Machine Learning Project – Predicting Fire Brigade Notional Price

CS7CS4, Trinity College Dublin, 2021 - 2022

MOHAMMAD MAHDI ESLAMI
[EMAIL]
[ID]

MISBAH RIZAEE
[EMAIL]
[ID]

ANKANA BHATTACHARJEE
[EMAIL]
[ID]

1. *Introduction*

London Fire Brigade is forced to allocate a large number of fire-fighters as well as pumps around London on a daily basis. This is a time-consuming process for the LFB[1], as well as a real grapple for fire-fighters.

In this project, we focused on working with the LFB's large dataset to understand how we could enhance this process with the help of machine learning. Our primary goal was to find the major elements that influence the cost of the fire events using a machine learning method that works best with our dataset. We discovered a number of ways that machine learning may increase awareness inside the organization and with a hope that it would help them make better decisions. In our project, the data analytics were performed in two ways: machine learning and statistical analysis. We first began by cleaning the dataset. This process, explained in section 2, involved removing some unnecessary records as well as selecting the most impactful features. We then applied some regression methods e.g Linear regression, svm, lasso, ridge and k-nearest neighbors to predict the price of fire brigade service. The performance comparison is available in section (5).

2. *Dataset and Features*

The data for London Fire Brigade Incident Records were downloaded in CSV format from the London Datastore website[2].

The dataset which we used contains number of incidents that the London Fire Brigade attended. For each incident, a variety of data is provided, including the incident's location, the time of the incident, Property Type, Pump Count and so on.

• *Data Cleaning*

We organized and summarized the data by selecting a sample that was representative of the entire dataset. We determined the ideal size of our sample by looking at the target values. We believed that a sample that is very small is not representative of the entire dataset, since some of the higher notional costs will be missed and our models will fail to predict unseen high values. On the other hand, a sample that with massive size is more than necessary. Then we began looking at the relationship between elements.

Our dataset had a number of features which seemed to be unnecessary and didn't have an impact on the target variables such as time and locations of the incidents. Prior to analyzing any data or creating models, we cleaned the data to achieve the highest possible performance. After performing data cleaning, our dataset had the columns reported in Table (1). Initially, our dataset contained 493000 records. Hence, we decided to select 20005 (17905 above £1000, 2100 under £1000) records. This made our dataset suitable to work with since it then covered large and small notional costs and ignores the excessive redundant values. As it was mentioned in the Table (1), our new dataset contains both categorical and numeric data

[1] London Fire Brigade

[2] https://data.london.gov.uk/dataset/london-fire-brigade-incident-records

that are used to predict the notional cost, therefore we converted the Incident Group labels into numeric values.

Column Name	Data type	Role
HourOfCall	numeric	Input Feature
IncidentGroup	categorical	Input Feature
NumStationsWithPumpsAttending	numeric	Input Feature
NumPumpsAttending	numeric	Input Feature
PumpCount	numeric	Input Feature
PumpHoursRoundUp	numeric	Input Feature
Notional Cost (£)	numeric	Target Value

Table(1) – Features of final dataset

- ***Data Normalizing***

The data records for some features contained values in a wide range. To balance this difference, we had to transform them to take a value between [0,1]. There were three options for this transform. Transform (2) and (3) would result in larger values for the evaluation metrices e.g. mean squared error since the difference would remain relatively large. Hence we used transform (1) for normalizing the data.

$$x_! = \frac{x_! - \mu_!}{\sigma_!} \qquad (1)$$

$$x_! = \frac{x_! - min}{max - min} \qquad (2)$$

$$x_! = \frac{x_!}{max} \qquad (3)$$

3. *Methods*

Since we were predicting the notional cost of the service, we used the regressors to capture the trend and predict the target values. The regression models that we used are as follows:

- ***Linear Regression and SVM:***

We tried training a linear regression model with the three normalization techniques on the dataset. We have also trained an SVM model. By analyzing the correlation between the different features of the dataset, it is found that the notional cost is positively correlated with Pump Hours Round and Pump Count.

