Capstone Project

Medhat Fawzy

Machine Learning Engineer Nanodegree

Jan 4th, 2023

Definition

Project Overview:

There's no doubt that AI has the potential to revolutionize all sorts of businesses. Telecommunication is no exception. Data analysis and Machine Learning can be extensively applied to the data collected by network operators to further understand networks, predict faults in the cell towers, mitigate congestion in the networks, and identify areas that need new infrastructure.

In this project, I'll be using two crowd-sourced data for the RSRP value and another one for the traffic volume.

A typical 4G network consists of several nodes connected, where each node serves users in the surrounding area. While users are accessing network services, their mobile phones record Key Performance Indicators (KPIs) which can help network operators assess their service quality.

There are many KPIs which evaluate different aspects of the network. One of them is RSRP (Reference Signals Received Power) a KPI measuring network coverage in the user's location. Traffic Volume is another KPI which measures how much data has been consumed by the user.

Problem Statment:

Each dataset has the corresponding KPI measurements collected from mobile phones of different users over a week. It also includes the user's location, operator, phone model and other information. A detailed description of each field can be found in 'DataDescription.xlsx'. The tasks involved are:

- A dashboard containing:
    - A time-lapse of a density map showing how users move during the hours of the day. You should see the density on the map changing according to people's activities every hour. Use the locations in the RSRP dataset.
    - A heat map of user locations for a selected operator with a dropdown menu to choose the operator.
    - A Hex map showing the areas with high downlink traffic for each operator. In this chart, the HexTile size depends on the sum of the Traffic Volume of all users in this area.

    - A Bar chart of RSRP per device type per operator with an option to choose the aggregation method of RSRP (avg, min, max and 90th Percentile).

- Imagine you are running a conversation with one of the operators whose data is in the dataset provided. The discussion is referring to any polygon of your choice that's at least 10km x 10km in size and has enough data samples. Can you help the operator to answer the questions below?
    - 1. Assuming that the coverage next week will improve in that polygon (i.e., RSRP will get better than other operators). What would be the impact on downlink and uplink traffic volumes?

    - 2. Samsung devices are the main handsets in our network. Can you predict the traffic volume growth, uplink and downlink, over time for these devices and compare it to the competitors?

Metrics:
The dashboard must have these four plots, three of which are maps and the last one is a bar chart. These are the main requirements in the dashboard and they must be completed.
To assess the impact of the RSRP value on the traffic volume, a regression model will be used, and for that model, I'll use the R^2 metric. Here, I'm studying the impact of one variable over another one. This is a classic regression problem and for this type of analysis, the R^2 metric is the best to assess the model.
For the prediction of the traffic volume growth, I'll use a holdout sample of the most recent data points to test our model. This is the most common way to assess a time-series model.

<u>Analysis</u>

Data Exploration:
The RSRP dataset contains around 3 million data points (2725353 rows) and has 11 columns. The dataset is collected across a span of 4 days. The dataset is from three operators. Looking at the country column, it seems that all these data points come from KSA. Another thing to notice here is that there are a lot of device manufacturers like Samsung, Lenovo and Huawei. We don't seem to have any iPhone devices here. This is because iPhone devices don't send this type of information to network operators. There are 7 columns of type categorical, two are floats, one is int and the first is DateTime type.

Exploratory Visualization:

Looking at the histogram plots of the RSRP valid values and the Traffic values (shown below), we can conclude the following:

- The RSRP values are distributed in a normal bell curve way (fig 1).
- The traffic volume values are highly skewed whether the uplink traffic or downlink traffic (fig 2).

