Middle East Technical University

Department of Computer Engineering

CENG300
SUMMER PRACTICE REPORT

JotForm Yazilim A.Ş.

Hacettepe Teknokent Safir Bloklar C Blok Kat:6

Student's Name:
Ozan Şan

Instructor:
Prof.Dr.Göktürk Üçoluk

September 29, 2019

Start Date: 10.06.2019
End Date: 22.07.2019
Total Working Days: 30

Student's Signature

Organization Approval

Contents

1 Introduction

I have completed my internship at JotForm Yazılım A.Ş., R&D office, at Hacettepe Teknokent, Ankara.
My internship was about Time Series Analysis (TSA), and an implementation of a prediction system, using metrics provided by JotForm. The metrics are numerical, and represents the state (mainly, financial state) of the company. Using SARIMAX modelling, I have made a system which makes predictions about the future, and, when in need, can be exported as a graph, and be further evaluated by the Data Science team of JotForm. This will be further explained in further sections of this document.

2 Project

The project I have completed is called JFM Overseer. Overseer is an application written in Python 3.7, and is capable of predicting time series. Below is a screenshot taken from the application:

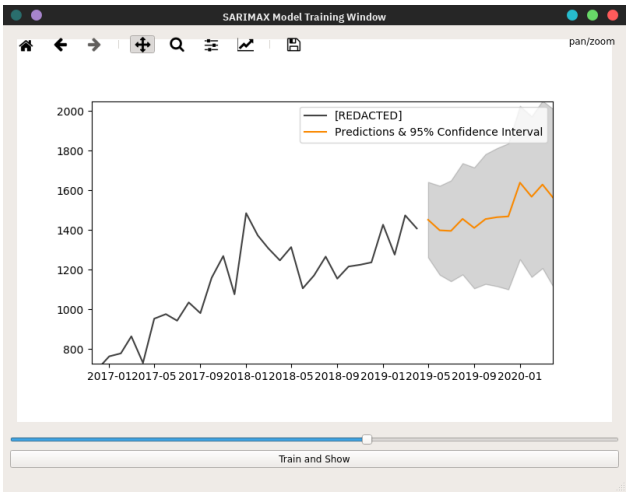


Figure 1: A Screenshot From The Application

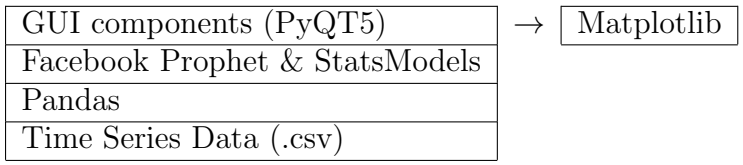
The components and how to set them up are explained in Section 2.8.

2.1 Analysis Phase

Analyzing the problem of efficiently forecasting time series was especially difficult because the texts for this problem is intended for statisticians and economists, but as a future computer engineer, I've found that practical examples on **Kaggle** and other Data Science learning sites were especially helpful.

2.2 Design Phase

The overall structure of the application is as in the following diagram:



In Section 2.8 the visual design phase is explained further.

2.3 Implementation Phase

This is quite a lengthy topic, and it will be discussed further, beginning with Section 2.4 and until Section 2.10.

2.4 Time Series Data Specifications

Overseer is a tool to analyze Time Series data, which must be given to the program as .csv files, with the following structure:

$$\begin{array}{|c|c|} t & y \\ t_0 & y_0 \\ \vdots & \vdots \end{array}$$

In this representation, t must correspond to a date-time value, mainly, parse-able by Matplotlib and other software used. The values for y must be integers.

This lets us create a function $F(t) = y$, which is useful in upcoming definitions. My goal was mainly to produce a prediction, namely, we have to define values m_t, n_t such that for any $t > t_{end}$, where the last observed point of the function F is at time $t_0 + end$, the equation

$$P(m_t < F(t) < n_t) = 0.95$$

must hold. This corresponds to creating a 95% probability interval for the possible observed values of the function F in the future, and means that we have made a prediction. The values m_t and n_t can be averaged to yield a possible future value of F. For making this possible, we must first structure the data in a suitable way.

