**Predicting Rent of House in Dublin**

Ayush Kalra, Student ID 20327429

Jing Wang, Student ID 21330849

Kaaviya Karunanidhi, Student ID 21331515

**1. Introduction**

Machine Learning plays a crucial role in our lives recently in all the major fields like health, transport, housing, banking, agriculture, medicine, etc. Houses are the essential needs in every human’s life and the price of houses have a positive impact on a country’s economy. It is evident that the prices of houses in Dublin increase every year by around 10% which is high and the prediction of this increase in house prices will help property owners and other stakeholders to decide the characteristics of house which is more correlated with the price which in turn helps to determine the accurate market value of the houses in different areas of Dublin.

In our Machine Learning group project, we are trying to predict the rent of houses in Dublin which could benefit people who are searching to rent or buy houses especially real estate agents and house vendors in Dublin according to their individual requirements. Data for the rent are taken from **Daft.ie** website using **web crawling** method and this website has all the required data for diverse types of houses with number of bathrooms, bedrooms, etc across all the areas in Dublin. We used different methods to predict the house prices in Dublin and evaluated the model accuracy for all the different models used. The models used in our project is Linear Regression, Lasso Regression, Support Vector Regression and XGBoost. For this project, we found the best fit parameters for every model using k fold cross validation and analysed the accuracy for different regression models we used. The accuracy of all the models is compared with the baseline model which is linear regression in our case. In addition, scatter plots are used to show the original and the predicted data.