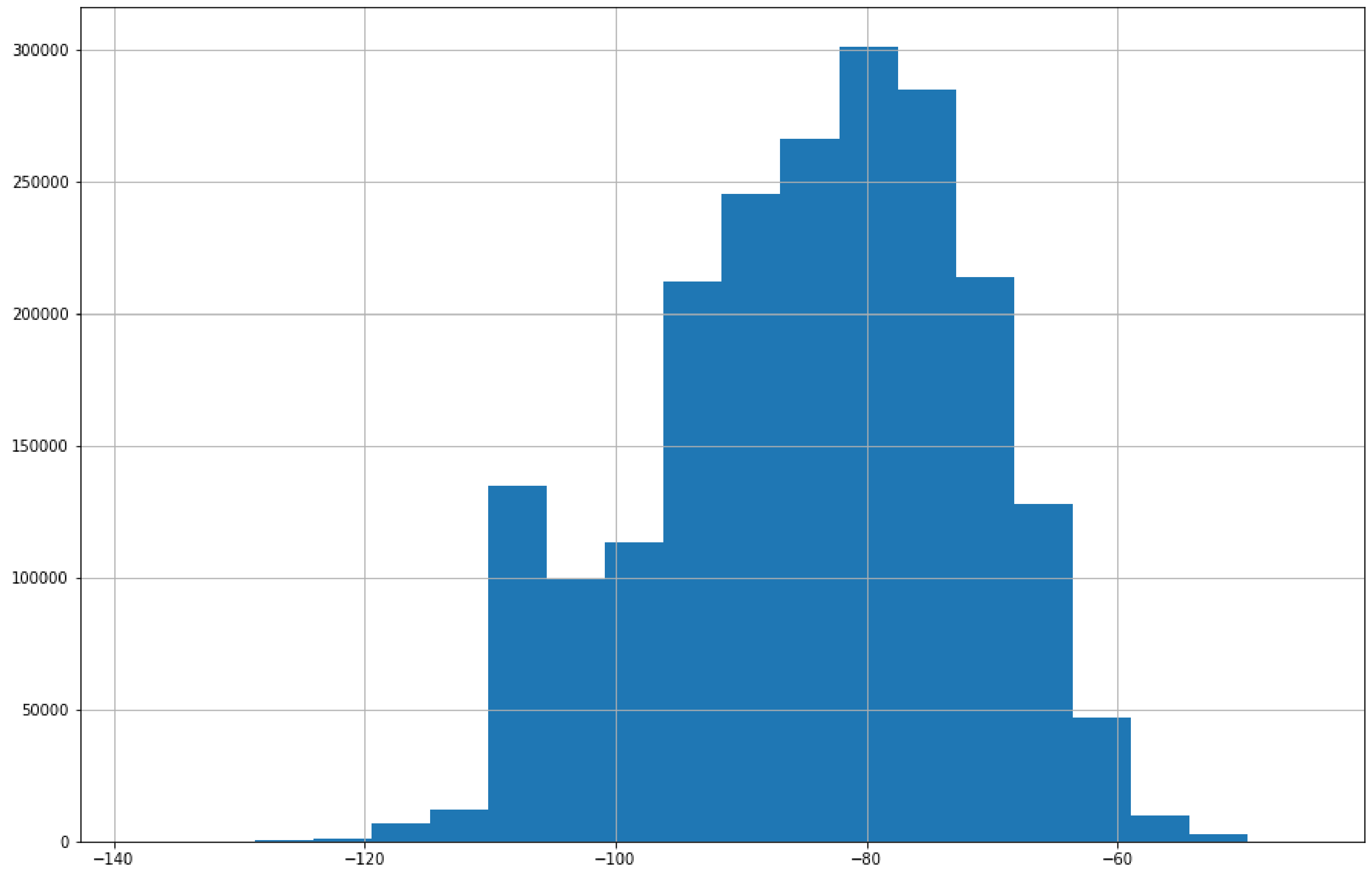


Fig 1: The histogram of the valid RSRP values

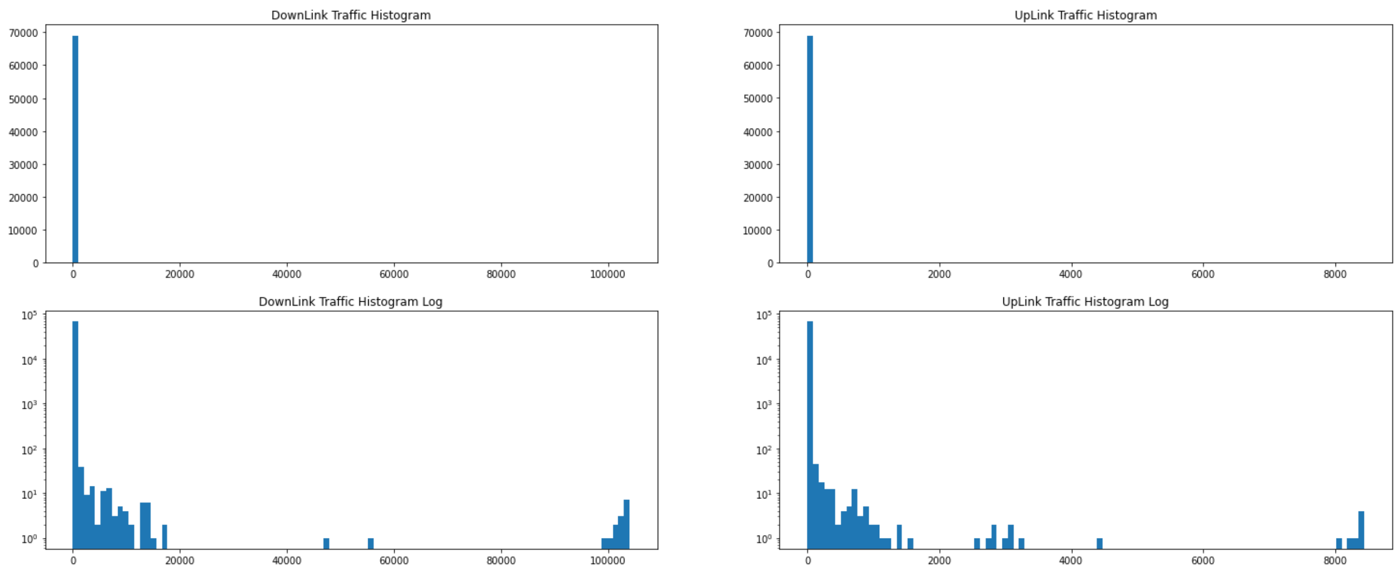


Fig 2: Histograms of the traffic volume for uplink and downlink. The bottom two plots are the logarithmic ones.

Algorithms and Techniques:

The Dashboard:

The dashboard was created using the Holoviz framework. Holoviz provides a set of libraries to help create the most dynamic plots and to also create and deploy highly-customizable dashboards. For this project, I used Holoviews, Datashader, Colorcet and Panel libraries to create the dashboard.

