done in this thesis. OpenRefine was another tool to check that data was suitable and has correct values.

In generally Data visualization entails the idea that people who see your data will understand what you show them.

## 2.1 Machine Learning

I took machine learning course to support my studies and to acquire more information about the topic. Machine learning is like teaching people to for example write in a new language: first you train it, test it and validate the training and then you take it in use.

Machine learning has been developed to be one of the fastest growing fields of information technology. Many companies are investigating things how to make their own learning models or they have already released their own closed or open-source libraries for commercial or non-commercial use.

We use machine learning on a daily basis for example when using Facebook or watching Netflix. Netflix for example gives us suggestions on what you might like to watch based on your earlier choices.

Artificial intelligence can be used for example in automatic cars to recognize obstacles or see traffic signs. From satellite pictures AI can see objects for example

airplanes parked at the airport (see Figure 1). Artificial intelligence can help cameras to recognize different languages and make translations so you can see the text in your native language in your device screen.

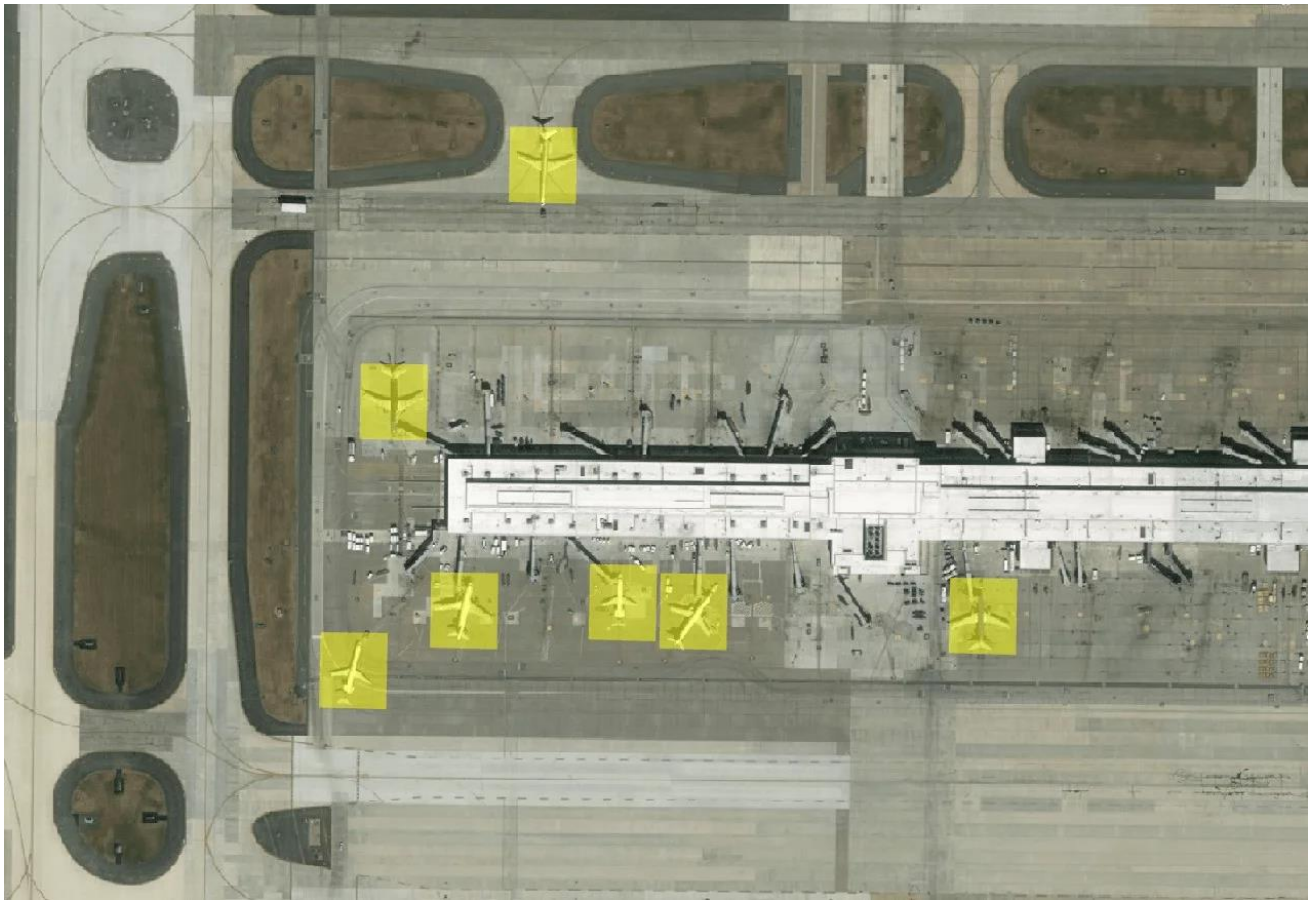

*FIGURE 1. Screenshot from* https://emerj.com/ai-sector-overviews/ai-applications-for-satellite-imagery-and-data/ *27.11.2023*

Artificial intelligence has been used in some ways over 30 years, but it was originally founded in 1956. The last ten years artificial intelligence has developed a lot but it has developed a lot under the radar. OpenAI and ChatGPT was things that makes it popular for everyone to use.

This is not necessarily only a good development. With this technology you can for example make false pictures and videos with sounds that are difficult to recognize as fake. You should not trust what these images or videos present to you.

Computer chess game has AI written in it and which is why it can make different kind of moves during the game. Game uses machine learning today so for example enemies are not in the same place or Game outcome is changed every time user plays the game and that is why it makes games more interesting.

With artificial intelligence, applications can do for example email categorising much easier for the user. It can automatically show that some email is spam, and some other email is an advertisement.

Machine learning can be used to make decisions for the user in everyday life by for example categorizing email as spam. The most successful kinds of machine learning algorithms are those that automate decision-making processes by generalizing from known examples. (Müller & Guido. 2016)

Deep learning uses neural networks to educate models for example to recognize from the image what is on it. This way of educating is similar to how people learn.

Machine learning has been used in healthcare, the military, and in games. Cancer recognition is field where machine learning can be used, and the military can use it for example to see from satellite pictures where the enemy is.

