Coursera

Applied Data Science Capstone Project

Final Assignment

"Predicting the Transport Cost of Individual Shipments"

Rogelio Elizondo

2021-02-19

Introduction

A logistic service provider usually performs as an intermediary party between customers who need to transport their orders and carriers who have the necessary equipment and resources to transport those orders. The process of assigning a customer order to a carrier is not always straightforward and might include several days of quoting and negotiation between all involved parties. On the one hand, the customers would like to receive an immediate cos estimation to the submitted transport orders in order to properly plan their financials. On the other hand, the carriers not always have straightforward tariffs which might include several clauses regarding shipped volume, weight, etc.

The logistic dispatcher is the employee from the logistic service provider company who receives customer orders and must assign a cost to the transportation of those orders. In order to do so, the dispatcher must gather relevant shipment data and request quotes from several potential carriers. Usually the dispatcher is not equipped with proper tools to simplify this job, and he must contact multiple carriers before finding out a quote which is acceptable for the customer. On top of that, assigning a cost is usually a manual process where human decisions are also involved. In the meantime, the customer would appreciate having an immediate response to the submitted orders in order for them to decide if the price is fair or not.

The process of assigning a preliminary cost to a shipment transport can be drastically accelerated by the use of a machine learning model. A regression model could be built to predict the cost of a shipment given its relevant attributes. Such attributes include origin and destination, weight, volume, date, etc. The objective of this mini project is to build a regression model able to help the dispatchers in predicting the cost of a transport shipment.

Data

In order to build a regression model as such, historic data is required. A logistic service company provided an historic data set for this purpose. The raw data set consisted of shipment data for the previous years and included the following columns

- Origin latitude
- Origin longitude
- Destination latitude
- Destination longitude
- Weight
- Loading meters
- Dangerous goods flag
- Shipping date
- Cost

The data set consisted of 9 columns and over 250K rows where each row represented an individual shipment. The first four columns reference to geographical locations, however the points were altered to maintain confidentiality. The weight and loading meters columns included float values which were previously normalized. Furthermore, the dangerous flag column had a Boolean value. Finally, the shipping date was provided as a string of the format YYYY-MM-DD.

Methodology

Exploratory Data Analysis

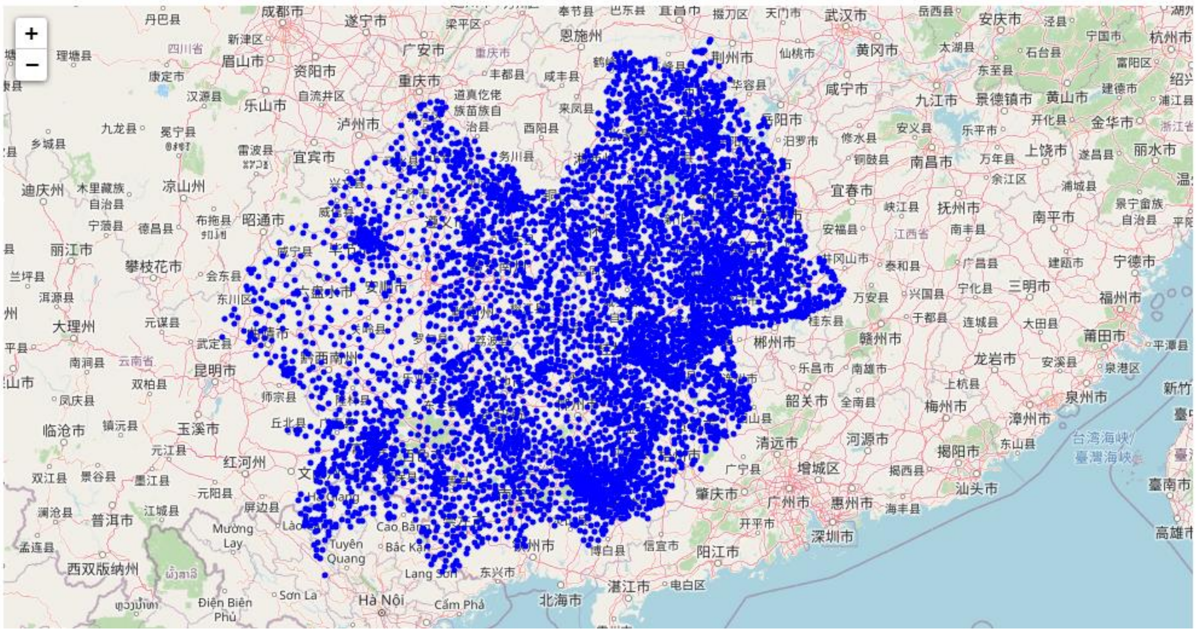
EDA was performed to understand more about the provided data. The raw data included 8 columns describing the shipment and 1 column assigning the cost. The cost is our target variable. In the following sections, each of the other columns will be further analyzed and described. An additional column was created to categorize the shipment cost, which will help to understand the impact of other columns over cost. The cost category was coded as follows.

```
data['CostCat'] = 'Note Categorized'
data.loc[data['cost'] <= 0.25, 'CostCat'] = 'Low'
data.loc[(data['cost'] > 0.25) & (data['cost'] <= 0.5), 'CostCat'] = 'Medium'
data.loc[(data['cost'] > 0.5) & (data['cost'] <= 1) , 'CostCat'] = 'High'
data.loc[data['cost'] > 1, 'CostCat'] = 'Expensive'
data['CostCat'].value_counts()
```

```
Medium       161716
Low           63846
High          25425
Expensive       168
Name: CostCat, dtype: int64
```