Holoviews is used to create many different types of dynamic plots. In the backend, it uses other libraries like Bokeh and Plotly and provides the option to choose which one. I use bokeh as a backend because it's more integrative with the rest of the libraries I'll be using. I used it to create the Hex Map and the bars plot, also with the rasterization method I used to plot these huge datasets.

Given this huge dataset, one's faced with the overplotting issue where the plots of these data points look like a mess because the samples are being projected over each other over and over, and even setting a low alpha value, doesn't help much. For this reason, I'll be using a rasterizing technique provided by Datashader. The points are projects in a canvas where each pixel acts like a bin in a histogram. Pixels with a high count of points are coloured with a high value of the colour map we will use. This will make the map look like fig 3 instead of fig 4.

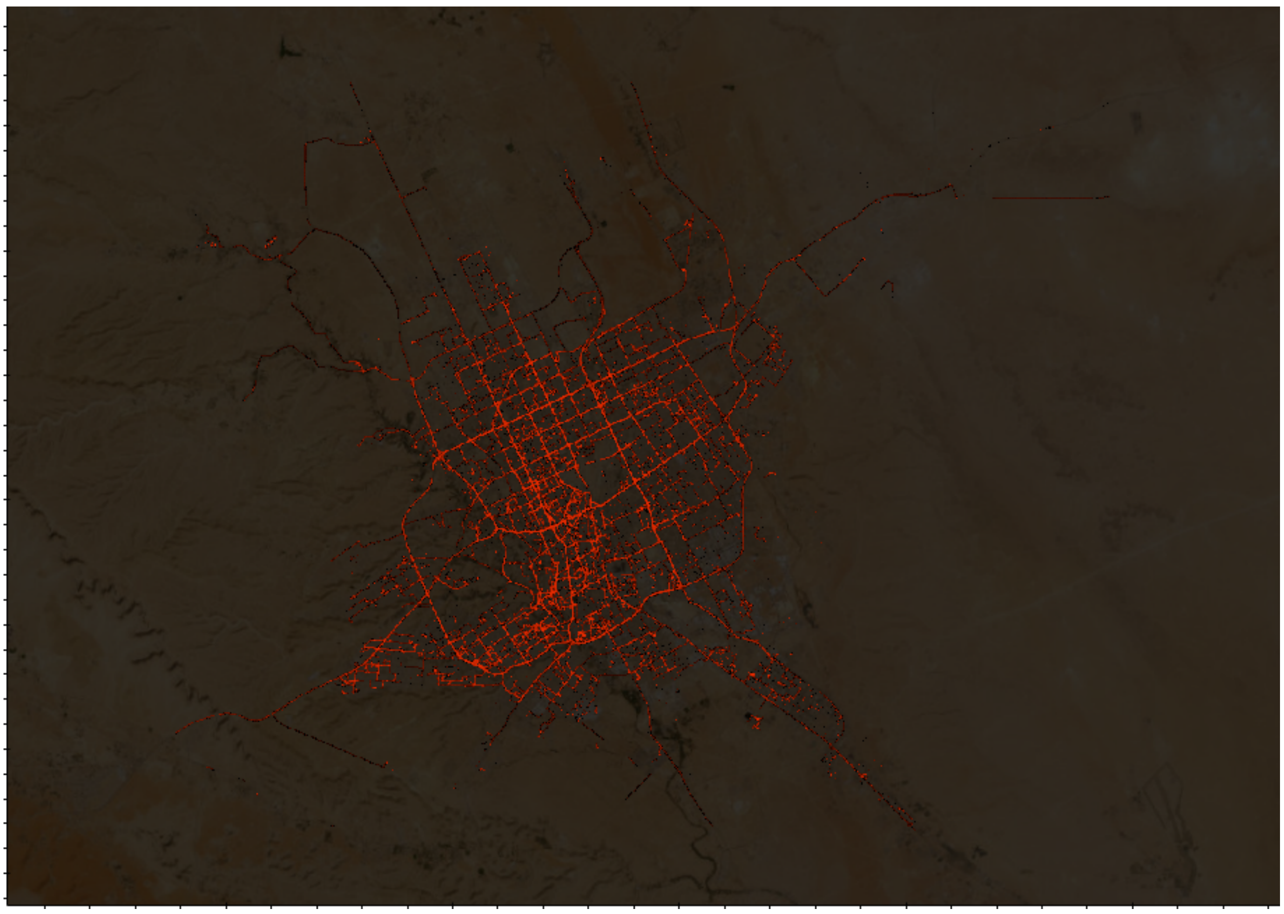


Fig 3: A map of our RSRP data points generated by Holoviews.