Questionable decision-making would be to give machine learning decide for example when the military needs to launch a missile or drop a bomb. Current AI and machine learning development is already going in that direction and fast.

## 2.2 Python

In my studies there were courses on using Python in data analytics and Python was in the first steps easy to use and understand. Python is an old programming language and nowadays it is widely used in machine learning, data visualizations and prediction models. First thing was to search right kind of libraries to make this thesis work happen.

Python can be run on macOS, Unix variants, Windows, Linux and also some mobile devices.

Python has not got many major versions. During the writing of this thesis the Python version is 3.12.1 and the major version is 3. Python releases come around once a year and it is a feature release. The last major version 3.0 was released in 2008.

R is more used in visualization and analysis whereas Python is more focused on for example machine learning and prediction models. R language was developed in 1992 so it is almost as old as Python.

Python is widely used in different kinds of prediction models, artificial intelligence, and machine learning. Python is one of the easiest programming languages to start with and has lots of libraries built around the language.

Most of the libraries and extensions are being programmed with C programming language.

### 2.3 Jupyter Notebook

Jupyter Notebook (see Figure 2) is a useful tool to run Python code. It also supports different programming languages. You can share notebooks and also make output for example HTML format so output can be viewed via web browser.

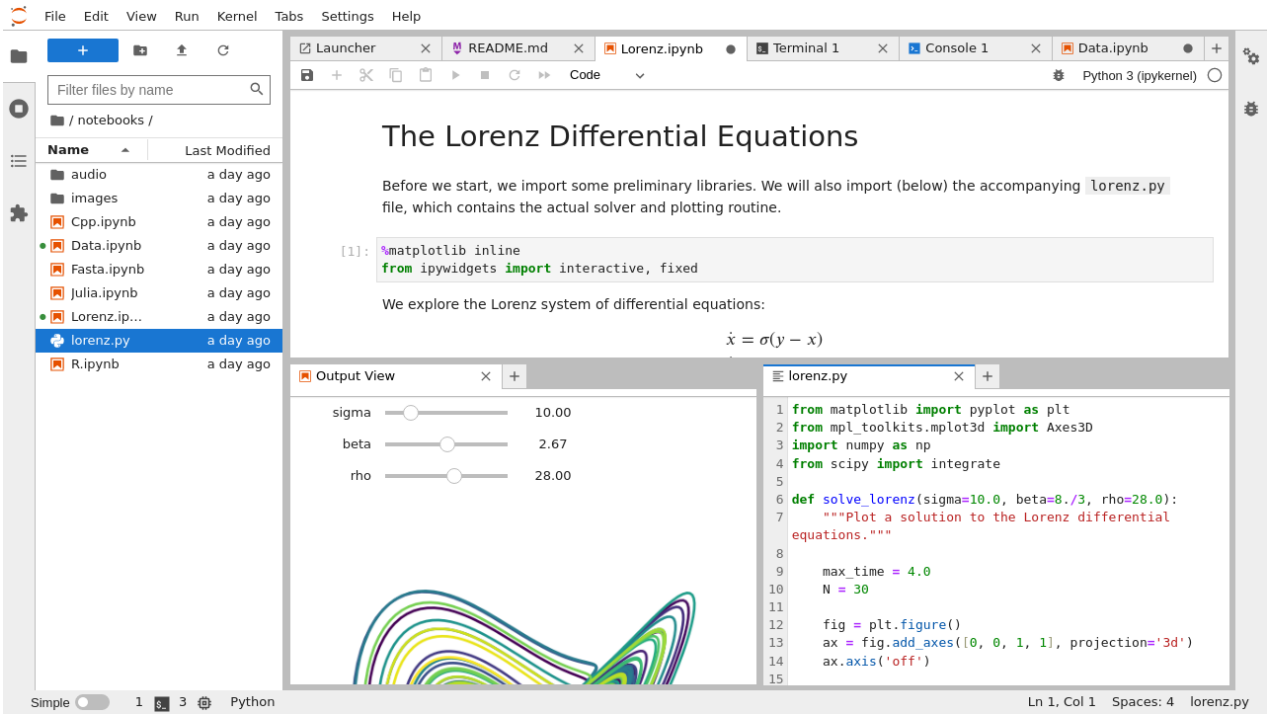


*FIGURE 2. Screenshot from https://docs.jupyter.org/en/latest/ Jupyter Notebook 27.11.2023*

Jupyter Notebook is easy to use and when coding Python language and with Jupyter Notebook you can see your outcome fast and easy. Jupyter Notebook supports many programming languages, we only need the Python support. (Müller & Guido. 2016)

Visual Code free version is a good tool for programming in different languages and it has many extensions. Visual Code show your errors when you write the code so it is easier to see what is wrong in it. One of them is Jupyter Notebook and with that you can run Python code on fly. Visual Code can be installed into macOS, Linux or Windows.

Visual Code (see Figure 3) and Jupyter Notebook are used to program this prediction model. Jupyter Notebook is widely used in Python programming.