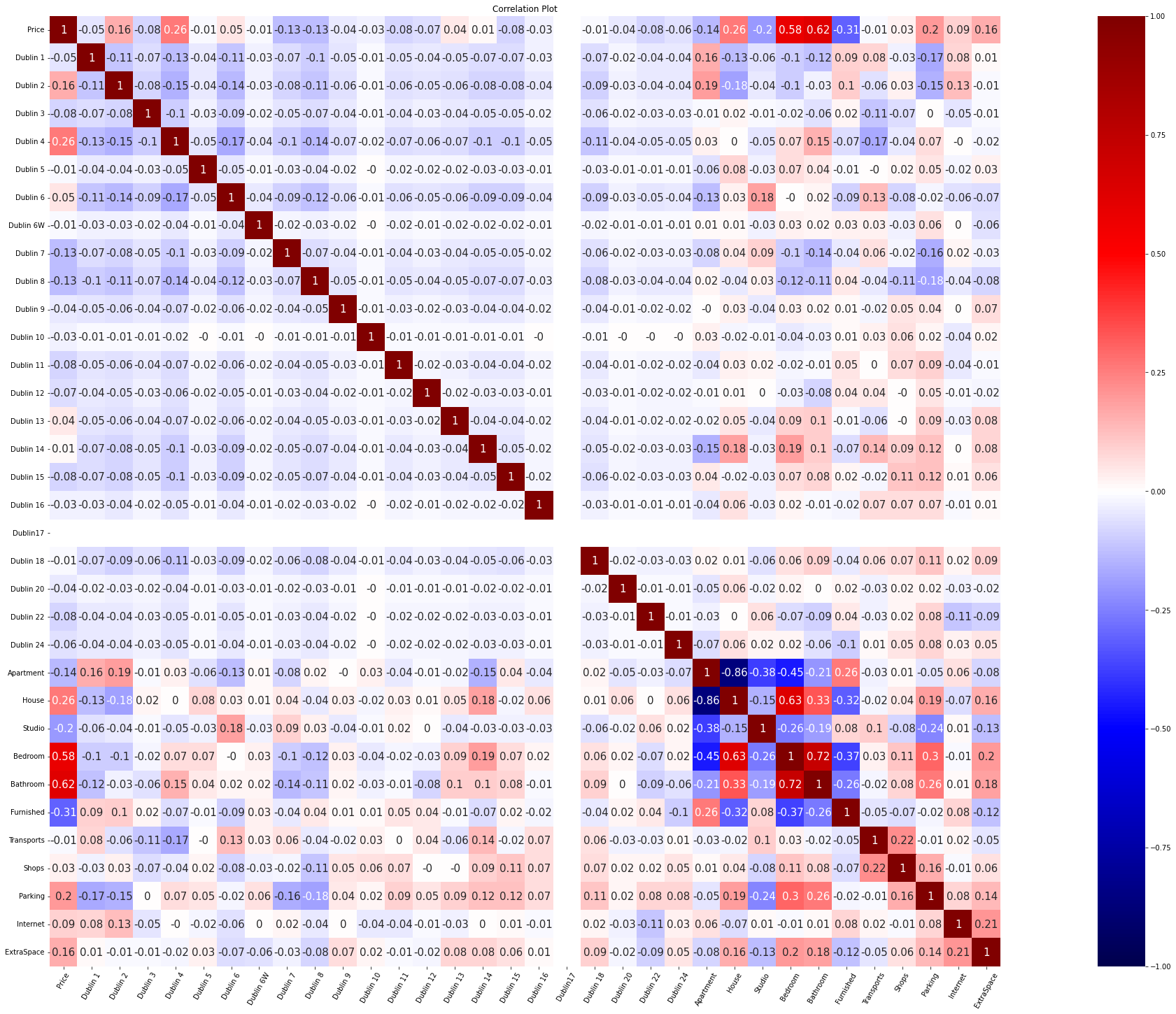
**2. Dataset and Features**

We made a web crawler programme to collect data from [**Daft.ie**](https://www.daft.ie/). The tools we used to do this were BeautifulSoup4 and urllib.request. Every property has 35 features including the rent price per month which is the target to be predicted. The details of the 34 are shown below:

1. Rent price
2. Dublin’s postal districts (23 districts in total, this will be saved as one hot data, so 23 features)
3. Property type (Apartment or House or Studio. This will be saved as one hot data, so 3 features)
4. Number of bedrooms (1 room: 1, 2 rooms: 2, 3 rooms: 3 ……)
5. Number of bathrooms (1 room: 1, 2 rooms: 2, 3 rooms: 3 ……)
6. Furnished or not (Not furnished: 0, Furnished: 1)
7. Whether is near to public transport (No: 0, Yes: 1)
8. Whether is near to shops (No: 0, Yes: 1)
9. Whether has parking lot (No: 0, Yes: 1)
10. Whether has the Internet (No: 0, Yes: 1)
11. Whether has Garden / Patio / Balcony (No: 0, Yes: 1)

Every data will be saved as a vector. For example, in a data with (1900, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1), the first element 1900 is rent price, the second to 24th elements (0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) is the one hot data for district which is Dublin 9 in this data, the following three elements (1, 0, 0) is the one hot data for Property type which is Apartment in this data, the 28th element 1 means the property has 1 bedroom, the 29th element 1 means the property has 1 bathroom, the 30th element 1 means the property is furnished, the 31st element 1 means the property is near to public transport, the 32nd element 0 means the property is not near to shops, the 33rd element 1 means the property has a parking lot, the 34th element 1 means the property has the Internet, the 35th element 1 means the property has garden or patio or balcony.

Finally, we scraped 747 data in total. Since most features are either 0 or 1 (notation for false/true), so we did not apply normalisation on our data. The correlation of different features is shown in Figure 1. According to Figure 1, price is highly related to the number of bedroom and bathroom. Regarding to the district, properties in Dublin 2 and Dublin 4 usually cost more, because these two areas have higher positive correlation with price. Among the three types of property, house usually cost more than apartment and studio. The feature of whether furnished or not has a negative correlation with price which surprised us. Regarding whether near to public transport or shops are almost no correlation to price, while properties have parking lot, Internet or extra space (garden, patio, balcony) usually cost more.

`

**Figure 1. Correlation between features of the Housing Data**

**3. Methods**

The processed and cleaned data is used for modelling to predict the house prices using the below 4 methods.

**3.1 Linear Regression Model:**

Supervised machine learning method which is used to model a linear relationship between dependent and independent variables and the sum of all squared residuals is minimized in this model where the residuals is defined as the difference between the predicted and target values.

The mathematical formula for the linear regression model is given as below:

Y= AX + ε

Here, Y denotes dependent variable, X denotes independent variable, A denotes coefficient of the linear regression, ε denotes random error.

**3.2 Lasso Regression Model:**

Lasso (Least Absolute Shrinkage and Selection Operator) regression is a type of linear regression where the data values are shrunk towards the central point (like mean) which performs L1 regularization which adds a penalty equal to the absolute value of the magnitude of the coefficients. This form of regularization can lead to sparse models with few coefficients; some coefficients may become zero, and the model may be removed. Larger penalties provide coefficient values that are closer to zero, which is great for making simpler models.