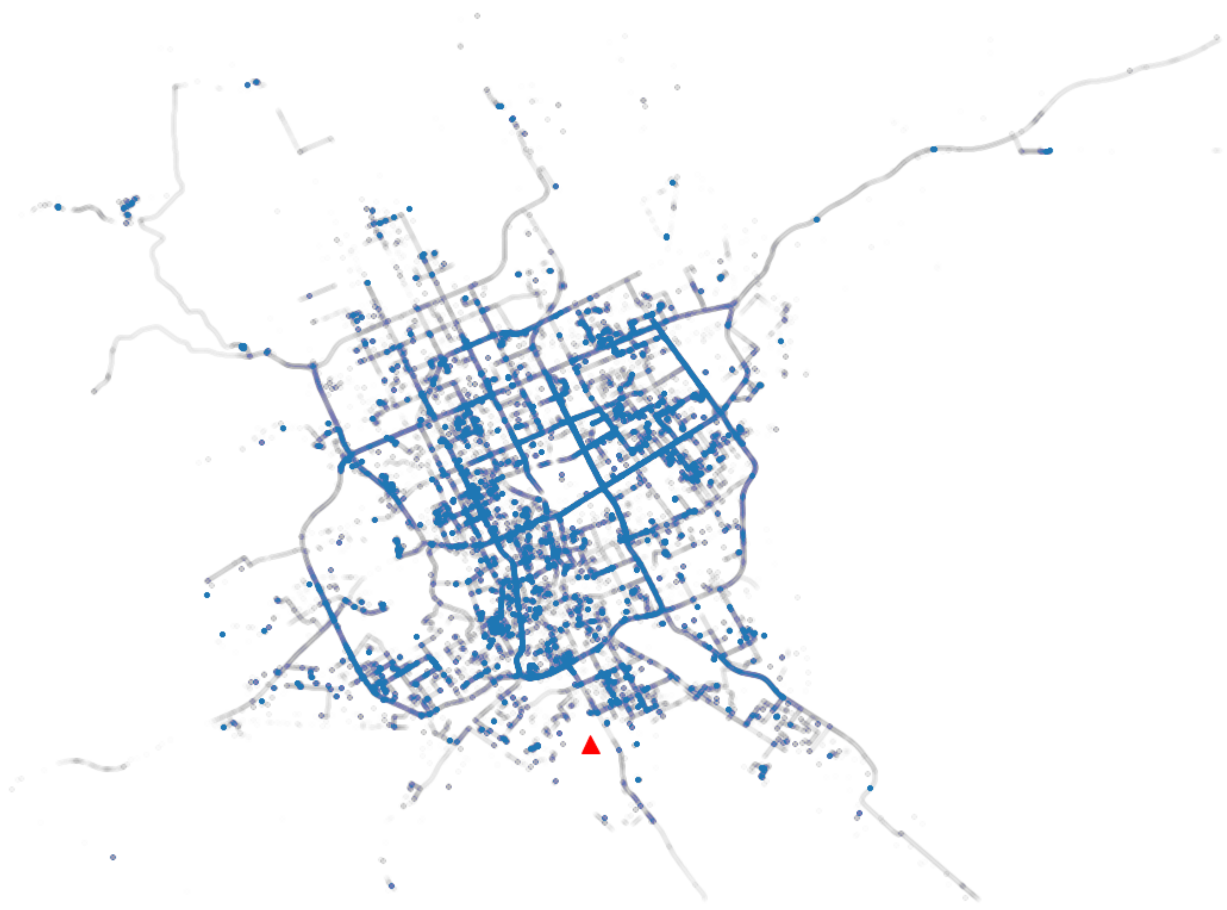


Fig 4: The same map generated by Matplotlib

It’s not just the rasterizing technique, the map in Holoviews will have hover and zoom-in functionalities.

The Prediction and Regression:

To predict the impact of RSRP improvement on the Traffic Volume, I used two regression models: Linear Regression and RandomForrestRegression.
In this problem, there are two variables, and the goal is to study the impact of one of them over the other. This is a classic linear regression problem. The output of the linear regression model will be the value of the dependent variable given the independent variable.

Benchmark:

I used the R^2 metric to check the accuracy of our model, optimally I was expecting at least an 80% accuracy. This number depends on the business objective and how the stakeholders will use our predictions.

<u>Methodology</u>

Data Processing:
The preprocessing for the two datasets, RSRP and TrafficVolume, is done in the notebooks '1.RSRP.ipynb' and '2.Traffic_Volume.ipynb' respectively.

In the RSRP notebook, the preprocessing was done in these steps:
1. The category columns are converted to category-type columns.
2. The 'Country' column is dropped.*
3. The values in the 'DeviceManufacturer' column that are the same but different in capitalization are replaced with one value.
4. The 'Timestamp' column is converted to the DateTime type.
5. The dataset is then sorted by time.
6. The duplicates are then dropped.
7. The processed dataset is saved for later usage.

*Note: the 'Country' column was dropped because all the values in it are the same.

In the TrafficVolume notebook, the preprocessing was done in these steps:
1. The category columns are converted to category-type columns.
2. The 'Country' column is dropped.
3. The 'Timestamp' column is converted to DateTime type.*
4. The empty rows in the timestamp are removed.
5. The dataset is then sorted by time.
6. The data is grouped using all the columns and by taking the average of the 'TrafficVolume'.
7. The new dataset columns are processed again as we did previously.
8. The data is split into uplink traffic and downlink traffic.
9. Some visualisation was done to see how skewed our datasets are and if there are any outliers.
10. The processed TrafficVolume dataset is saved for later usage.

*Note: while converting the 'Timestamp' column some values were invalid, these should be dropped after the conversion.

After preprocessing the two datasets in hand, the joining step comes.
In the '3.Merging_Data.ipynb' notebook, the preprocessing was done in these steps:
1. Checking that there are no duplicates.
2. The invalid RSRP values are dropped.
3. The traffic volume data is grouped based on all the columns and the average of the traffic volume.