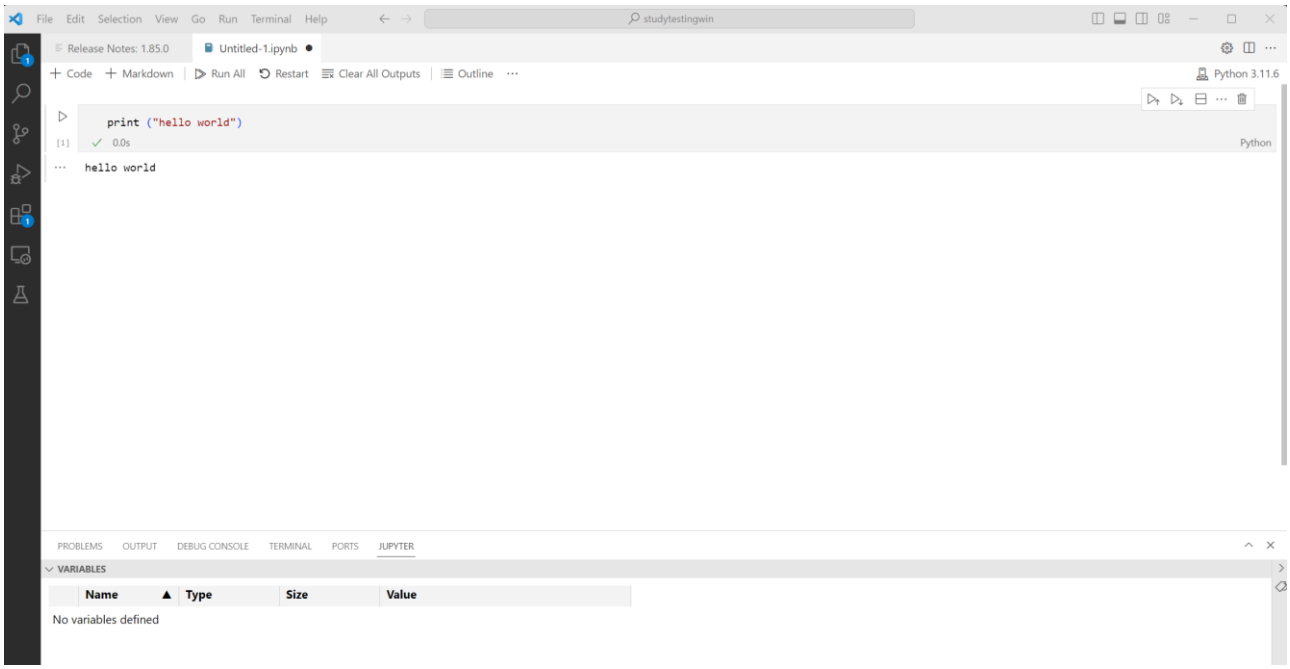


*FIGURE 3. Screenshot from application Visual Code 27.11.2023*

## 2.4 Pandas

Pandas library is used in this thesis to read and write CSV files to data frames. Pandas can also read and write data in Excel or different kinds of databases using SQL language. Pandas is used in Data Science and Machine Learning as one of the best tools for handling data. Pandas uses DataFrame which is a table close to an Excel sheet but Pandas has many methods to operate and modify a table.

> *Another valuable tool provided by pandas is its ability to ingest from a great variety of file formats and databases, like SQL, Excel files, and comma-separated values (CSV) files." (Müller & Guido. 2016)*

Prediction model data is in CSV file and uses Pandas to read and write CSV files.

## 2.5 Prophet

The prediction model used in this thesis is the Prophet time series forecasting model which is designed to be easy and simple to use. Prophet is developed by

Facebook (now known as Meta). Weather reports are one good example of time series forecasting. Seasonality in weather forecasts is often used because for example in Finland there are four different seasons. Seasonality is used for those seasons. *You accomplish these tasks and goals through an intuitive but very flexible programming interface that is designed for both the beginner and expert alike. (Rafferty. 2023)*

Prophet is easy to use and understand and with just a couple of lines of code, you have a graph to show your prediction from your original data. *The chart in the weather app on your phone showing the expected temperature for the next 7 days? That's also the plot of a time series. (Rafferty. 2023)*

There can be different variables like war or natural disaster that make our predictions go wrong way since these are not the kind of things that we already know, or these have no seasonality.

Basic mathematic equation of Prophet is $\gamma(t) = g(t) + s(t) + h(t) + \in_t$. Model forecast value $y$ and time value $t$ is given by the $y(t)$ function. Function has four components, summed together or multiplied together. $g(t)$ the growth component, $s(t)$ the seasonality component, $h(t)$ the holiday component and $\in_t$ is the error term. *To really understand what is happening, we'll need to break down each of those components." (Rafferty. 2023)*

Prophet Library has everything that is needed to do this prediction model. Prophet Library is simple to use with clear output showing prediction. Prophet also has built-in country-specific holidays that can be used if there is need for country specified seasonality in prediction.

### 2.6 TensorFlow

TensorFlow is a machine learning platform that offers pre-trained machine learning models for you to use. It provides machine learning solution on different skill levels. Spotify uses TensorFlow models to generate playlist for users. TensorFlow was developed by Google. *Machine-learning algorithms require many mathematical*

*operations. Often, an algorithm boils down to a composition of simple functions iterated until convergence. (Shukla. 2018)*

TensorFlow was too complex and heavy for the purposes of this thesis. TensorFlow has no easy way to use holidays and seasonality. For office space prediction TensorFlow is way too complex and there was no seasonality or country holidays built-in to the model.

### 2.7 PyTorch

PyTorch is used for cars to see objects like nearby cars, motorcycles, bikers and other obstacles. PyTorch is developed by Facebook*. Many machine learning researchers and practitioners from academia and industry have adapted PyTorch to develop deep learning solutions, such as Tesla Autopilot, Uber's Pyro, and Hugging Face's Transformers. (Raschka et al. 2022)*

PyTorch was not the right library in this thesis since it was too complex for time series forecasting and not so easy to implement if needed in the future. There was no seasonality or country specified holidays built-in to this library. Building these would have taken time.

# 3 DATA COLLECTION

The theoretical framework is to collect needed libraries to make a prediction model. I use a different kind of prediction model that has been tested in other environments and made suitable for calculating and predicting the needs of the office space. The dataset comes from programs and public figures and merges those into one data file.

Basic data will be collected from the system that collects utilization rates in our Turku office. We began to use the same system in Hämeenlinna in January 2023 called Worksense and later every new office that my current employer opens. Secondary data will be collected from our HR program and the last part is enrichments with public data and all the needed variables that we need for this to calculate the best prediction model. The dataset will be merged into a CSV file

## 3.1 Data Normalizing

*Normalization is a technique for producing a set of relations (data represented logically in a two-dimensional format using rows and columns) that possesses a certain set of properties. (Oppel. 2009)* Data needs to be clean, standard, and in the right format. Data used in this model was checked with OpenRefine (see Figure 4) which is a free open-source tool to clean and transform data.