2.5 Pandas

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.[1]

2.5.1 Installation

Since our app runs best on Anaconda distributions, the suitable way to install pandas is:

```
conda install pandas
```

One can also use, in a plain Python environment,

```
pip install pandas
```

2.5.2 Usage

An example of loading a .csv file, processing it to a uniform format, namely, a "timeseries":

```
import pandas as pd
df = pd.read_csv(FILENAME)
# FBProphet expects the data
# to conform to a format, as such:
df.columns = ['ds', 'y']
# We need to parse the dates
```

```
# to yield date-time objects
df['ds'] = df['ds'].apply(pd.to_datetime)
# After this, df is ready to be processed.
```

If needed, multiple sources can be loaded and processed to yield a single table using Pandas, as if working with a spreadsheet editor.

2.6 Facebook Prophet

Prophet is a forecasting procedure implemented in R and Python. It is fast and provides completely automated forecasts that can be tuned by hand by data scientists and analysts [5]. The process used in automating FBProphet was explained in [6].

2.6.1 Installation

For this package, and with all other packages, it is recommended to use Anaconda distributions.
For an Anaconda distributed Python installation, one can simply use:

```
conda install -c conda-forge fbprophet
```

to install Facebook Prophet. For certain situations (for example, working in a computer with little storage space),

```
pip install fbprophet
```

should still work.

2.6.2 Usage

The usage of FBProphet follows a simple direction, we first load and structure the data for Prophet, import fbprophet and start an instance of the class "Prophet", fit the data and create a future dataframe for Prophet, then see the results for the future dataframe with the help of Prophet. As example:

```
# In previous sections, we have
# already loaded the data in the
# 'df' variable.
import fbprophet as fbp
prophet = fbp.Prophet()
# We then fit the timeseries:
prophet.fit(df)
```

```
# We create a future dataframe
# for seeing the results
# We provide the periods as 12
# for yearly seasonality
future = prophet.make_future_dataframe(periods=12)
# we can now see the results
forecast = prophet.predict(future)
# forecast variable will have
# columns as 'yhat', 'yhat_lower',
# and 'yhat_upper'
```

2.6.3 Possible issues with Facebook Prophet

The developers of Facebook Prophet have their own datetime format for parsing, and this causes issues. In the source code of the package, there is a line:

```
pandas.plotting.deregister_matplotlib_converters()
```

This line of code breaks the system if we are including Prophet after Pandas as I've used **pd.to_datetime** for converting strings of dates into datetime objects, and the solution in my case was adding the following line wherever I've imported Facebook Prophet:

```
pandas.plotting.register_matplotlib_converters()
```

This sadly breaks the plotting functionality of Facebook Prophet, but in my case, as I've done all of my plotting with Matplotlib, it does not really matter.

2.7 Statsmodels

statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.[3]

2.7.1 Installation

As usual, we both have a Conda way and a native Pip way of installing the package.

```
conda install -c conda-forge statsmodels
```

or...

```
pip install -U statsmodels
```

2.7.2 Usage

The usage for this environment is significantly more involved than of Facebook Prophet. First of all, since statsmodels offers a lot of statistical framework, it can simply not operate on the principle "set, and go". We will use a type of modelling in Econometrics called a **SARIMAX** model, short for "Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors model". We need to determine three variables (namely, P, D and Q, for each of the AutoRegressive, Moving-Average and Residual parts) for trying a SARIMAX model, and select the best one for the purpose to display on the screen.
Normally, this process is done by a statistics graduate, or a senior Data Scientist with a professional background, but, in our case, we have found a more adept solution for our problem.
For assessing a model's usability, Akaike's Information Criterion was used. The model which yields the lowest AIC score was to be used.
The following example code, as it is implemented in the project, uses a simple grid search to find the variables P, D and Q. Itertools module facilitates an easy interface for checking all possible combinations within constraints. The following code segments represent how this step was implemented within the project:

```
import itertools
import statsmodels.api as sm
import numpy as np
# We need to limit the computation time,
# So we are only looking on the range [0, 2].
p = d = q = range(0, 2)
# Generating all possible combinations of these variables
pdq = list(itertools.product(p, d, q))
# Seasonal data will be assumed to have the same variables
# Seasonality is set to 12, indicating months.
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in pdq]
resBest = np.inf # Initially, we will assume the worst
for param in pdq:
```

```
  for param_seasonal in seasonal_pdq:
    try:
      mod = sm.tsa.statespace.SARIMAX(time_series,
                                      order=param,
                                      seasonal_order=param_seasonal,
                                      enforce_stationarity=False,
                                      enforce_invertibility=False)

      results = mod.fit()
        if results.aic < resBest:
          resBest = results.aic
          paramreality = param
          paramss = param_seasonal
     except:
       continue
```