- ***Ridge and Lasso:***

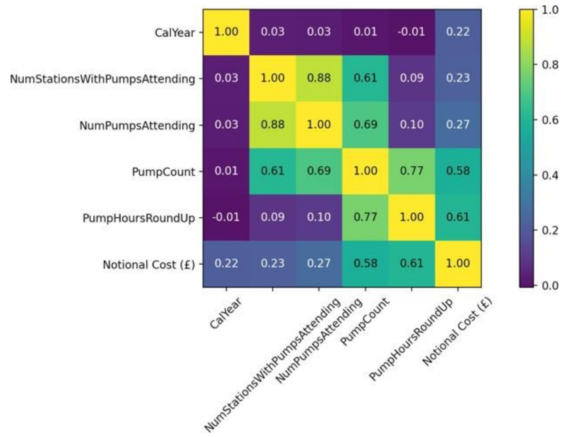
In this project we also constructed and compared Lasso regression model and Ridge regression model. Since these two models deal with multi collinearity we wanted to understand the influence or change in the target variable when one of the independent variables or features changes. We looked into the performance of these two models.

- ***Kernelized K-Nearest Neighbors:***

Although the data has a linear behavior, to familiarize ourselves to different models and make sure nothing is missed out, we used kernelized Knn model.

4. *Experiments and Results*

5. Before these two models were applied, we calculated the coefficients to get a better understanding of the relationship between a Notional Cost and other independent variables. There are several approaches to detecting multicollinearity. One method is use NumPy corrcoef() function. Beside using this function, we used the trial and error method to come up with the most important features. The heatmap (1) clearly demonstrates the high relationship between Pump Hours Round Up and Notional Cost as well as Pump Count and Notional Cost. This indicates that an increase in these independent variables (Pump Hours Round Up and Pump Count) will result in an increase in the Notional Cost. Since one of the main goals of our research in a regression model was to determine how each predictor affects the target variable, therefore these investigations and the experiments were required.



Figure(1) – Correlation between the features

- ***Evaluation***

Mean squared error, mean absolute error, and R squared error are the major metrics we use while building regressors. The average squared difference between estimated and actual values is measured by the Mean Squared Error (MSE). The Mean Absolute Error (MAE) is a measure of errors between paired observations reflecting the same occurrence, in this instance the comparison of expected and target values. MSE and MAE are both optimized at zero, therefore lower values indicate higher prediction accuracy. When comparing the MAE to the MSE, it's worth noting that the MAE gives smaller mistakes less weight. The R2 score is the difference between forecasting a constant output and reducing square error. When the model predicts flawlessly, R 2 = 1; when the performance — no better than the mean value, R 2 = 0.

$$\frac{1}{m}\sum_{i=1}^{m}(y-\hat{y})^2 \qquad \text{(Mean Squared Error)}$$

$$\frac{1}{m}\sum_{i=1}^{m}|y-\hat{y}| \qquad \text{(Mean Absolute Error)}$$

$$1-\frac{\sum_{i=1}^{m}(\theta^T x^{(i)}-y^{(i)})^2}{\sum_{i=1}^{m}(\theta^T x^{(i)}-\hat{y})^2} \qquad \text{(R2 Score)}$$

- ***Linear Regression***

We used the sklearn's linear regression to train the model with the dataset. The data was divided randomly into 20% test and 80% training data.

$$y=\theta_{!}+\theta_{''}x_{''}+\theta_{\#}x_{\#}+\theta_{\$}x_{\$}+\theta_{\%}x_{\%}+\theta_{\&}x_{\&}+\theta_{\text{'}}x_{\text{'}}$$

Coefficients reported:

Coefficient	Value
$\theta_{!}$	-0.00004.5701
$\theta_{''}$	0.0000131111
$\theta_{\#}$	0.0000746312
$\theta_{\$}$	0.0001608901
$\theta_{\%}$	-0.000758666
$\theta_{\&}$	0.02.2767194
$\theta_{\text{'}}$	0.989151211

Table(2) – Coefficients of the linear regression

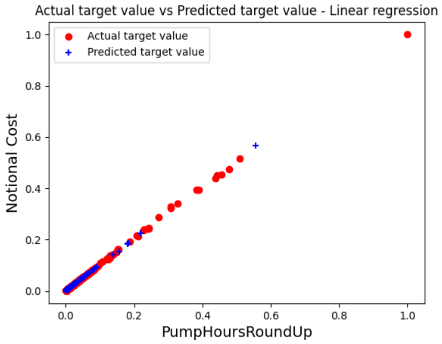
It can be seen that as Figure() suggested, the last input feature which is Pump Hours Round Up has the most influence on the prediction since the model has set its corresponding coefficient around 1. This is the reason why the data pattern will behave linearly.