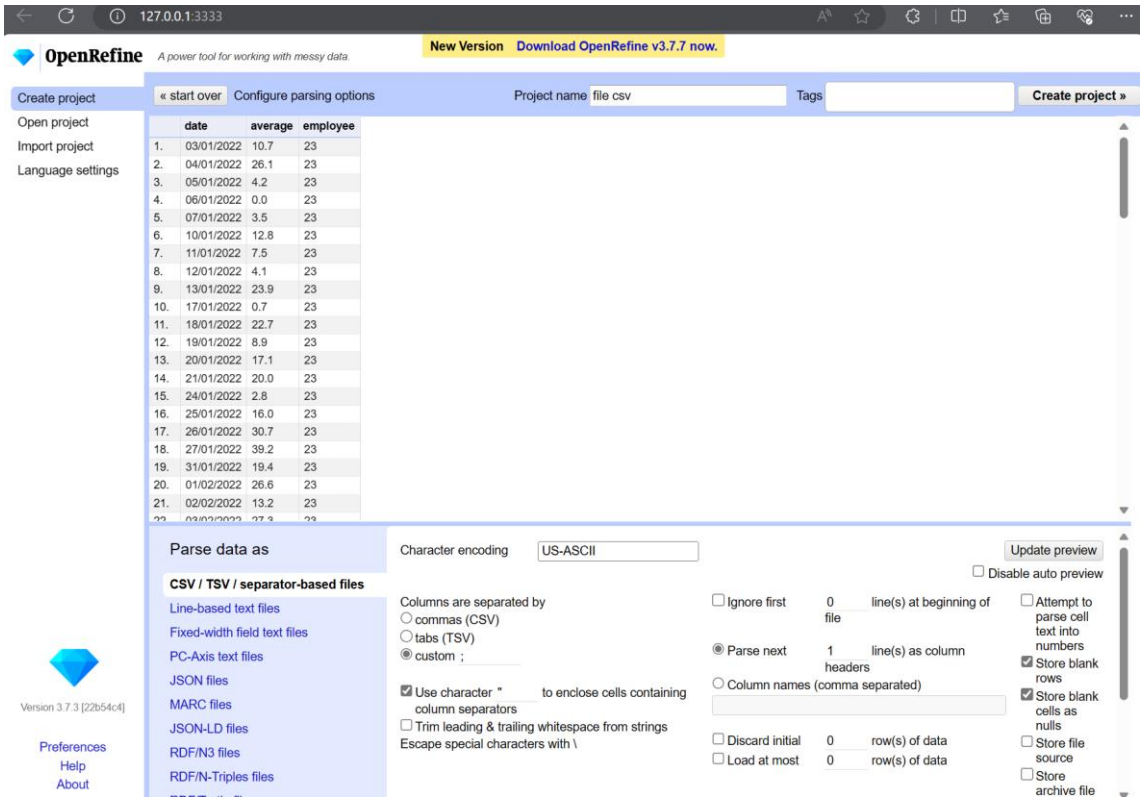


*FIGURE 4. Screenshot from application OpenRefine 27.11.2023*

### 3.2 Data Ethics

Data ethics provides information on how to use, collect and store data. *Data ethics purports to provide answers to these questions, or at least a framework for wrestling with them. (Grus. 2020)*

There is legislations and regulations on what kind of data you can show, keep, and present also GDPR tells how long you can keep data. In this thesis, data is used for internal use, and it has been collected internally. Everyone knows where data comes from and where data is being used.

### 3.3 Data Variables

Write variables used in the model are Finnish holiday, employee count, and average usage of places in the office. Finnish holiday can be used with Prophet library with a minimum amount of coding. Average usage seems to be a good

way to get the overall value of usage percent. Read the original file, take the date and employee count, and every ten rows count the average value of usage then write it to a new CSV file.

```
import csv
import pandas as pd
with open('original_file.csv', newline='') as csvfile:
    reader = csv.reader(csvfile, delimiter=';')
    next(reader)
    data = list(reader)
df = pd.DataFrame(data, columns=['date', 'usage', ‘employee’])

df['usage'] = df['usage'].astype(float)

daily_averages = df.groupby(df.index // 10)['usage'].mean()
new_data = []
for i in range(0, len(data), 10):
    date = data[i][0]
    emp = data[i][2]
    avg = daily_averages[i // 10]
    new_data.append([date, avg, emp])

with open('file.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile, delimiter=';')
    writer.writerow(['date', ‘average’, ‘employee’])
    writer.writerows(new_data)
```

With this code from the original source of place usage can be made to calculate average from daily usage.

# 4 PREDICTION MODEL

The approach is to test different versions of the prediction model and see what kind of model works best in this case. The dataset contains collected data from Worksense program, HR program, and public data where we get environment data that have been collected for example from employment figures in areas.

The first attempt was to create this prediction model using TensorFlow and Keras. However, this was not useful so the next step was to test PyTorch which was not so good either so I ended up using Prophet library from Meta (Facebook).

Prophet provides everything that is needed to make prediction used in this thesis.

## 4.1 Model

Import libraries pandas, prophet, dash and plotly.graph_objs (see Figure 5).

```
# Import libraries
import pandas as pd
from prophet import Prophet
import dash
from dash import dcc, html
from dash.dependencies import Input, Output
import plotly.graph_objs as go
```

*FIGURE 5. Screenshot of code import libraries 27.11.2023*

Data Loading and Preprocessing (see Figure 6)

```
# Load the CSV data into a pandas DataFrame from file.csv
try:
    data = pd.read_csv("file.csv", delimiter=";")
except FileNotFoundError:
    print("Error: CSV file not found.")
    exit()

# Check if the column names are spelled correctly and if not then show error message
if 'average' not in data.columns or 'employee' not in data.columns:
    print("Error: 'average' or 'employee' column not found in CSV file.")
    exit()
```

*FIGURE 6. Screenshot of code to load CSV file 27.11.2023*

Load data from a CSV file named "file.csv" and if file is not present then it gives error message. It checks if the required columns (‘average’ and ‘employee’) exist in the data frame and prints an error message and exits if they do not. Data pre-processing is performed on the loaded data. The 'date' column is converted to a datetime format, and additional columns for 'year' and 'month' are created to help with analysis (see Figure 7).

```
# Preprocess the data
data['date'] = pd.to_datetime(data['date'], format='%d/%m/%Y')
data['year'] = data['date'].dt.year
data['month'] = data['date'].dt.month
```

*FIGURE 7. Screenshot of code preprocess data 27.11.2023*

Define data frame for place and employees (see Figure 8).

```
# Define a DataFrame for place usage prediction
place_data = data[['date', 'average']].rename(columns={'date': 'ds', 'average': 'y'})

# Define a DataFrame for employee count prediction
employee_data = data[['date', 'employee']].rename(columns={'date': 'ds', 'employee': 'y'})
```

