Analysis of Ireland Property Prices

2022

By

Ginu varghese
Under the supervision of
Dr. Jack Mc Donnell

School of Informatics and Creative Arts
Department of Computing Science and Mathematics

ACKNOWLEDGEMENTS

All the data used for this project was collected from the PSRA website. The data collected from this website is strictly used for research purposes without manipulating the data.

I would especially like to thank Dr. Jack Mc Donnell, the Program Director of Data Analytics program in Dundalk Institute of Technology. As my lecturer and supervisor, he has taught and helped me more than I could ever give him credit for here. He gave me suggestions and advice in each stage of my thesis and helped me to do the thesis more efficiently.

I am grateful to Dr. Fiona Lawless, the head of the Department of Computing Science and Mathematics and all the lectures from the Data Analytics program, Dr. Siobhan Connolly Kernan, Dr. Rajesh Jaiswal, Dr. Peadar Grant and Dr. Abhishek Kaushik, who provided me extensive professional knowledge and taught me a great deal about data science skills during lectures, laboratories, and projects.

This work is self-funded, and I would like to thank Dundalk Institute of Technology (DkIT) for facilitating the MSc in Data Analytics program that has helped me to work on this project.

Additionally, I am grateful to all the researchers who has done research in this area which helped to refer their works in times of doubts and the online materials which was available to learn all the technologies used in the thesis and to enhance my data science skills.

I would like to thank my parents, Varghese A.D and Lissy Varghese, whose love and guidance are with me in whatever I pursue. They are the ultimate role models.

I wish to thank my friends and siblings, who provide unending inspiration, support and love.

DECLARATION

I hereby declare that the work described in this project is, except where otherwise stated, entirely my own work and has not been submitted as part of any degree at this or any other Institute/University.

Signed: 

Name: Ginu Varghese

Date: 30/08/2022

ABSTRACT

The house prices are increasing every year since 2012 in Ireland, and it is demanding the analysis and prediction of property prices. The main objective of this study is to conduct both statistical and machine learning analysis on the property prices of Ireland from 2010 to 2022 collected from the PSRA website. Factors influencing the house prices county, year, province, location of property, property size and type are considered for analysis. Different machine learning algorithms like Linear Regression, Support Vector Machine regression, Decision Tree, XG Boost, Neural Network, and Random Forest regression were used for analysis to identify the better fit model. Statistical tests and visualizations were generated for the analysing the effect of different attributes on property prices and the results implied that the province, year, property size, and location are statistically significant and are affecting the house prices significantly. The machine learning algorithms XG-Boost, Neural Network and Support Vector Machine was found to be the better algorithms for predicting the property prices based on the R squared, Root Mean Squared Error and Mean Absolute Error values.

The code designed for this project is available on GitHub in the following link: https://github.com/ginuvarghese16/Dissertation.

ABBREVIATIONS

Abbreviation	Meaning	Page
SVM	Support Vector Machine	**1,2,3,8,10,11, 13,14,22,27,53.**
LR	Linear Regression	**2,3,8,13,14,39,42,43,44,46,47,48,50,51,52,55,56,63**
RF	Random Forest	**1,2,3,5,6,7,8,9,10,11,13,14,15,17,20,23,24,27,28,31,39,56,57,58,60**
DT	Decision Tree	**1,3,8,13,14,17,23**
XG-Boost	eXtreme Gradient Boosting	**3,13,14,24,56,57,58,60,63**
ANOVA	Analysis of Variance	**3,26,39,40,41,42,63**
COVID-19	Coronavirus disease	**2**
GBM	Gradient Boosting Machine	**3,7,9,14**
L1	Lasso Regression	**3,14**
L2	Ridge Regression	**3,14**
VS	Visual Studio	**5,39,43,47,51,59,60**
GAM	Generalized additive models	**6,26**
GWR	Geographically Weighted Regression	**6**
ML	Machine learning	**6,9,13,25,27,31,39,42,43,44,46,50,51,52,55,56,57,58,63**
K-NN	K Nearest Neighbor	**8,13**
MSE	Mean Squared Error	**7,8,11,13,15,27,28,29,57,58,63**
MAE	Mean Absolute Error	**7,8,11,13,15,27,28,57,58**
RMSE	Root Mean Squared Error	**7,8,11,13,15,27,57,58,63**