4. The resulting dataset columns are then reconverted again.
5. The two datasets, TrafficVolume and RSRP, are then merged together into a single dataset based on the columns : ["LocationLongitude", "LocationLatitude", "RadioOperatorName", "RadioNetworkGeneration", "RadioConnectionType", "RadioMobileDataEnabled"]
6. Then the resulting dataset is filtered based on the values of the timestamp on the first dataset and the values of the timestamp in the second one. RSRP is sampled at random timestamps while TrafficVolume is sampled every 15 minutes. The dataset will be filtered based on whether the RSRP timestamp falls in the 15 min range of the traffic volume.
7. The dataset columns are then processed.
8. The dataset is sorted by the timestamp of RSRP and the index is reset.
9. The timestamp of RSRP is dropped.
10. The dataset is then grouped based on all the columns and the average of the RSRP because some duplicates were found.
11. The dataset is split into uplink and downlink and then processed and saved for later usage.

Prediction:

The prediction process can be split into two main stages:

1. The regression model training stage.
2. The time-series forecasting stage.

During the first stage, the regression model was trained on the preprocessed training data. This was done in a Jupyter notebook (titled “4.Machine_Learning.ipynb”), and can be further divided into the following steps:

1. The Downlink and Uplink traffic datasets are loaded into memory.
2. The datasets are checked to make sure they are in the right format.
3. For the linear regression task, a polygon must be chosen, so we can perform the regression on it.
4. The data must be first scaled before we fit the linear regression model on it. I’ll use the MinMaxScaler.
5. After fitting the model, The accuracy is calculated.
6. After fitting the LinearRegression model, I’ll fit another model which is the RandomForestRegressor.
7. The data is first split into test and train datasets.
8. The data is then reshaped to comply with the model input required format.
9. The RandomForestRegressor model is then fitted on the training set and tested using the test set.

10. Conclusions are then made about the correlation between RSRP and traffic volume.

During the second stage, a time series will be created using the available data then forecasting models will be used, I'll use Pycaret and the Prophet model. This is done in the following steps:

1. First, I'll get the 99.5 % percentile of the Traffic Volume in the uplink and downlink devices.
2. The datasets will then be grouped and only the timestamp, traffic volume and operator columns will be kept, while of course only choosing the rows that have Samsung as a device manufacturer.
3. Preprocessing will be made and then each dataset will be split according to the number of operators we have.
4. The resulting datasets will then be resampled into 15 minutes periods and the missing values will be interpolated.
5. The datasets will then be plotted to study their characteristics, whether they have seasonality, trend or they are just white noise.
6. The ACP and PACP functions are plotted as well.
7. After plotting, the Pycaret package is used to train and fit several models on the data and to choose the best one.
8. We plot the prediction on the test set and see how it performs. We also try to forecast some feature values.
9. After using the Pycaret, we use the Prophet model. The data columns must be renamed before fitting the model to the data.
10. After the preprocessing step, the model is fitted and used to forecast data.
11. The model output is plotted and we look at its accuracy.
12. Conclusions are made.

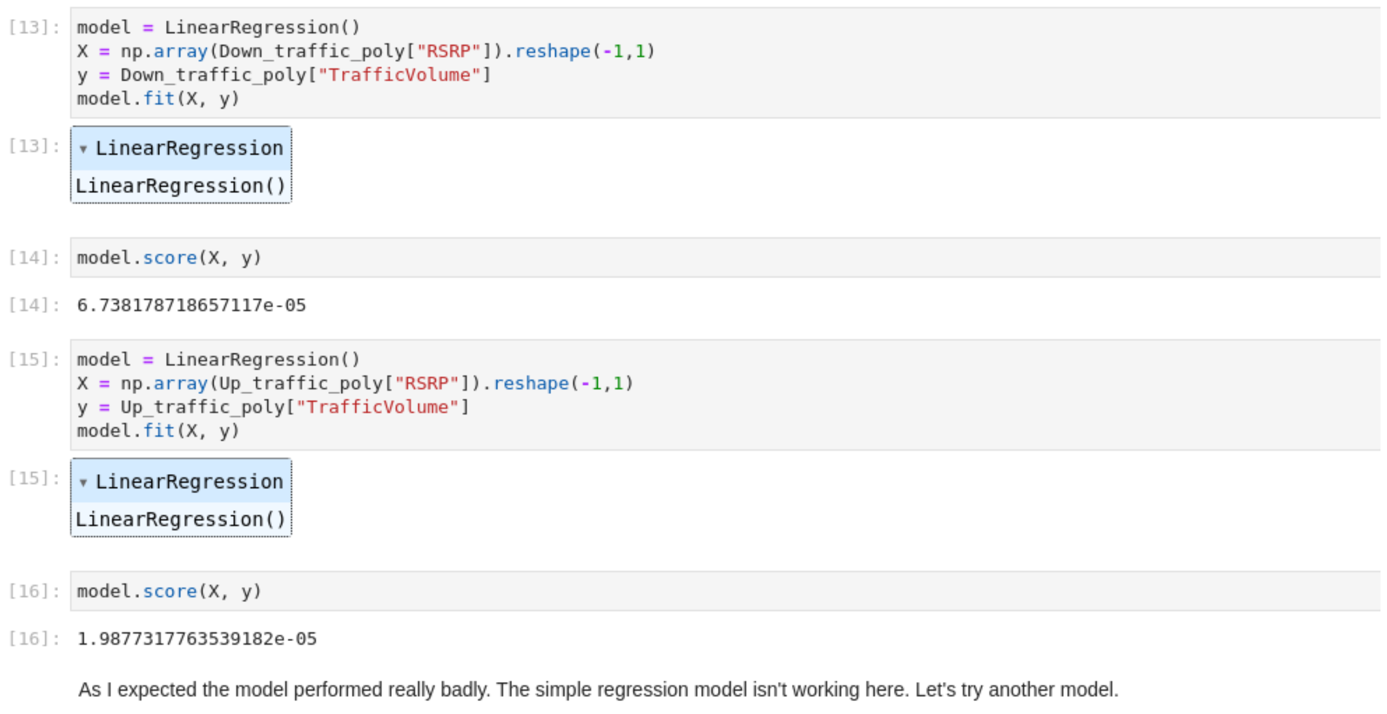
Refinement:
The two previous methods were automatic and not a lot of manual tuning was done. In this stage, I'll use the ARIMA model and do then start tuning the model and see how it fits the data. The ARIMA model takes 7 parameters p,d,q for AR, I, and MA parts and P, D, Q, s for the seasonal component of the model.
This will be an iterative approach where I try different combinations of these parameters and using some metrics like R^2, RMSE and MAE I'll judge the model fitness of the data. Some plots will be used as well to judge the fitness of the data like scatter plots and line plots.