*FIGURE 8. Screenshot of code data frames 27.11.2023*

Two Prophet models are created: one for predicting average place usage and the other for predicting employee count. The Prophet models are configured to include yearly seasonality patterns. Finnish holidays are included and fitting the model (see Figure 9).

```
place_model = Prophet(yearly_seasonality=True)  # Add seasonality
place_model.add_country_holidays(country_name='FI')  # Add Finnish holidays
place_model.fit(place_data) # Fit a Prophet train model for place usage prediction


employee_model = Prophet(yearly_seasonality=True)  # Add seasonality
employee_model.add_country_holidays(country_name='FI')  # Add Finnish holidays
employee_model.fit(employee_data) # Fit a Prophet train model for employee count prediction
```

*FIGURE 9. Screenshot of seasonality, holidays and model fit 27.11.2023*

```
# Predict the future for both place usage and employee count
future_years = 5  # Years to predict
future_place = place_model.make_future_dataframe(periods=365 * future_years, freq='D')
forecast_place = place_model.predict(future_place)

future_employee = employee_model.make_future_dataframe(periods=365 * future_years, freq='D')
forecast_employee = employee_model.predict(future_employee)
```

*FIGURE 10. Screenshot of code future years and period 27.11.2023*

Future dates are generated for predictions. By default, it extends the predictions by five years into the future, but you can add the number of years you like to have the time of the prediction (see Figure 10). Predictions are made using the trained Prophet models.

```
# Rename column 'yhat' to 'Predict'
def rename_columns(df):
    return df.rename(columns={'yhat': 'Predict'})

forecast_place = rename_columns(forecast_place)
forecast_employee = rename_columns(forecast_employee)
```

*FIGURE 11. Screenshot of code renaming column 27.11.2023*

The code renames the 'yhat' column in the prediction results (see Figure 11). It changes it to 'Predict' to improve clarity.

```
# Load actual data from the same DataFrame
actual_data_2022 = data[data['year'] == 2022]

# Save predictions to a new CSV file
predictions_csv = pd.concat([forecast_place[['ds', 'Predict']], forecast_employee[['ds', 'Predict']]]).reset_index(drop=True)
predictions_csv.to_csv("predictions.csv", index=False)
```

*FIGURE 12. Screenshot of code actual data 27.11.2023*

Actual data is shown in diagrams to clarify change between actual data and prediction (see Figure 12). This code for actual data is from 2022. Running a prediction model makes also CSV output file and this output file can be used to make visualization for example in Power BI.

Dash web application is created using the dash (see Figure 13). It defines the layout of the web application including the HTML structure and graph components for visualizing predictions and actual data.

A callback function is defined using the @app.callback decorator. The function updates the graphs based on user interactions, specifically when users zoom in on the graphs.

The actual data for the year 2022 is plotted on both graphs. It is extracted from the original data dataframe. Predictions for place usage and employee count are also plotted on their respective graphs. Application is run to show prediction model graphs.

```
# Create a Dash app and figures
app = dash.Dash(__name__)


def create_layout():
    return html.Div([
        html.H1("Place Usage and Employee Count Predictions vs. Actual Data"),
        dcc.Graph(id='predictions-actual-data-place'),
        dcc.Graph(id='predictions-actual-data-employee'),
        dcc.Input(id='input', type='text', style={'display': 'none'})
    ])

app.layout = create_layout()

@app.callback(
    Output('predictions-actual-data-place', 'figure'),
    Output('predictions-actual-data-employee', 'figure'),
    Input('input', 'value')
)
def update_graph(value):
    fig_place = go.Figure()
    fig_place.add_trace(go.Scatter(x=actual_data_2022['date'], y=actual_data_2022['average'], mode='lines', name='Actual Usage of Places'))
    fig_place.add_trace(go.Scatter(x=forecast_place['ds'], y=forecast_place['Predict'], mode='lines', name='Predicted Usage of Places'))
    fig_place.update_layout(title="Place Usage Prediction vs. Actual Data",
                            xaxis_title="Year",
                            yaxis_title="Average Usage")

    fig_employee = go.Figure()
    fig_employee.add_trace(go.Scatter(x=actual_data_2022['date'], y=actual_data_2022['employee'], mode='lines', name='Actual Employee Count'))
    fig_employee.add_trace(go.Scatter(x=forecast_employee['ds'], y=forecast_employee['Predict'], mode='lines', name='Predicted Employee Count'))
    fig_employee.update_layout(title="Employee Count Prediction vs. Actual Data",
                               xaxis_title="Year",
                               yaxis_title="Employee Count")

    return fig_place, fig_employee

if __name__ == '__main__':
    app.run_server(debug=True)
```

*FIGURE 13. Screenshot of code create dash app 27.11.2023*

# 5 DATA VISUALIZATION

Data is visualized using free and then paid software. Power BI is the main visualization tool for company users. Power BI will be used as the, second visualization tool.  Power BI is easy to use and has built-in functions to get reports and dashboards which are visually good. Power BI can also transform data if necessary.

Dash is a solution to show visualization to users and it is free. If you want to use full Dash potential, you need for example one Linux workstation or server and install Python and all the needed libraries in it. Then add it so that you can see web services on that machine, and you can then see output of your code. *Good visual presentations tend to enhance the message of the visualization. (Wilke. 2019)*

Visualization can look like this map where different shade represent different income (see Figure 14).