**3.3 SVR:**

SVR stands for Support Vector Regression. It is a supervised learning algorithm that is used to predict discrete values. It uses the same principle as SVMs (Support Vector Machines). The basic idea behind this model is to find the best fit line and, in this model, it is a hyperplane containing the maximum number of points. Unlike the other models which usually try to minimize the error between real and predicted values, the SVR tries to fit the best hyperplane within a threshold value. The threshold value is the distance between the boundary line and the hyperplane. The various hyperparameters for SVR are:

1. Hyperplane
2. Kernel
3. Boundary Lines

**3.4 XGBoost:**

XGBoost is also known as Xtreme Gradient Boosting is a supervised machine learning algorithm that is used for both classification and regression problems. It belongs to a family of boosting algorithms and uses Gradient Boosting at its core. Boosting is nothing but a sequential technique that works on the principle of ensemble. Ensemble learning involves training and combining individual models to get a single prediction. In this method, new models are created that compute the error in the previous model and the residuals are added to make the final prediction.

**4. Experiments**

The mentioned regression models are trained and the performance is evaluated on the test data and the accuracy of prediction is calculated for all the different models and compared with each other in order to find the best and good fit model.

In experiments, the housing data is split into training and test data using train\_test\_split function from scikit learn library. 20% data points are reserved as test set:

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 10)

The performances of models are evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). MAE is the mean of absolute error between true value and predicted value. It can be expressed as:

MAE =

MSE is the mean of squared error between true value and predicted value. It can be expressed as:

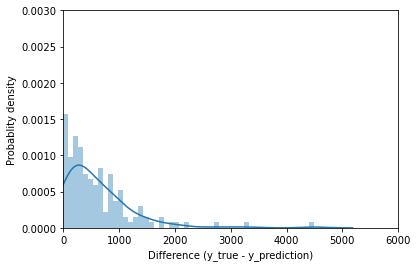
MSE =

RMSE is squared root of the mean of squared error between true value and predicted value. It can be expressed as:

RMSE =

Here, denotes true value, while denotes predicted value.

**4.1 Linear Regression Model**



Multiple linear regression is used for experiments, which has multiple feature variables and only one target variable. There is no hyper-parameter need to be tuned for linear regression model.

Figure 2 shows the distribution plot for linear regression model. The absolute differences between the actual and predicted house prices are plotted in the right graph of Figure 2. The x-axis represents the absolute difference, while the y-axis represents the predicted value of house price. From the distribution plot, most of the data is equal to about 200, while several absolute differences are larger than 4000.

**Figure *2.* Distribution plot of difference between true and predicted values (Linear regression)**

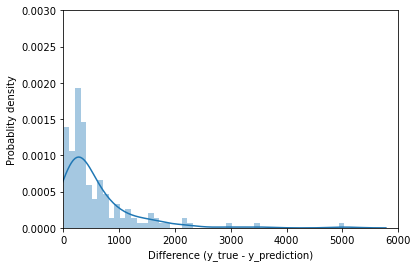
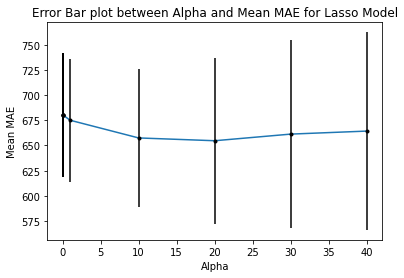
**4.2 Lasso Model**

We implemented Lasso model to predict the house prices in Dublin and used 8-Fold cross validation method for different value ranges of alpha to find the best alpha.

**Alpha Range = [0.0001, 0.001, 0.01, 0.1, 1, 10, 20, 30, 40]**

From the error bar plot, it is evident that the error is less when the value of alpha is equal to 20 which means that the Lasso model performs well in predicting the test data when compared to the other values of alpha as the mean error is less in this case. The error bar plot in Figure 4 shows the different values of alpha and its mean MAE.

From the error plot, the mean MAE value is less when the value of alpha is equal to 20 and it is high for all the other values of alpha which gives the best parameter value (alpha) for the Lasso Model. Now the Lasso model is compared with the other regression models by evaluating the model on the test dataset with the chosen value of alpha equal to 20 by using the below code.

   
Figure 4 shows some results plot for lasso regression model. In the scatter graph, the x-axis represents the true value of house price, while the y-axis represents the predicted value of house price. It is clearly seen from the scatter plot that the linear relationship between the actual and predicted values is not so good as the true value gets larger, the predicted value is usually lower than the true value.