RMSLE	Root Mean Square Logarithmic Error	**8**
ANN	Artificial Neural Network	**8,11,12,13,22,23,25,62**
GA	Genetic Algorithm	**8,10**
HGA-ANN	Hybrid genetic algorithm–artificial neural network	**8**
BP	Back Propagation	**8,10,11**
LSSVM	Least Squares Support Vector Machine	**8**
PLS	Partial Least Squares	**8**
NN	Neural network	**3,8,9,11,12,13,22,23,25,45,56,58,60,62**
PCA	Principal Component Analysis	**9,11**
GM	Grey Algorithm	**10**
PSO	Particle Swarm Optimization	**10**
BPNN	Back Propagation Neural Network	**11**
RBF	Radial Basis Function	**11**
CRISP-DM	Cross-Industry Standard Process for Data Mining	**15,16**
PSRA	Property Services Regulatory Authority	**4,17,63**
CSO	Central Statistics Office	**17**
API	Application Programming Interface	**17,61**
ISM	Interpretative structural model	**14,62**

GDPR	General Data Protection and Regulations	**61**
HTTPS	Hypertext Transfer Protocol Secure	**61**
VAT	Value Added Tax	**4,19**
MLP	Multilayer Perceptron	**25**
KDE	K Desktop Environment	**22**

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

An accurate house price prediction is important for prospective homeowners, real estate developers, investors, banks, governments, tax assessors, insurers, and mortgage lenders (Frew and Jud 2020; Ihre 2019). House price in Ireland has been highly volatile due to several factors such as availability, material cost, population density, and rising rents (Reddan 2018). Several prospective homebuyers are struggling to secure enough money for their first home (Coughlan 2022). Having an accurate prediction regarding the house price will help these buyers to plan for their first home. Different machine learning models such as Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT) can predict house prices considering several factors that are not considered while using traditional methods for house price predictions.

While traditional methods use factors such as type, size, quality of finish, location, number of floors, area of the property, availability of facilities such as schools, hospitals, and grocery stores; modern machine learning algorithms can factor in several more such as inflation, salary, environmental factors, and geography (Hurley et al. 2022). Furthermore, machine learning will help to identify house price determinants that are selectively applicable for the stakeholder who will be using these predictions. However, individual contributors that influence the house price will be different for each state and country. So, it is always a benefit for the stakeholders to know the changes that are predicted to happen soon to the property market to act accordingly.

The house prices in Ireland have been increasing every year since 2012 and the prospective homeowners are finding it difficult to find a home within their budget. Ireland faced a construction boom with wage growth, bank credit and rapid increase in property prices in the early 2000s (Jose Doval Tedin et al. 2020). However, from 2007 to 2013, during the crisis time the house prices decreased sharply by almost 50%. As the economy recovered, the prices increased from 2013 onwards and by 2020 the rents have reached 32% higher than the previous years (Jose Doval Tedin et al. 2020). Even though the house prices have hit a new record, beyond the records during the Celtic tiger years (FitzGerald 2007), several investors are considering properties as a long-term investment, which is further driving the house prices. Most of the model that predicts

housing prices are not defining the investment aspect of owning a house as one of the variables that can affect the housing prices. Traditional methods have totally ignored the trends that are observed in major cities like Dublin, Manchester, Barcelona where vulture investors are investing in properties instead of companies, which is one of the highly influential factors in driving the house prices (Petrov 2009). The COVID-19 pandemic has shown a few factors that can drive the house prices by 50% (Sullivan 2021). Disruptions in supply chain and unavailability of the work force has also been a factor that has to be considered while using different methods for predicting house prices. The COVID-19 pandemic has provided the opportunity to work from home that caused less expenses in daily commute, coffee, and other expenditures that in turn added to the savings for buying houses. When people started working from home, proper workspace became a necessity. This has led to a trend of people owning houses farther from cities, but with better facilities. Different countries have reported this to be a factor to drive house prices in rural areas of the countries (Charlie Weston 2021). Also at least a few people have considered the opportunity to work from home as a factor to consider buying houses that are far from cities but have more facilities (Sullivan 2021). These factors have affected an indirect increase in housing prices, especially in Ireland. During the COVID-19 pandemic people were unable to spend money on holidays, dining out, outdoor activities, entertainments like movies and concerts, and instore shopping which accumulated into the savings and thereby increasing the buying power. Thus, those savings were converted to long term investments and a major portion of that investment was in real estate which drove the property prices even further. Furthermore, a country like Ireland which highly depends on external sources for building materials that are being brought to Ireland by freight will be affected and house prices will be highly influenced by any changes that will be seen in supply chain (Gazette Desk 2021). Along with these, conflicts between countries can drive the material cost and fuel cost that will affect the house prices and these events are highly unpredictable (MacFarlane 2022). In this scenario, it is important to know the trends in the property market and to understand the factors influencing the property prices.