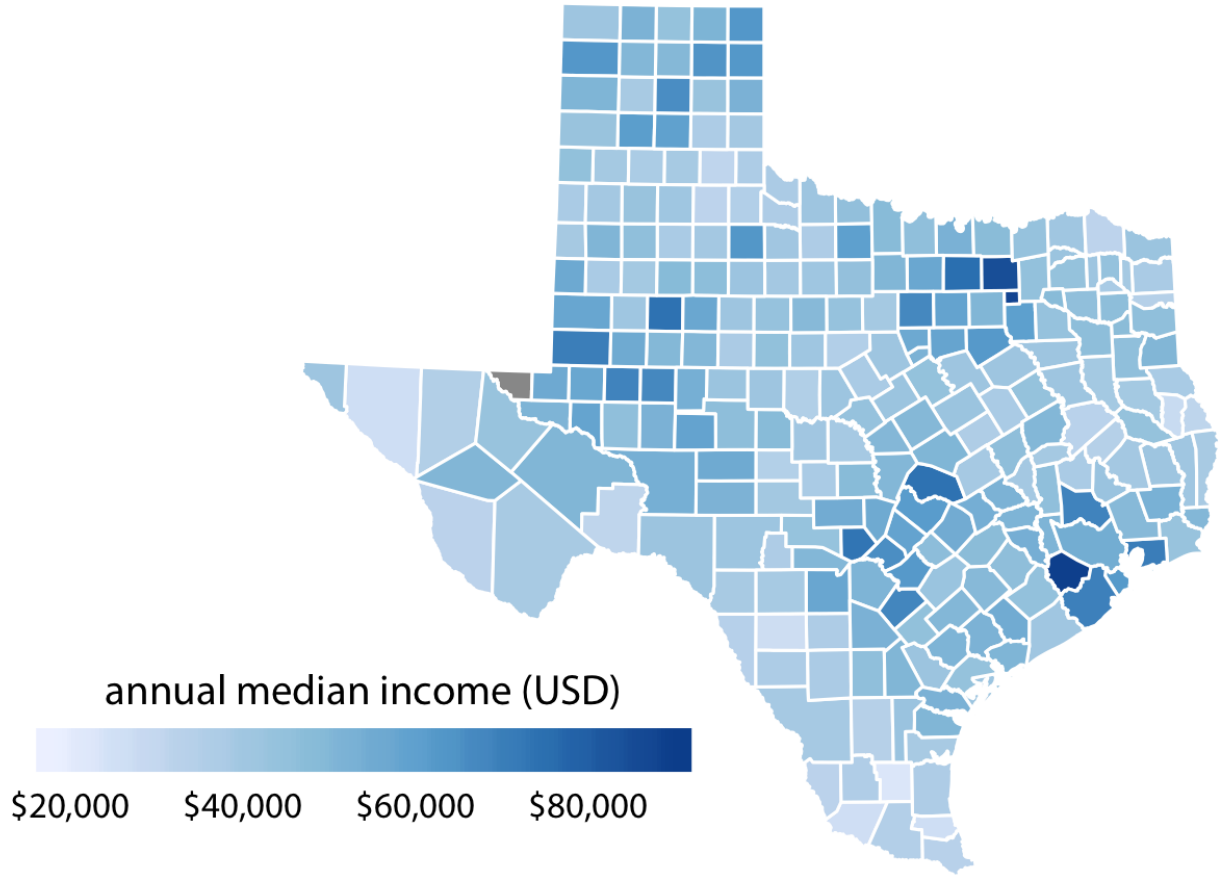


Figure 14. (Wilke 2019) *27.11.2023*

Lines are a good way to represent this prediction model; they are clean and simple to understand. In Dash, you can also change the style of your graphs easily using CSS or Dash Bootstrap Components library. With Dash Bootstrap Components you can create a styled app with a responsive layout.

Data visualization is one form of storytelling and you need to make it so simple that everyone can understand what message of that report or dashboard is.

## 5.1 Power BI

Power BI will be used to make visualizations. It provides a good way to show different kind of graphs and you can make for example buttons to change year, month or whatever suits for your purposes. In Power BI you can make different kind of connections to data (see Figure 15).