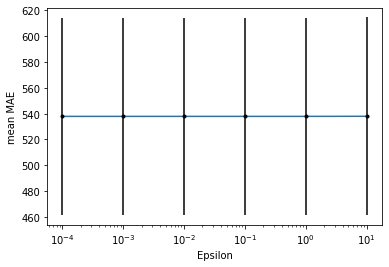
**Figure *3.* Error plot between alpha and mean MAE for Lasso Model**

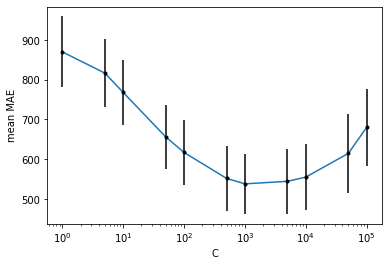
**Figure *4.* Distribution plot of difference between true and predicted values (Lasso regression)**

The absolute differences between the actual and predicted house prices are shown using the distribution plot in Figure-4. The x-axis represents the absolute difference, while the y-axis represents the predicted value of house price. From the distribution plot, most of the data is equal to about 200, while several absolute differences are larger than 4000.

**4.3 SVR**

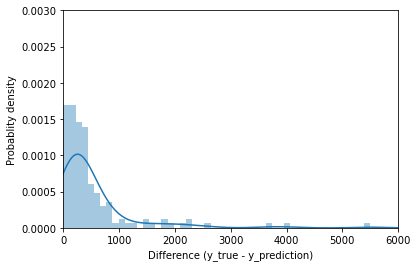
There are only two hyperparameters for the SVR (Support Vector Regression) and for this model we have selected both the parameters to change. The first parameter **“C”** which is also known as regularization parameter and the other parameter known as **“Epsilon ε”**. **K-fold** is set to **8**, for the same reasons given above.

**Regularization Parameter C Range [1, 5, 10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000]**



**Figure 5. Error plot between c and mean MAE for SVR Model**

**Figure *6.* Error plot between epsilon and mean MAE for SVR Model**

****We have taken wide range of C [1-105] to get an idea around the point where the value of Error is minimum. The hyper-parameter C is a regularization parameter which not only reduces the norms of weights but also helps in getting low error on training data as well. The value of C =1000 is chosen. This value is chosen based on the Figure 6. As it is clearly seen on increasing the value the error gets reduced to 500 for value of C= 1000 and on further increasing the value the error is increased again.

**Epsilon ε Range [0.0001, 0.001, 0.01, 0.1, 1, 10]**

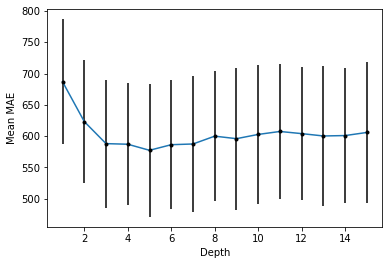
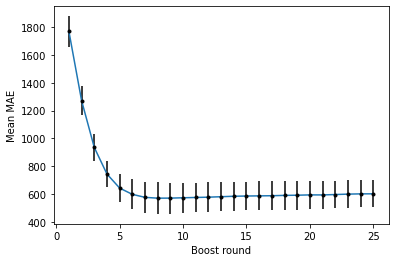
**Figure *7.* Distribution plot of difference between true and predicted values (SVR Model)**

The hyperparameter ε is used to fit the training data.

As from the Figure 7 it is clearly seen that the error value is same. Therefore, we have decided that to set the value of **ε=1** because error was same at all points and it is simpler

**4.4 XGBoost**

There are different parameters and hyperparameters for the XGBoost but for this model we have selected to change two of the hyperparameters namely **“max\_depth”** and other is the **“num\_boost\_round”** and rest are chosen with them by default values**. K-fold** is set to **8,** for the same reasons given above.