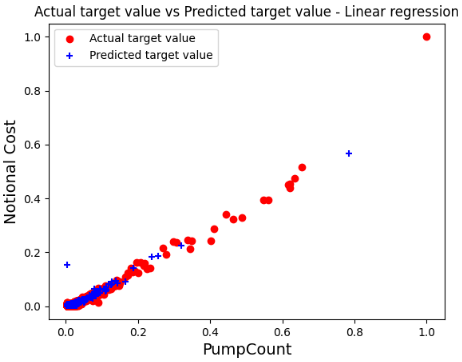
The regressors performance metrics are reported below:

R2 score	MSE	MAE
0.999453815	0.000235913	0.00007978

Table(3) – Evaluation Metrics of the linear regression

The variations of the notional cost as a function of Pump Count and Pump Hours Round Up are plotted as shown below:

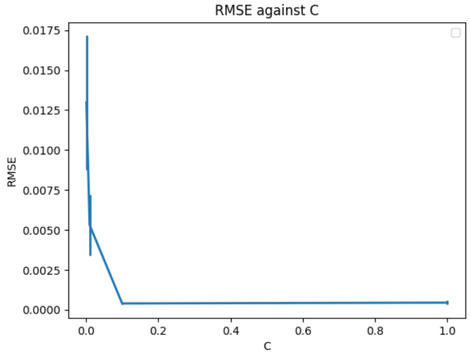




Figure(2) – Prediction vs training set – linear regression

- ***SVM***

For training the SVM model we use LinearSVR from sklearn and the value of the regularization parameter C is evaluated over a range of values: 0.001, 0.01, 0.1 and 1. The optimal value of C is chosen from K-fold cross validation and plotting the root mean square error of the models with each value of C. The C vs RMSE graph is as shown below:



Figure(3) – RMSE vs C – SVR

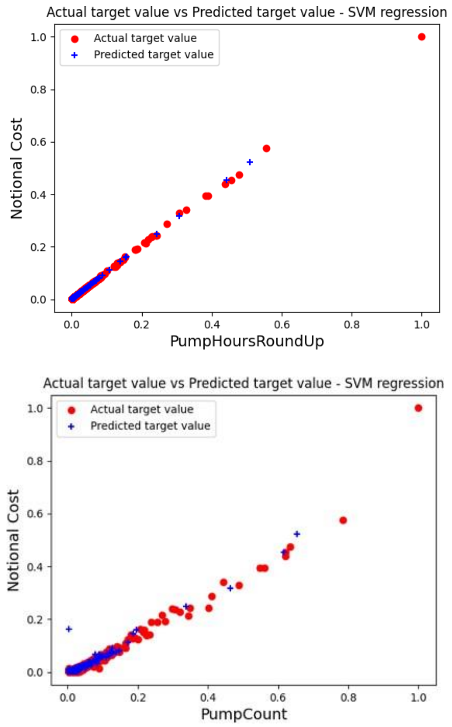
From the graph above, we can see that the RMSE hits the lowest value at C=0.1 and retains the same value for all higher values of C. Table (4) are the performance metrics of the model.

As stated earlier, the notional cost is positively correlated with the features Pump Hours Round Up and Pump Count. We have plotted the predicted and actual values of the notional cost against these variables and the results are as shown in Figure (4).

Both graphs confirm the positive correlation between the output y with the input features

Alpha	R2 score	MSE	MAE
0.001	0.179875	0.012973	0.002066
0.01	0.864707	0.005307	0.000914
1	0.998746	0.000408	0.000111
10	0.998683	0.000455	0.000138

Table(4) – Evaluation Metrics of the svm model



Figure(4) – Prediction vs training set – SVR

- ***Ridge and Lasso***

Next we evaluated the performance of Lasso and Ridge. To achieve this goal, first we chose a range of C values from 0.001 to 100 and for each C, we created a model using Lasso and Ridge regression. Next we used the K-fold split datasets into 5 folds and then we ran multiple iterations.

According to the table below, the Ridge regression model significantly outperforms the Lasso regression model. This can be imputed by the fact that unlike Ridge that uses L2 regularization, Lasso uses L1 which in this case, it fails to capture data behavior.