Geo Locations

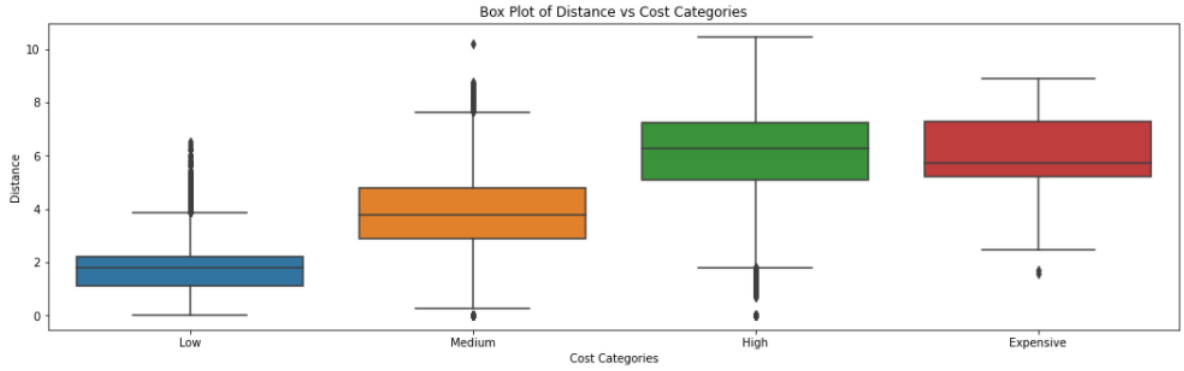
The first 4 columns represented the latitude and longitude values of the origin and destination of the shipment. It was known upfront that their values were shifted to protect data confidentiality. A folium map was created to visualize the region of such points.



The image above points out that the locations fall in China and the north part of Vietnam. Furthermore, the shape of the points resembles the shape of Germany, which might be the original source. In case we had the original points, calculating driving distances and even flagging national and international shipments could had gave us further insights. However, since we know that is not the case, we stick to a simple Euclidean distance, which was calculated as follows.

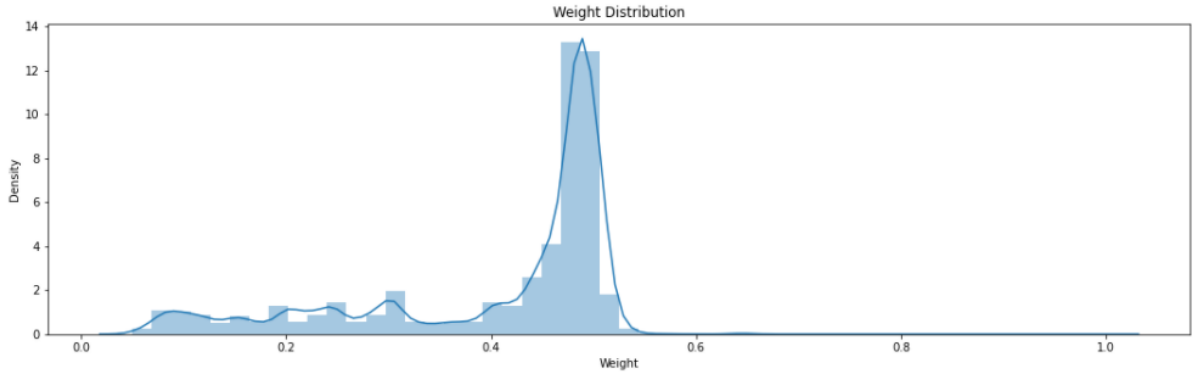
```
def my_euclidean(row):
    a = row['origin_latitude']
    b = row['origin_longitude']
    x = row['destination_latitude']
    y = row['destination_longitude']
    d = math.sqrt((a-x)**2 + (b-y)**2)
    return d
```