Machine learning algorithms can assist us in predicting the housing prices while considering all these factors. These models can be trained using data that has all the mentioned attributes. Generally, scholars have used hedonic model, Linear Regression (LR), Random Forest (RF), and Support Vector Machine (SVM) algorithms for

predicting house prices (Ja'afar et al. 2021). The data used for training machine learning models will be a key factor in determining the accuracy of the model. Historically, Ireland has a large database in relation to price and traditional factors that affect the price of property (CSO 2022). Having a model that can be trained with this data will result in better accurate predictions for the future. Along with this the same model can be expanded to be used in different countries that are having similar structures.

This research is based on both statistical and machine learning methods to analyse and predict the property prices of Ireland from 2010- 2022. For statistical analysis I used hedonic regression, linear regression, and ANOVA methods. Machine learning methods such as SVM, DT, XG-Boost, RF, NN, and Linear regression (LR) algorithms are employed to predict the property prices. Hedonic model is one of the most popular methods used for house price prediction and it considers the house as a combination of many attributes (Limsombunc et al. 2004). The main goal of hedonic model is to estimate the contribution of different attributes to the price of property (Montero and Fernández-Avilés 2014). SVM has been considered by many researchers since it works well with high dimensional data, unstructured and semi-structured data. Also, the outliers have less influence on the SVM, and larger amount of data can be modelled using SVM (Advantages of Support Vector Machines (SVM) n.d.). In addition to this, SVM has been found extremely popular in commercial field for predicting the sales of the company (Ho et al. 2020). When it comes to RF, it is been used for prediction in many applications due its ability to reduce the over fitting (Ho et al. 2020). RF also provides higher accuracy compared to other models and it can also deal with the larger datasets (Mbaabu 2020). Compared to other algorithms, DT requires less time for data preparation while pre-processing and data normalization and scaling is not required for it. Both classification and regression problems can be solved using the DT (K 2019). In XG-Boost algorithm, it has the in-built ability to deal with the missing values and is faster than the gradient boosting machine (GBM). It is also referred as the regularized form of GBM because it has in-built Lasso regression (L1) and Ridge regression (L2) which reduces the overfitting (Kumar 2019). Linear regression is considered as one of the simplest algorithms and the over fitting problem in modelling can be avoided by using the regularization and cross validation techniques. It also works well with systems that has less computational power and has a noticeably lower time complexity (Waseem 2022). The research will be based on the property sales data of Ireland from 2010 – 2022

and it is collected from the Residential Property Price Register page of the PSRA website. The research scope and goals of the research are listed in the Table 1 and the core technologies used in the research are mentioned in Table 1.1.

Research questions	Project Goals
1. What is the trend in the property prices over the years?	• Find out the property prices in each year. • Find out the counties in which most and least houses are sold. • Find out the house prices in each county/province. • Find out the year in which most houses were sold. • To find out the trend in the property pricing over the years. • Find out either the New or secondhand dwelling has the highest prices and is it inclusive of VAT. • To analyze how the house prices depends on the type of house and its size. • To estimate the different factors affecting the property prices. • To evaluate the effect of postcodes in determining the property prices. • To determine the impact of latitude and longitude on the property prices.
2. Which County has the highest demand for properties?	
3. Does the property price depend on which county the house is in? Is there any relationship between the counties and the prices?	
4. Is there any relationship between the size of the properties, type of properties (like new or second hand) and the property prices?	
5. What are the factors contributing to the price of the properties?	
6. Is the post codes are impacting the house prices?	
7.How does the latitude and longitude effect the property prices?	

Table 1. 1 Research questions

Technologies Used
Programming language: Python (Jupyter Notebook, VS, Spyder)
Python libraries: • Pandas • Numpy • Matplotlib • Seaborn • scipy. stats • statsmodels • Plotly • Geopy • Keras
For visualization: Seaborn, Matplotlib, Plotly
For version control system: GitHub.

Table 1. 2 Technology used for the research

This paper is structured as follows; chapter 2 gives the related works, the life cycle of the project is described in chapter 3, data exploration is presented in chapter 4, and methods used in the research are mentioned in chapter 5. Chapter 6 describes the different evaluation methods used in the project and the chapter 7 focuses on the results and discussion which demonstrates the outcome of the initial and final analyses performed. The ethical considerations related to the research are mentioned in chapter 8, then conclusion and future work is given in chapter 9 and finally references are mentioned at the end of the report.