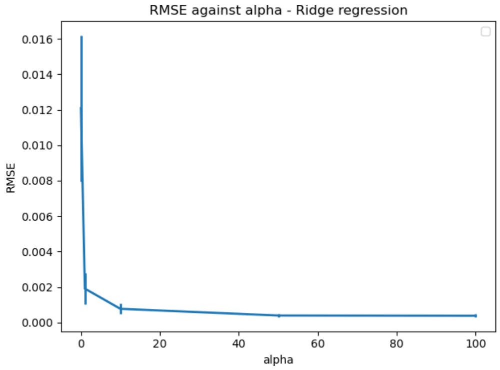
$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{1}{C}\sum_{j=1}^{n}(\theta_j)^2 \qquad \text{(L2)}$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{1}{C}\sum_{j=1}^{n}|\theta_j| \qquad \text{(L1)}$$

C	R2_ridge	MSE_Ridge	MAE_Ridge	R2 lasso	MSE lasso	MAE_lasso
0.01	0.297139	0.012085	0.002517	-0.00339	0.014322	0.03373491
1	0.977765	0.001908	0.000721	-0.00339	0.014322	0.0337616
10	0.986035	0.000768	0.000326	-0.00339	0.014322	0.03376186

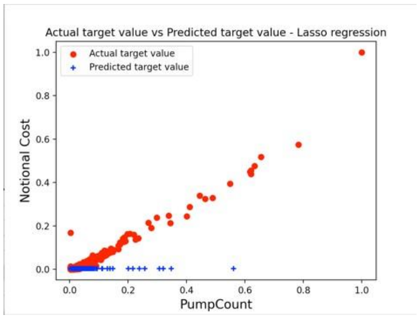
Table(5) – Evaluation Metrics of the lasso and ridge model

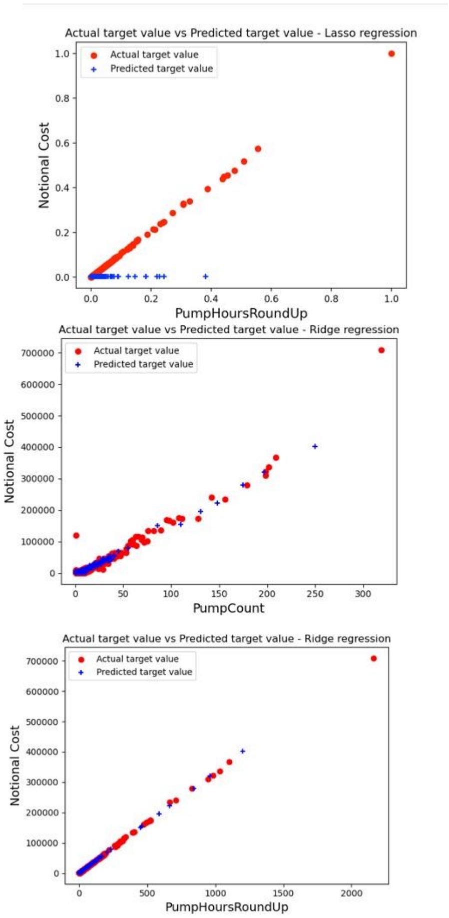
As the C gets larger, we get an almost straight and constant line which means if I choose any alpha value from 10 to 100, there is not a big difference in the mean-square.



Figure(5) – RMSE vs C – Ridge regression

After finding the optimal C value for each model, we split the data set in the ratio of 80% and 20%. Then we trained the models with the optimal alpha value. Finally we plotted the actual target value versus the predicted target value using two independent variables (Pump Hours Round Up and Pump Count).





Figure(6) – Prediction vs training set – Lasso and Ridge

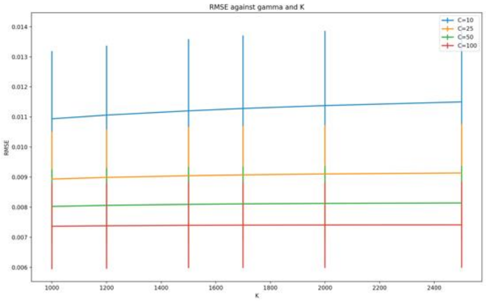
- ***K-Nearest Neighbors***

We used cross validation On all of the models, we utilized cross validation to select the hyperparameters. The k value and the value were the hyperparameters we chose using cross validation.

Since the dataset has relatively large number of records (20005), for k we used the values 1000, 1200, 1500, 1700, 2000 and 2500. And for the gamma if kernel function we used values of 10, 25, 50 and 100. The RMSE almost levels out for values of k >=1000, as shown in the error plot below(left).