After this -lengthy- process is completed, we can take a look at our best results, using:

```
mod = sm.tsa.statespace.SARIMAX(endog = self.time_series,
      order = self.sarimax_parameters[0],
      seasonal_order= self.sarimax_parameters[1],
      enforce_stationarity= False,
      enforce_invertibility= False)
res = mod.fit()
pred = res.get_prediction(start = pd.to_datetime(time_series.index[-1]),
      dynamic = True,
      end = END_OF_PREDICTION)
confidence_interval = pred.conf_int()
```

This typically yields similar results to Facebook Prophet's predictions, but allows for more fine tuning of the model, although it costs more time, it is recommended to run a SARIMAX model on the data even though the Prophet predictions are usable as they are.

2.7.3 Possible Drawbacks of StatsModels

Performing a grid search is not a computationally easy task. A quad-core CPU takes minutes until completion of a model for a range of $[0, 4]$. And

because no multiple-core functionality is defined for SARIMAX models as it seems in Statsmodels, Core 0 of the machine's CPU will be loaded heavily. This can be bypassed by implementing functionality of multiple core execution, but, as this models are enough for monthly resolutions, this was not needed.

2.8 PyQt5 (QT Framework)

PyQt is a set of Python v2 and v3 bindings for The Qt Company's Qt application framework and runs on all platforms supported by Qt including Windows, OS X, Linux, iOS and Android. PyQt5 supports Qt v5.[4]

2.8.1 Installation

```
conda install PyQt5
```

or...

```
pip install pyqt5
```

Appropiate Qt bindings will be located and installed as necessary.

2.8.2 Usage

Usage of the Qt framework is done in two different ways, one of them is coding the GUI elements in Python, and the other is using Qt Designer. For all intents and purposes, Qt Designer is much easier to use when positioning and constraining different elements on the viewport.
After completing a design for a particular window, Qt Designer outputs a file with .ui extension. This can be loaded in Python to define further elements, such as relevant function calls for when certain buttons are pressed, etc. Here is an example of such an usage:

```
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUi
# We inherit a class, QMainWindow
# to be able to have basic functionality
# in our window.
class MatplotlibWidget(QMainWindow):
  def __init__(self):
    # Inherited class is initialized
    QMainWindow.__init__(self)
```