CHAPTER 2

LITERATURE REVIEW

Several studies have been performed in the past to analyse and predict the real estate and residential property prices to aid the customers as well as the developers. Hurley & Sweeney (2022) studied the impact of post codes and address in Irish property prices based on the data from January 2018 to November 2018. They focused on analysing the property prices in Dublin with the dataset of 5028 properties for the development of geospatial statistical models where the post codes and addresses are being mislabelled. They also performed text mining to create additional variables that describes the features of the properties and its surroundings. A spatial hedonic regression model was used to separate the spatial and non-spatial contributions of property features to the resale value. Generalized additive models (GAM), regression kriging and Geographically Weighted Regression (GWR) have also been used since these methods provide greater interpretability with smaller data requirements. Three different Machine learning (ML) methods like Decision Tree, Random Forest and K-nearest neighbor algorithms were also applied on the data to evaluate how these ML models perform with smaller datasets.

Their models shown a reduction in median absolute percentage error with an increasing model complexity i.e., 12% for the hedonic model and 9.6% for linear model with spatial surface. And the authors also stated that although ML models are widely used for property prices prediction, they did not have the probability-based uncertainty intervals and the interpretability of the statistical spatial models. They also added that the random forest models may not fit well with the areas outside Dublin since the housing turnover is low outside Dublin and in that case statistical spatial modelling can work more efficiently. In Ireland, where property valuations are currently based on comparison to recently sold neighboring properties, the authors claim their model has higher applicability and will help to improve property tax computations and site value estimates because the model is not only based on the property value but uses the spatial location scaling (Hurley et al. 2022).

Machine learning models were used to predict house prices in Godavari district of Andhra Pradesh, India by Thamarai and Malarvizhi (2020). The models were built to help the people buy suitable houses for their needs. Decision tree regression, decision

tree classification and linear regression models were performed on the data based on the attributes of the property like number of bedrooms, age of the house, availability of school near the house and shopping malls available nearby the house location. Attribute selection algorithms were used to remove the redundant features to reduce the impurity in the process before splitting the data for modelling.

To predict the availability of houses according to the requirement of the user, they used the decision tree classifier which gives responses like yes or no to show whether a house is available or not. Along with this, regression methods like decision tree regression and multiple linear regression were used to predict the prices of the houses. The dataset used for the modelling was a real-time data acquired from the Godavari district with all the attributes of the house and modelled using the scikit learn, a machine learning tool.

The main dataset was divided into train and test data and the decision tree classifier is performed using the training dataset. The accuracy of the model is then checked using the test data. Mean Squared Error (MSE), Mean Absolute Error (MAE) and root mean squared error (RMSE) were used to evaluate the performance of both the classification and regression models. The house price prediction with decision tree algorithm produced an output with some data record prices predicted with lesser deviations. From the multiple linear regression, it is found that the number of bedrooms is the feature having high influence on house price and age of the house is the feature having less influence on house price. From the performance metrics it is found that the prediction of house price using multiple linear regression has higher performance than the prediction using the decision tree regression. The authors also stated that the developed model can be used to predict the availability and prices of houses for any new attributes according to the users and the overall accuracy can be increased in the future with a large dataset and by identifying the best features.

Quang et al. (2020) conducted a study to compare housing prices prediction using traditional and advance machine learning methods. They used three different machine learning models: Random Forest, XGBoost and LightGBM. Further, two machine learning techniques are used, Hybrid Regression and Stacked Generalization Regression for prediction. For the analysis, the Beijing house price data from 2009 and 2018 was used and feature engineering was performed to select the appropriate features. In addition to this, exploratory data analysis was done to discover the patterns in the data.

The machine learning models were applied on the training data which is split from the actual data and Root Mean Square Logarithmic Error (RMSLE) was used for evaluating the performance of the models. Results from the study shows that the Random Forest method is prone to overfitting even though it has lowest errors. The Hybrid regression method is better performing among all three methods. They also stated that although the Stacked regression method has the worst time complexity, it is the best choice when accuracy is considered(Quang et al. 2020).