**Figure 8. Error bar plot between depth and mean MAE for XGBoost Model**

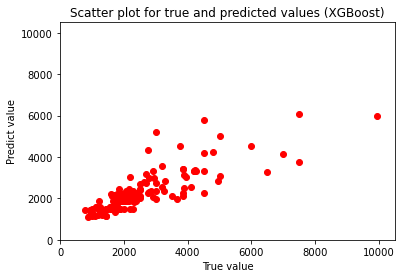
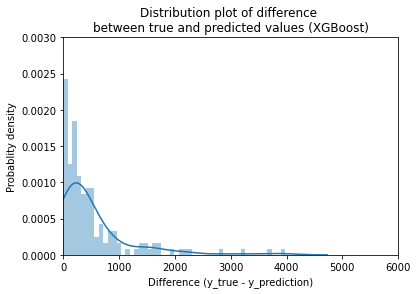
**Figure 9. Error bar plot between boost round and mean MAE for XGBoost Model**

**Max\_depth** Range [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15] The default value for this is 6 and so the values around them are taken into consideration for the optimal selection of the value. Max\_depth is the Maximum depth of the tree. Increasing this value will make the trained model more and more complex and is likely to overfit. **Max\_depth = 5** is chosen. Figure 8 shows the error bar plot between depth and mean MAE for XGBoost Model. Although on selecting the value of max\_depth of 5 the error is not zero, but the value of MAE error is lower than the other values and on further increasing the value of the variable to 12,13 the error has increased but it is somewhat constant and since the error is lowest at 5. Hence the value of hyperparameter is selected to be 5.

The second hyperparameter we changed is the num\_boost\_round.

**Num\_Boost\_Round Range [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25]**

Although there is no default value for this parameter, we have taken in account comparatively low values to analyze the score of the Mean MAE. It corresponds to the number of boosting rounds. It is passed as a standalone argument to the model. The value of this parameter depends on the other parameters as well, so it is tuned every time a parameter is set. Figure 9 shows the error bar plot between boost round and mean MAE for XGBoost Model. **Num\_Boost\_Round = 8** is chosen. On selecting the value for this variable, the Mean error is not reduced to zero, but it is minimum than the other values that are passed to the function. And, from Figure 9 it is clearly visible that although on increasing the value the error is lower, but it is somewhat constant.

**Figure 10. Plots for XGBoost Model**

**4.5 Results and Discussion**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **MAE** | **MSE** | **RMSE** |
| **Linear Regression (Base line)** | 609.8255 | 769146.4430 | 877.0099 |
| **Lasso Regression** | 577.2366 | 790857.5720 | 889.3017 |
| **SVR** | **514.8628** | 815178.3896 | 902.8723 |
| **XGBoost** | 525.2525 | **747381.3959** | **864.5122** |

**Table 1. Results of prediction**

The results are shown in in Table 1. Linear regression model is used as the baseline for comparison. As for MAE, lasso regression model, SVR model and XGBoost outperform linear regression model. In addition, it is evident that SVR regression model got a best MAE of 514.8628, which means it performs better than all the other models. On the other hand, MSE and RMSE are lower for the XGBoost regression model (864.5122) when compared with all the other models. Hence, the good fit models in our case are XGBoost model and SVR Model. However, the MAE of SVR is lower than linear regression, whereas the RMSE of SVR is higher than linear regression. It implies the prediction of SVR may has a larger variance.

Even we drew a conclusion that both SVR and XGBoost perform better than other models, the prediction values still have large error to the true values. As a result, the models we tested may not be efficient for this task. As future works, we can introduce different models and carry out evaluation using larger datasets.

**5. Summary**

In this project, house rent in various parts of the Dublin is predicted using Machine Learning Regression models. Multiple features were considered for training the model and found that the lasso and linear regression model does not perform well as it has large errors. The XGBoost and SVR models performs better than that of the linear and lasso models but still the accuracy is expected to be more when we use our model in real world applications. The performance of the models can be increased by training a larger dataset and with more features so that the good fit model with less error can be found.

In future, deep learning methods can be used to train a large dataset with multiple features so that the model performs well on the test data and the prediction accuracy will be increased with low values of errors and this could be used in real world applications.

**6. Contributions**

**Ayush Kalra-**Worked in coding and report part for SVR and XGBoost models. Worked in Methods part of the report and helped summarizing the format of the report.

**Jing Wang-**Worked in web crawling to extract and clean data from the daft website and reviewed for the necessary changes in the report and the coding part whenever required. Worked in Dataset and Features section of the report.

**Kaaviya Karunanidhi-**Worked in coding and report part for Linear and Lasso Regression models. Worked in Introduction, Results and Summary part of the report.

**7. Code**

We have uploaded the programmes on Github. Here is the link: <https://github.com/xxxiaojing/Daft-rent-price-prediction>.