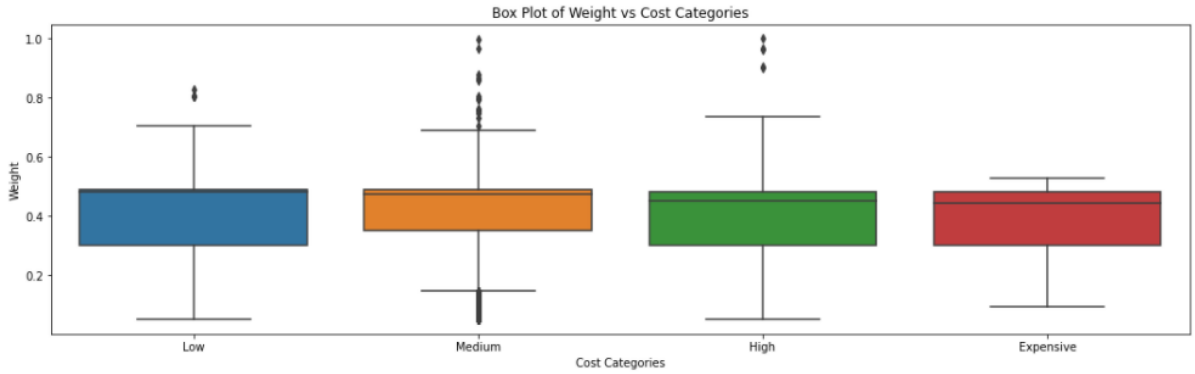
Combining the calculated Euclidean distance with the cost category column we can confirm that distance definitely impacts the shipment cost, as showed in the following boxplot. In the figure we can easily distinguish between the first 3 cost categories. However, high and expensive shipments seem to be not distinguishable from each other.



Weight

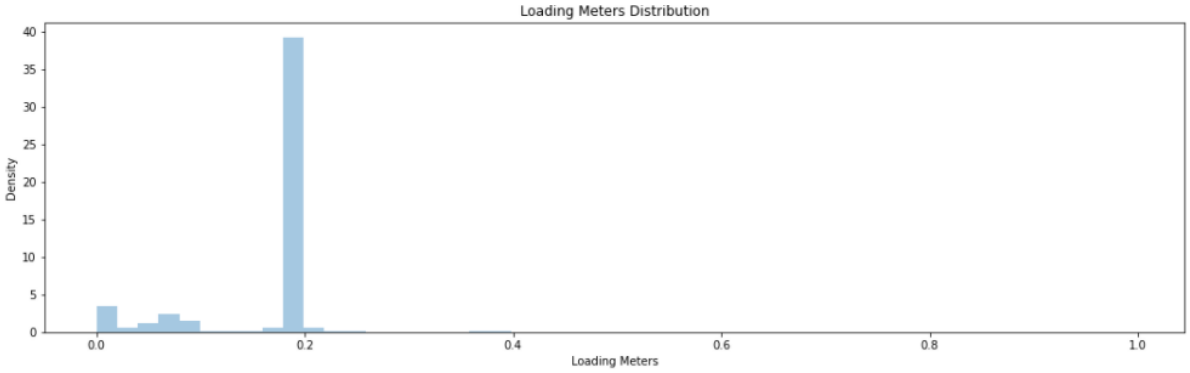
The next column for our analysis is shipment weight. In this case, weight is strongly concentrated around a value of 0.884 with almost all of its values falling below 0.6. The boxplot shows us that weight by itself is not enough to distinguish among the cost categories.