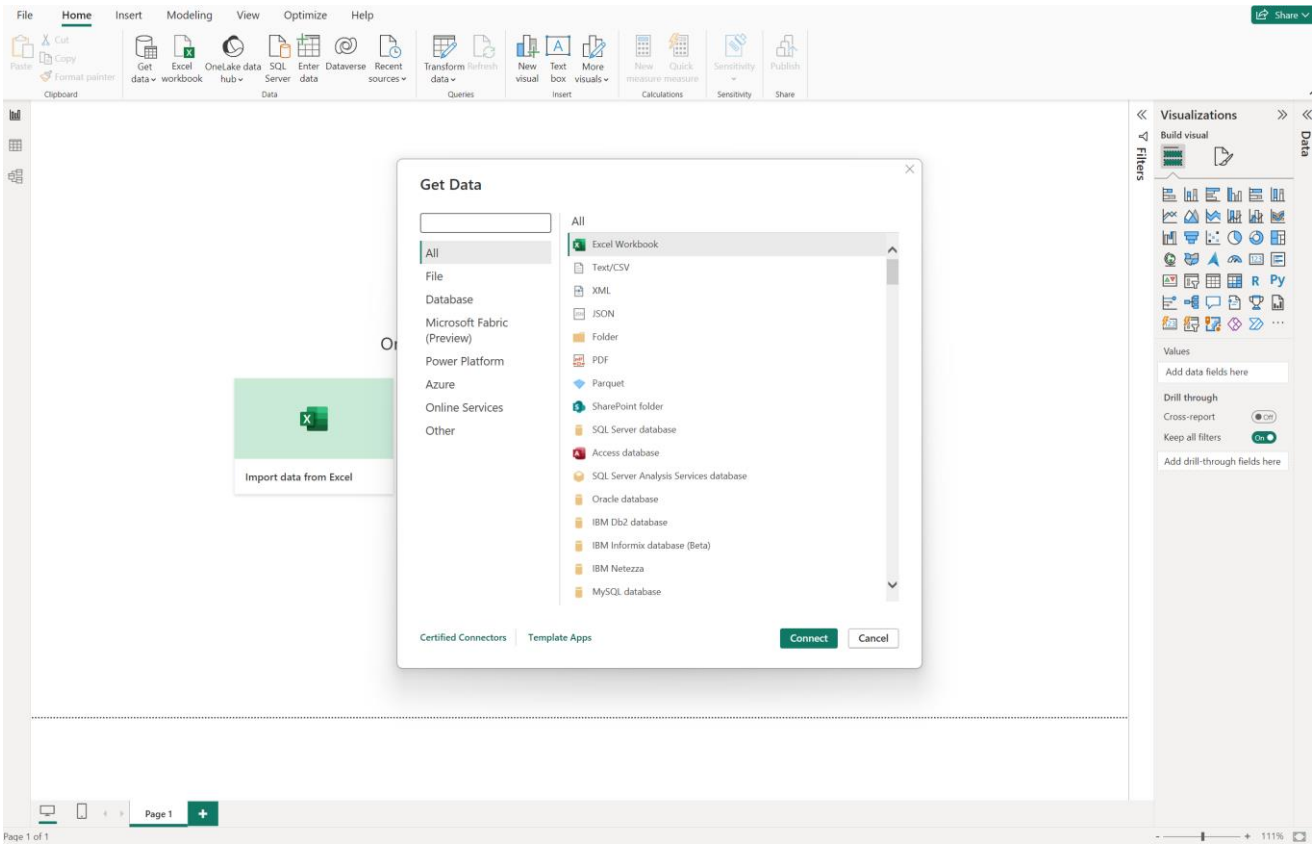


*FIGURE 15. Screenshot of Power BI 27.11.2023*

Power BI has built-in tools, like getting data from different sources, tool to transform data so it can be used to make good and helpful visualization. *Business intelligence tools allow multiple individual tables and charts to be combined on a single page or report. (Deckler. 2022)*

Below prediction model is shown in Power BI (see Figure 16). Output is almost like what Dash provides but Power BI needs a paid license to work. Also, people who want to see your graphs or dashboards need a license or you need to have an expensive version of Power BI in which case you can embed graphs for example on the company's intranet page.

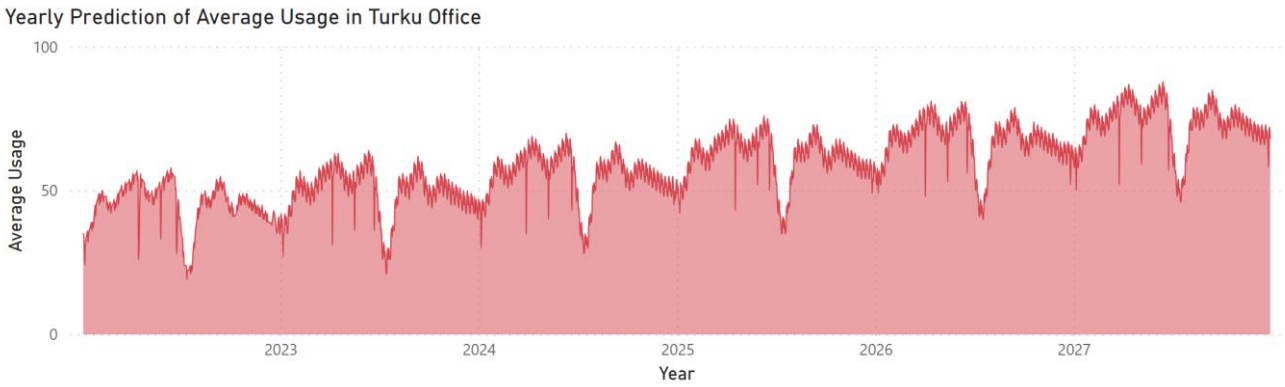


*FIGURE 16. Screenshot of Power BI graph 27.11.2023*

## 5.2 Dash

Dash is an open-source Python framework for building interactive, web-based data visualization applications. It was created and is actively maintained by Plotly, a data visualization company. Dash was first released in 2017 and has gained popularity for its ease of use, flexibility, and the ability to create interactive dashboards and web applications for data exploration and presentation.

Dash is a framework that can be used to make web applications easily. With this framework there is no need to know HTML, CSS or other programming languages or libraries.

Prediction model would be easy to show and embed for example on companies' intranet page. Model produced two different graphs, average usage and prediction (see Figure 17) and employee count at the office actual persons and prediction (see Figure 18).

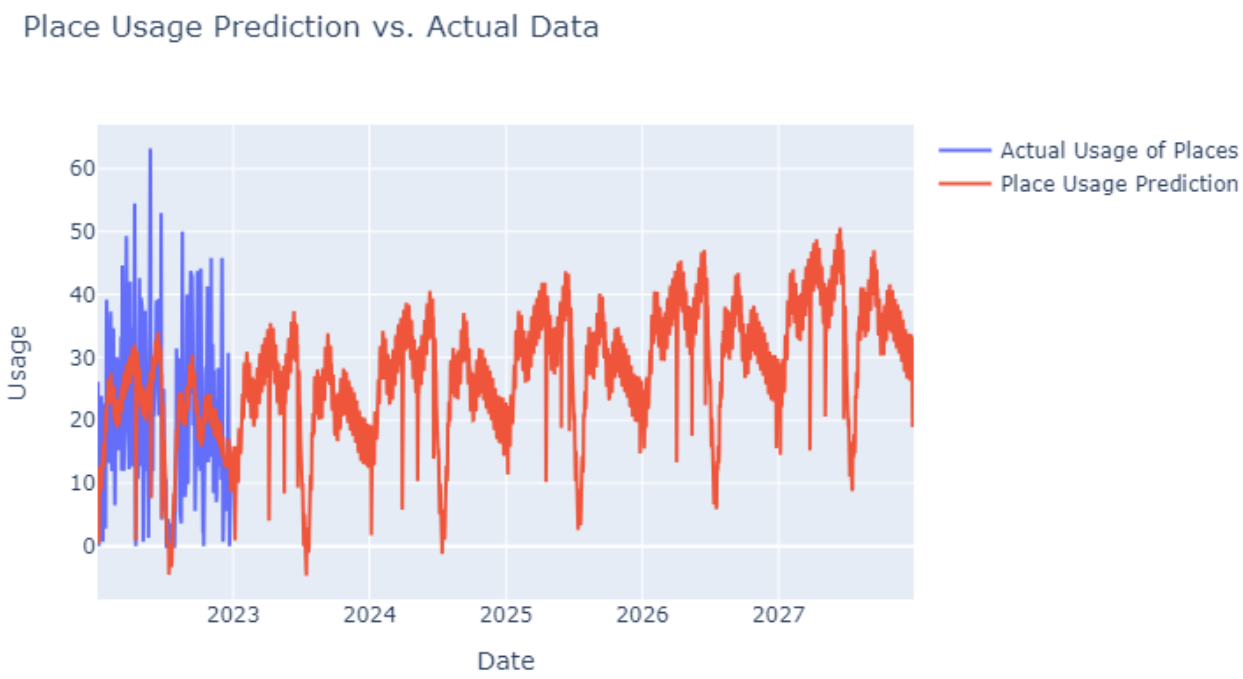


*FIGURE 17. Screenshot of dash place graph 27.11.2023*

Output from prediction model using Dash. Employee count prediction and actual data from 2022.