Results

Models Evaluation and Validation:

For the regression models, I used a test set to train the RandomForestRegressor and to evaluate both models using the R^2 metric.

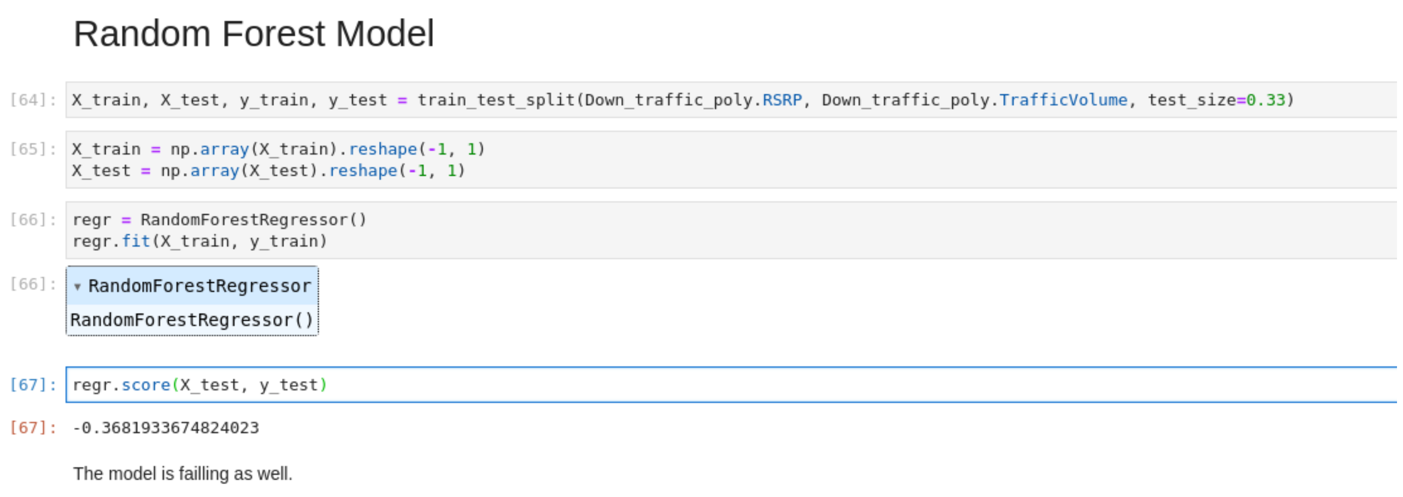
Both models showed that Traffic Volume isn't dependent on the RSRP value.



```
[13]: model = LinearRegression()
      X = np.array(Down_traffic_poly["RSRP"]).reshape(-1,1)
      y = Down_traffic_poly["TrafficVolume"]
      model.fit(X, y)

[13]: ▾ LinearRegression
      LinearRegression()

[14]: model.score(X, y)

[14]: 6.738178718657117e-05

[15]: model = LinearRegression()
      X = np.array(Up_traffic_poly["RSRP"]).reshape(-1,1)
      y = Up_traffic_poly["TrafficVolume"]
      model.fit(X, y)

[15]: ▾ LinearRegression
      LinearRegression()

[16]: model.score(X, y)

[16]: 1.98773177635391982e-05
```

As I expected the model performed really badly. The simple regression model isn't working here. Let's try another model.

Fig 5: The Linear Regression Model



Random Forest Model

```
[64]: X_train, X_test, y_train, y_test = train_test_split(Down_traffic_poly.RSRP, Down_traffic_poly.TrafficVolume, test_size=0.33)

[65]: X_train = np.array(X_train).reshape(-1, 1)
      X_test = np.array(X_test).reshape(-1, 1)

[66]: regr = RandomForestRegressor()
      regr.fit(X_train, y_train)

[66]: ▾ RandomForestRegressor
      RandomForestRegressor()

[67]: regr.score(X_test, y_test)

[67]: -0.3681933674824023
```

The model is failling as well.