Loading Meters

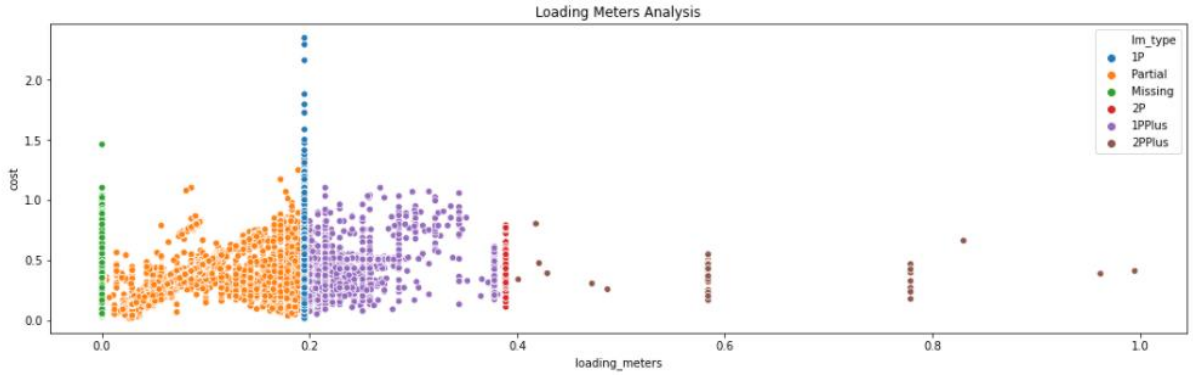
The next column for analysis is loading meters. This time, its value is even more concentrated than weight itself, around the value of 0.195. There are also several records with a value of 0.000 value which could represent missing measurements. An interesting observation was noted when plotting the loading meters against the cost. There seem to be certain factors of 0.195. In the logistic industry, is common that shipments simply occupy one pallet and the fact of having factors of 0.195 only strengthens this idea. For that reason, a categorical value was created from the loading meter float column. The categorical value, distribution and scatter plot are showed below.




```
def my_lm_category(row):
    lm = row['loading_meters']
    if lm == 0.0: return 'Missing'
    elif lm < 0.195: return 'Partial'
    elif lm == 0.195: return '1P'
    elif lm < 0.389: return '1PPlus'
    elif lm == 0.389: return '2P'
    elif lm > 0.389: return '2PPlus'
    else: return 'Other'

data['lm_type'] = data.apply(my_lm_category, axis=1)
data['lm_type'].value_counts()
```

```
1P         191700
Partial     37076
Missing     17209
1PPlus       4525
2P            565
2PPlus         80
Name: lm_type, dtype: int64
```