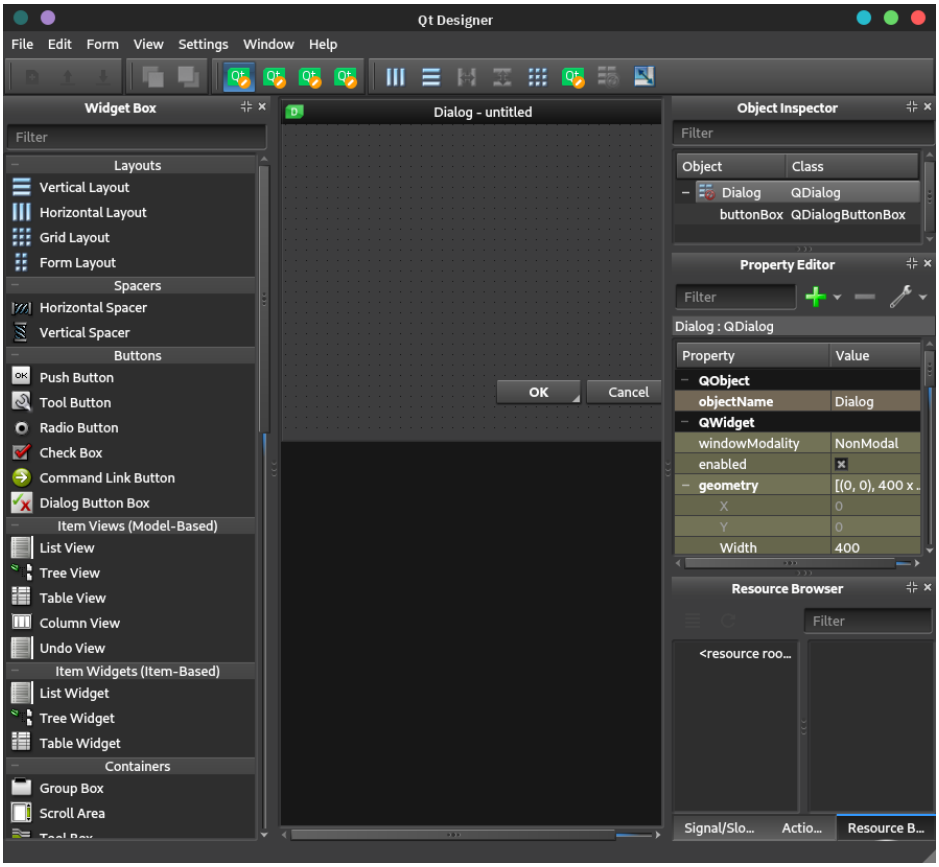


Figure 2: An overview of Qt Designer.

```
#.ui file is loaded
loadUi("mainWindow.ui", self)
# Then, necessary adjustments can be made
self.setWindowTitle("JFM Overseer")
# for example, "pushButton_select_files" is
# a name defined in "mainWindow.ui", and
# QT keeps track of each button.
self.pushButton_select_files.clicked.connect(CALLBACK)
# This can be further examplified.
```

2.8.3 Possible drawbacks of PyQt5

Qt Library has a design choice built into it, and while working with layouts, and not using constraints, it has a tendency to break. Also, grid layouts are not snapping correctly, and this causes further trouble in setting up the Ui.

2.9 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.[2] Qt5Agg is an user interface toolkit supported by Matplotlib, and is used in the project for displaying graphs.

2.9.1 Installation

Matplotlib comes bundled with the Anaconda distributions. For native Python, one can use:

```
pip install matplotlib
```

2.9.2 Usage

A sample usage of Matplotlib is shown below:

```
from PyQt5.QtWidgets import *
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg
from matplotlib.figure import Figure
canvas = FigureCanvasQTAgg(Figure())
vertical_layout = QVBoxLayout()
vertical_layout.addWidget(canvas)
canvas.axes = self.canvas.figure.add_subplot(111)
canvas.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Expanding)
canvas.updateGeometry()
# Inherited from QWidget, __super__ refers to this
__super__.setLayout(vertical_layout)
ax = canvas.axes
ax.clear()
# For a selected time series data as y,
y.plot(ax = ax)
```

```
# For slider functionality,
x_minimum, x_maximum = ax.get_xlim()
y_minimum, y_maximum = ax.get_ylim()
# Variable "aspect" will be provided from the slider
ax.set_aspect(float(abs((x_maximum - x_minimum) /
                        (y_maximum - y_minimum))) *
                        aspect)
self.MplWidget.canvas.draw()
```

2.9.3 Possible issues with Matplotlib

Matplotlib, if not treated correctly, takes the datetime values as integer values. That is why, we can be greeted with a graph with a high slope, and an aspect ratio of 1:12, for example. That is why I've provided the window with a slider. An example is shown below:

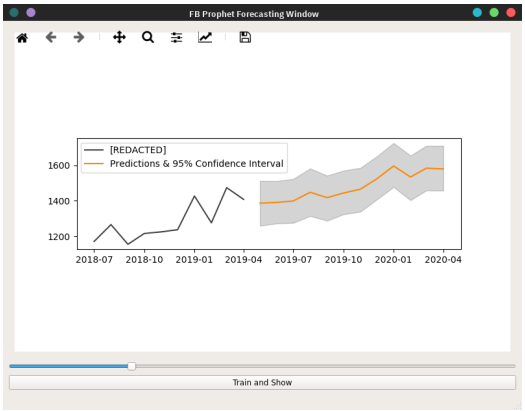


Figure 3: Slider at a position on the left-hand side

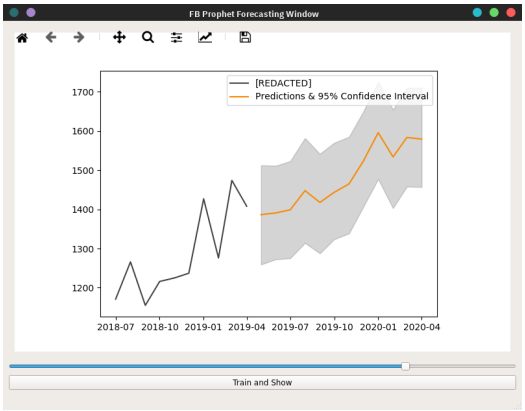


Figure 4: Slider at a position on the right-hand side

2.10 Testing Phase

The software was mainly tested by hand. The problems discussed in Section 2.6.3 were the most time consuming, and issues with Matplotlib components (see section 2.9.3) were resolved with a GUI slider.

On the accuracy of the forecast, we've seen MAPE (Mean Absolute Percentage Errors) while training as in the range of 1% to 2%, but some types of data proved to be extremely difficult to forecast, receiving errors of about 7%. As a computer engineering student with no economics background, these results seemed to be an outlier in the tests, and were left to be fixed by the data science team at the company.
Further tests on the accuracy of predictions were presumably done by the company, but as of time of this report, these tests were not disclosed with us.

3 About JotForm Inc.

JotForm is an international company, having offices in San Francisco, Izmir, and Ankara. Managerial staff is primarily located in Ankara office, which is also the home for the main R&D hub for the company.

Figure 5: A View From JotForm's Ankara Offices

3.1 Structure of the company

In JotForm, employees are operating in teams, named by themselves, such as Amplify, Hermes and so on. The main responsibility of a team is not divided between different products of the company. This way, they can keep their focus, and work more reliably as a team. A team, for example, may work on Form Builder for a certain amount of time, and not in other products such as PDF Editor. This allows for an efficitent **Agile** work environment.
On fridays, all teams from Izmir and Ankara offices of JotForm come together for a meet, called the Demo Day. This facilitates for the presentations of the work done in the week for each team. This day is also the day for us interns to perform their presentations about their projects.

3.2 Products offered by JotForm

3.2.1 Form Builder

Form Builder tool is the main product of the company. Offering a drag-and-drop interface for creating forms, analyzing data easily and making the analysis of the results from the form easier (using JotForm Sheets) is the main focus of the company. Since 2006, the founding of the company, they operate a Form Builder interface, which is active to this day. Main differences from competitors such as Google Sheets is the ease of use, and options for payment such as PayPal, Stripe etc. , which makes the product a standalone candidate for e-commerce websites, handling payments and billing information all by itself.

3.2.2 PDF Editor

In the last year, JotForm offices in Ankara started a project on PDF documents. Since the main focus of the company is on the forms, and customers repeatedly reporting that they need to convert their PDF forms to some online form, a team in Ankara office has built a tool for the purpose. In PDF editor, users can upload their fill-able PDF forms, and automatically transform them into an online form, hosted on JotForm. This feature also works in the other way around, as in, a user can take an existing online form in JotForm, and turn it into a fill-able PDF file format.

3.2.3 JotForm Sheets

Primarily to relieve their customers from needing to resort to Google's services, mainly, Google Sheets, the company developed a tool to deal with Spreadsheet files. This is integrated into their form builder environment, so that the customers can perform aggregate functions similar to Microsoft Excel™, analyze large amount of responses to their forms, and export to useful file formats, such as .csv files.

References

[1] Pandas Community. Pandas description, 2019. URL: `https://pandas.pydata.org/`.

[2] H. John, D. Darren, F. Eric, Michael D., and Matplotlib development team. Matplotlib description, 2012. URL: `https://matplotlib.org/`.

[3] P. Josef, S. Skipper, T. Jonathan, and statsmodels developers. StatsModels description, 2018. URL: `http://www.statsmodels.org/stable/index.html`.

[4] Riverbank Computing Limited. PyQt description, 2018. URL: `https://www.riverbankcomputing.com/software/pyqt/intro`.

[5] Facebook Open Source. Prophet description, 2019. URL: `https://facebook.github.io/prophet/`.

[6] Sean J. Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018. `arXiv:https://doi.org/10.1080/00031305.2017.1380080`, `doi:10.1080/00031305.2017.1380080`.