Fig 6: The Random Forest Model

For the time-series models, a holdout sample of the most recent data points was used to evaluate the models created. I have used three methods of forecasting on this dataset: Pycaret, Prophet and ARIMA. The first two methods are automatic, the last one was a manual approach I used because the first two failed.
All of the models used didn't perform well when evaluated, meaning that given the dataset at hand, the forecasting process isn't possible. Check fig 7, fig 8, fig 9

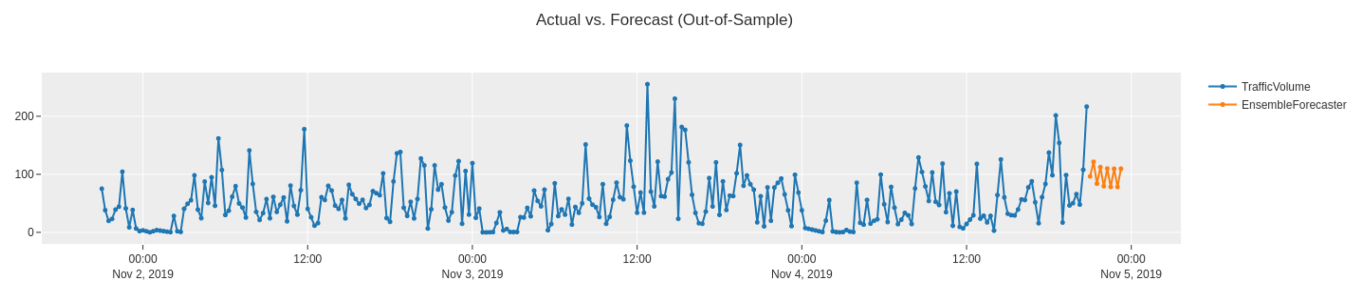


Fig 7: The Pycaret ensemble model

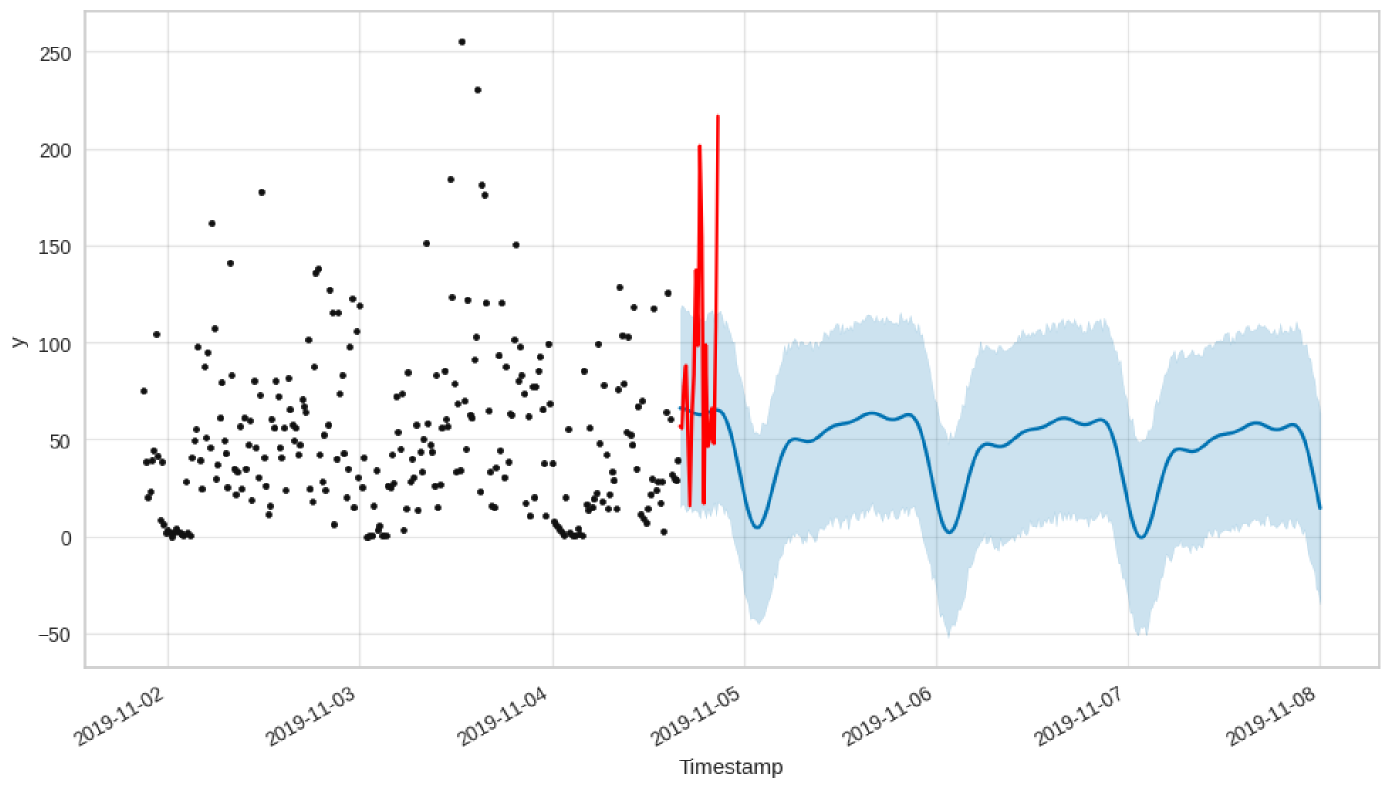


Fig 8: The Prophet model

```
: plt.scatter(Down_traffic_samsung_A, prediction);
```