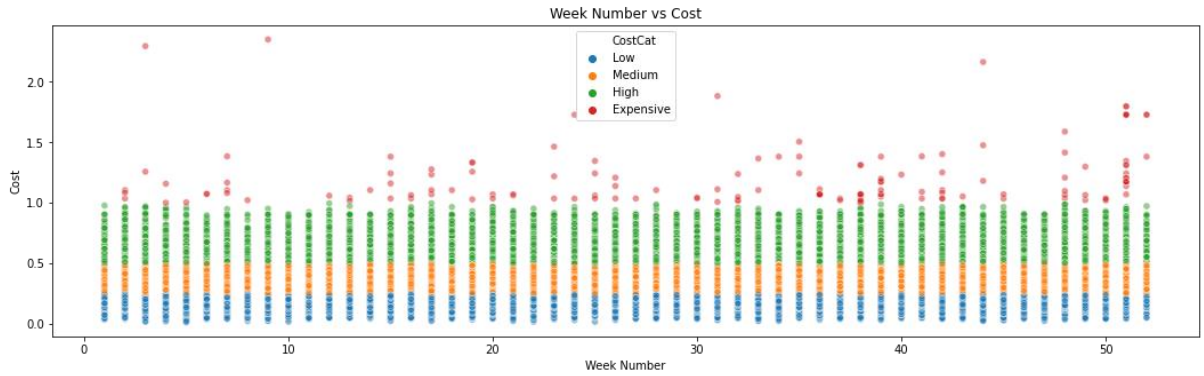


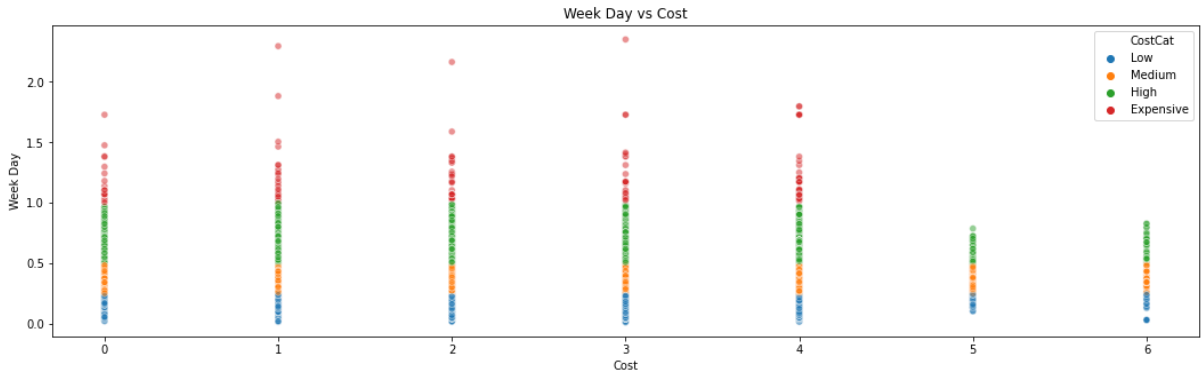
Dangerous Goods

The next column for analysis is dangerous goods. Unfortunately, there is not much to discover from this column itself. The idea behind it is to distinguish between those shipments which include dangerous goods to those who do not. However, all the rows in our data were marked as a Boolean true. This unfortunately states that the column is not useful for our regression and for that reason it was dropped.

Shipping Date

The shipping date column was given with a format of YYYY-MM-DD. It covered the time period from beginning of 2017 to early 2019. Several ideas could come into place regarding associating cost and date. On the one hand, one could imagine that certain seasons are more expensive than others, for example Christmas. In the other hand, shipping on the weekends is usually also more expensive than shipping during the week. The following two plots visualize such ideas. However, in both cases, the hypothesis was wrong. There was only a slightly increase in shipment cost towards the end of the year and the expensive shipments never fall into the weekend.





Modeling

We are able to produce a pre-processed data set after EDA. Once again, using the regression approach we will predict the shipment cost. The features we will use to predict the cost are:

- Distance
- Weight
- Loading meters
- Loading meters type
- Year
- Month
- Distance Weight
- Distance Weight Squared

From the above columns, distance, loading meters, year and month come directly from our raw data. Year and month are extracted from the shipping date original column. Furthermore, the loading meters type is a categorical column created from the loading meters column which has floats values. Finally, the last two columns are manually created under the assumption that distance and weight have a interaction that impacts costs.

Since we have several categorical columns in our pre-processed data, we need to encode them. For this we used the OneHotEncoder class from Scikit Learn. This basically translates our categorical columns into many Boolean columns, which aid the regression model. The class train_test_split also from Scikit Learn was used to split between training and testing data.

Five different models were built, all of them from the Scikit Learn library. A simple linear model, ridge regression, lasso regression, decision regression tree and random forest regressor were used. Cross validation, also from Scikit Learn was used during the training process for evaluation. Finally, the r^2 was computed for all models on testing data.

Results

The pre-processed data was divided into 70 percent training and 30 percent testing data. The results of the computed r^2 on testing data are showed below.

```
Score on Test Data
Linear Model Score      :   0.6441968929974466
Ridge Regression Score  :   0.6441893968847746
Lasso Regression Score  :   0.6441983495419472
Decision Tree Score     :   0.6647958139112322
Random Forest Score     :   0.7894916212306891
Random Forest will be used as Model!
```

The performance of the first four models was very similar. The random forest clearly outperformed the rest of the models and therefore should be selected as the model to perform predictions in the future. The r2 achieved by it, 0.789, can be considered as good but still far from excellent.

Discussion

The trained random forest model already aids the dispatchers with getting “immediately” a good estimate of the shipment cost given certain shipment attributes. The following bullet points brainstorm on several recommendations and points to improve the model performance

- If real latitude and longitude points were given, a proper driving kilometer distance could be computed. Since distance is one of the main drivers of shipment cost, this should improve the model.
- In logistics, it is common to have a fixed cost for a from postal code to postal code combination. This could also be achieved if the raw data included addresses, or at least postal codes. The reason behind this idea is that several routes have a higher shipment traffic and therefore can achieve lower kilometer rates.
- The random forest model clearly outperforms the rest of the tested models, however there are other models that could be tested as well. For example, a gradient boosting or light GBM might improve model performance.

Conclusion

The objective of this mini project was to aid logistic dispatchers in their task of supplying an approximated shipment transport cost immediately to their customers, without having to interact with the carriers. In order to do so, several regression models were trained on shipment attributes, such as distance to be transported, weight, loading meters, year and month. Five regression models were tested: simple linear regression, ridge regression, lasso regression, decision tree regressor and random forest regressor, all of them from Scikit Learn. The random forest regressor outperformed the rest of the models, reaching an r2 of 0.789 which can be considered as good.

With the built model, the dispatchers would be able to immediately predict the shipment transport cost whenever they provide the same shipment features. An intermediate step should be scripted to preprocess and encode the data in the same way.

As a final remark, the random forest performance is acceptable but could still be improved. Main ideas behind improvement relate to a much better understanding of the geographical data involved and the use of more complex random forest type approaches.