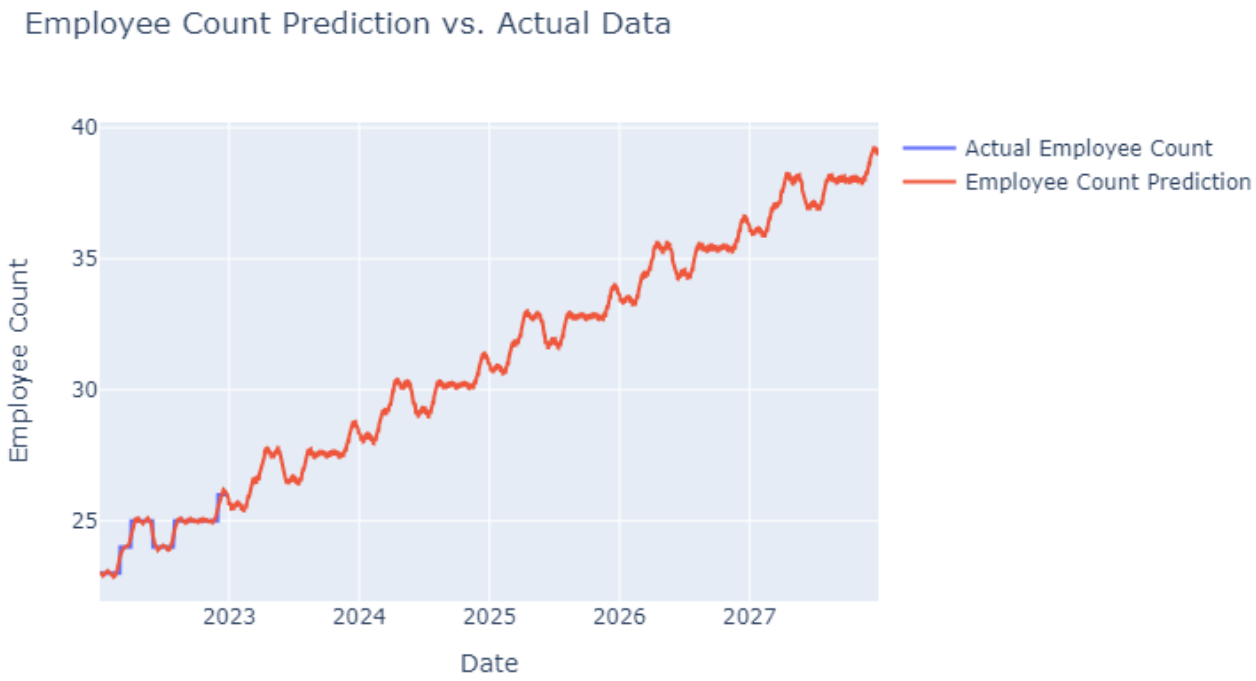


*FIGURE 18. Screenshot of dash employee graph 27.11.2023*

# 6 CONCLUSION

Firstly, I tried to do this model with TensorFlow and PyTorch but those where too complex to use in this prediction model and they did not have all the needed functionalities build in. Holidays and seasonality would need considerable programming work and therefore I looked for a more suitable library. TensorFlow and PyTorch are good for deed learning neural networks. For example you can make model with those than can accurately tell l you what is in picture and another example would be model that is used when self-driving car investigates its surroundings.

I needed library that is fast and simple. So I searched and first tried TensorFlow but it felt too complex and the same was with PyTorch but then I found Prophet. Prophet library has everything that was needed to build this model and it was easy to use. Prophet has all the needed functionalities and worked well with seasonality data.

Actual prediction model works but it needs more data so it can predict future in a more accurate way. After further data collection then model needs to be run again.

Data-drive decision-making point of view gives now only little value but when the model can be more accurate than the prediction model will help when a company seeks out new office space. The decision of the office spaces is not entirely only viewed by usage percent because more factors need to be considered when looking at the office spaces. Other variables in office space are needed for example secure room to make work that needs a room that has been checked out by an outside evaluator. The need for meeting rooms is also one thing that should be considered when looking at the office spaces and how many need a fixed seat at the office, so they are every day at the office.

## REFERENCES


Andreas C. Müller, Sarah Guido 2016. Introduction to Machine Learning with Python O'Reilly Media, Inc. Search date 10.08.2023. https://learning.oreilly.com/library/view/introduction-to-machine/9781449369880/. Access required.

Greg Rafferty 2023. Forecasting Time Series Data with Prophet. Second Edition. Packt Publishing. Search date 01.12.2023. https://learning.oreilly.com/library/view/forecasting-time-series/9781837630417/. Access required.

Nishant Shukla 2018. Machine Learning with TensorFlow. Manning Publications. Search date 02.08.2023. https://learning.oreilly.com/library/view/machine-learning-with/9781617293870/. Access required.

Sebastian Raschka, Yuxi (Hayden) Liu, Vahid Mirjalili 2022. Machine Learning with PyTorch and Scikit-Learn. Packt Publishing. https://learning.oreilly.com/library/view/machine-learning-with/9781801819312/. Access required.

Andy Oppel 2009. Data Modeling, A Beginner's Guide. McGraw-Hill. Search date 08.07.2023. https://learning.oreilly.com/library/view/data-modeling-a/9780071623988/. Access required.

Joel Grus 2019. Data Science from Scratch. Second Edition. O'Reilly Media, Inc. Search date 01.06.2023. https://learning.oreilly.com/library/view/data-science-from/9781492041122/. Access required.

Claus O. Wilke 2019. Fundamentals of Data Visualization. O'Reilly Media, Inc. Search date 13.07.2023. https://learning.oreilly.com/library/view/fundamentals-of-data/9781492031079/. Access required.

Greg Deckler 2022. Learn Power BI. Second Edition. Packt Publishing. Search date 11.09.2023. https://learning.oreilly.com/library/view/learn-power-bi/9781801811958/. Access required.