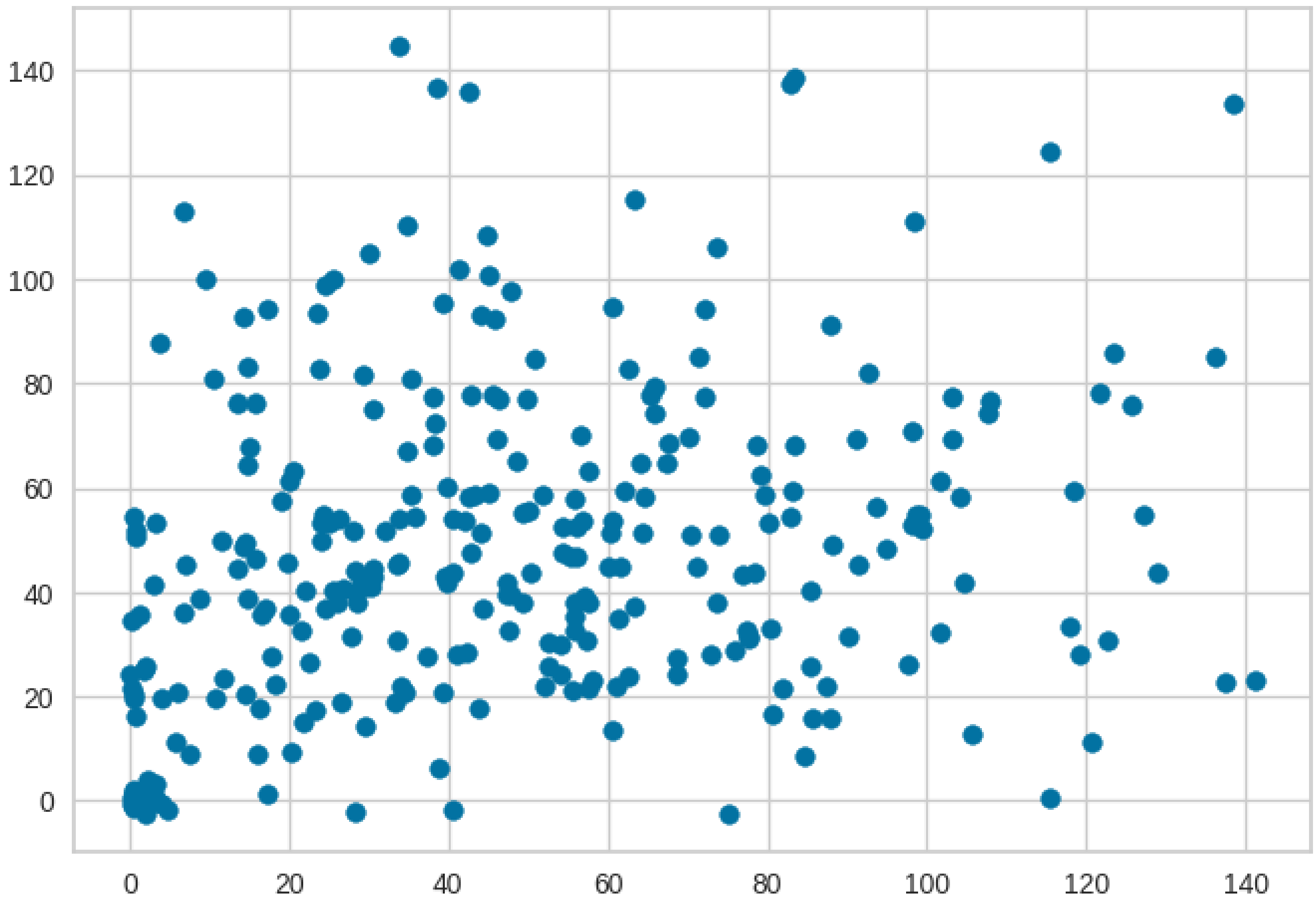


Fig 9: A scatter plot of the Traffic volume and the prediction from the ARIMA model

Justification:

After trying 3 different methods, including an iterative manual method, no good accuracy was achieved are the accuracy of our models was no better than a random guess. No model achieved an R^2 value higher than 0.8 or even 0.5 meaning that these models can't be used to forecast future values or predict any values. Looking back at the data at hand we see that we might need more data over a longer time span. In fig 10 I included the results of the ARIMA model.

```
: score = mean_squared_error(Down_traffic_samsung_A["TrafficVolume"], prediction, squared=False)
  score

: 39.99167320944451

: score = r2_score(Down_traffic_samsung_A["TrafficVolume"], prediction)
  score

: -0.3686179493611026

: score = mean_absolute_error(Down_traffic_samsung_A["TrafficVolume"], prediction)
  score

: 30.641393590753935
```

Fig 10: The R2, RMSE and MAE of the ARIMA models.