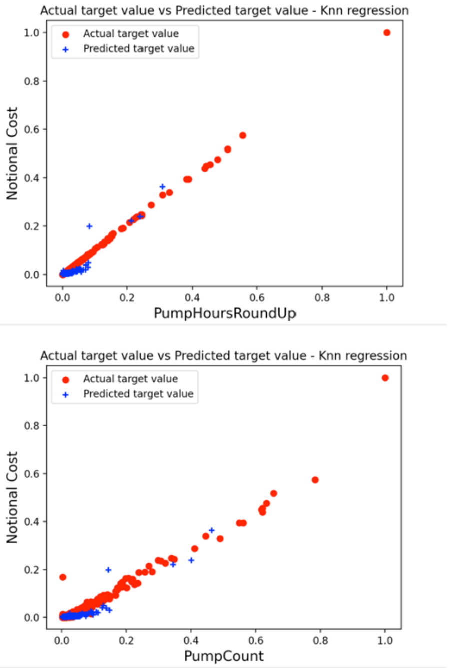
However, choosing a k that is too large may result in overfitting, so we will use k = 1000 in our final model. Table () reports the metrics for the kernelized Knn model.

It can be seen in Figure () that the model has tried to capture the data behavior. However, since the data pattern is linear, linear regressor models would be a better choice to predict the notional cost.



K	R2 score	MSE	MAE
1000	0.675559	0.007364	0.002005
1200	0.673848	0.007381	0.002022
1500	0.672424	0.007395	0.002031
1700	0.671793	0.007402	0.002038
2000	0.671244	0.007407	0.002042
2500	0.670788	0.007412	0.002047

Table(6) – Evaluation Metrics of the Knn model



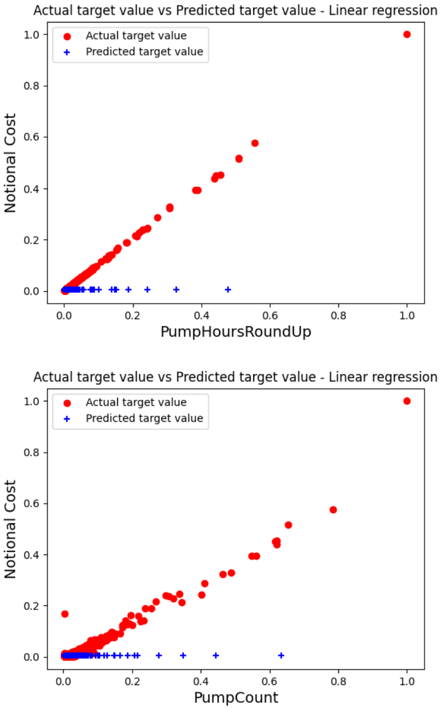
Figure(7) – Prediction vs training set – Knn

• *Baseline comparison*

We wanted to understand how a baseline classifier would perform if it were to predict the mean value of the notional cost. We used DummyRegressor from sklearn to understand baseline performance. The metrics are as follows:

R2 score	MSE	MAE
-0.000026182	0.01924471	0.00312287

Table(7) – Evaluation Metrics of the Baseline model



Figure(8) – Prediction vs training set – Baseline

5. *Conclusion*

We initially suspected that, since in the space we created with the final input features, the data pattern is linear, the linear regressors would be the most accurate predictors. We used other regression model e.g. Knn to examine and compare the performances. Table (7) summarizes our results. As expected, the linear models outperform the other regression methods. Therefore, the top three recommended methods are linear regression, svm and ridge regression respectively. It can be concluded that in such service, the amount of hours in which the pump will be on directly determines the price. Although other features would impact the price, but Pump Hour Round Up will outshadow the their effect.

Regressor	R2 score	MSE	MAE
Baseline	-0.00002618	0.01924471	0.00312287
Linear	0.989453815	0.000235913	0.00007978
SVM	0.97932822	0.00018668	0.0000830
Lasso	-0.0003274	0.0146075	0.00308205
Ridge	0.94846813	0.004589195	0.00081356
Knn	0.8479669	0.005241063	0.00178085

Table(7) – Performance Comparison between the models

6. ***Summary***

The project was started by searching for a dataset of which a meaningful research could be conducted. We selected London Fire Brigade Incident records to see what features might influence the price of such service and how the notional price could be predicted.

The raw data had several columns, some of which were gradually removed from the final dataset as we proceeded and tried training models with multiple features and received undesirable results.

Since we aimed to predict the price, we used the common regressors for this purpose.

We noticed that as we initially expected, the linear regressors e.g. linear regression, svm and ridge outperform other regressors as data pattern is linear in the space we created with the final input features.

We concluded that the only elements that would determine the price of the service, is the amount of hours in which the pumps will remain on.