To understand the factors influencing the property prices, Decision Tree (DT), Random Forest (RF) and linear regression (LR) methods were used by Yee et al. (2021) using the Malaysian dataset from April 2017 to December 2019, to help the buyers and sellers who need to finance in the property market. The accuracy of the models is evaluated based on the R squared, RMSE and MAE values. Natural Language Processing was used to transform the data so that it is readable by the machine. The outcome of the study explained that the RF is the better model with higher accuracy compared to the DT and LR (Yee et al. 2021).

Alen Ihre (2019) discussed the machine learning algorithms K-Nearest Neighbour (K-NN) and Random Forest (RF) regression to predict the house prices using the Ames dataset of 3000 observations. Five-fold cross validation was performed on the dataset to minimize the bias and grid search algorithm was utilized to select the best number of hyperparameters for the prediction. Finally, the RF was found to be the best performing model based on the MAE values. However, there is small differences in the actual price and the predicted price, and the author suggested that the results could be improved by using a larger and less biased dataset (Ihre 2019).

Artificial Neural Network (ANN) is used for analysing the real estate price by Shi and Li (2009) to evaluate the house price determinants. An improved Genetic Algorithm (GA) was used to optimize the weights of the neural network. The results of the study shown that the GA-ANN is more capable of determining the house price determinants more time efficiently and the errors of the HGA-ANN model was found to be lower than the back propagation (BP) and the genetic algorithm (GA) models.

Support Vector Machine (SVM) has been employed for predicting and forecasting house prices by Mu et al. (2014), Phan (2018), Ho et al. (2020), Chen et al. (2017), Gu et al. (2011) and Wang et al. (2014).

House value forecasting-based machine learning methods by Mu et al. (2014) aimed at helping the developers and government to take decisions regarding developing real estate in the Boston area. The authors collected the data from the UCI data sets and support vector machine (SVM), least squares support vector machine (LSSVM), and partial least squares (PLS) algorithms were applied on the training data to predict the housing value. From the prediction results it is found that the SVM and LSSVM has better efficiency with the nonlinear data. PLS algorithm is better for linear data due to the simplicity of the algorithm. They also added that to achieve best forecasting effect and an optimal solution, SVM can be used (Mu et al. 2014).

Phan (2018) utilized the SVM technique to predict the house prices in the Melbourne city of Australia to help the house buyers and sellers. Neural network (NN), Polynomial regression, Linear regression and Regression Tree models were also developed along with SVM to identify the best fit. The data used in the study was downloaded from the Kaggle website and it has the house sold houses transaction from 2016 to 2018. Data imputation and descriptive analysis techniques were performed on the data prior to modelling. Along with this, principal component analysis (PCA) was also performed to select the desired features and stepwise method was utilized for subset selection. The results shown that the SVM with the subset selection method gives the best efficiency with lower errors. When regression tree and linear regression delivered almost equal prediction result, the polynomial regression gave better accuracy with lower errors. The neural network seemed to be not working well with the available dataset. The authors also stated that regression tree and neural network worked more faster than the SVM, where PCA with SVM took more time than SVM with stepwise (Phan 2018).

SVM methods were employed by Ho et al. (2020) for predicting the property prices in Hong Kong. Random forest (RF) and Gradient Boosting Machine (GBM) were also utilized along with SVM to compare the performance of algorithms. The dataset used was a sample data with over 40000 housing transactions in a time of 18 years. Correlation matrix was used to determine the features which should be included in the models. The results from the performance metrices revealed that the RF and GBM were able to estimate the house prices better than the SVM with smaller errors. They also found that the ML algorithms need more computation time than the traditional Hedonic pricing model and among the three models used, SVM is the better choice for forecasting

when speed is the priority and RF and GBM should be used if the accuracy is considered (Ho et al. 2020).

To predict the housing prices in the Taipei city of Hong Kong, SVM models were implemented by Chen et al. (2017). By using stepwise multi regression, the support vectors were found, and a SVM hedonic price model was built using the support vectors, the structural and the spatial variables to predict the house prices in the Taipei city. The SVM model is then developed based on the identified support vectors to forecast the future housing prices for the data from 2007 to 2010. One of the advantages in using SVM is that it is not depending on the probability distribution assumption and hence it could plot the input variables into a high dimensional feature space. To compensate the bias variance trade-off, five-fold cross validation has been used for testing and training in the analysis. The outcome from the study points out that the SVM can be considered as a superior approach which legitimise the issues in the multiple regression analysis and combining the hedonic approach with SVM is feasible for non-linear modelling (Chen et al. 2017).

Gu et al. (2011) used the SVM methods along with hybrid genetic algorithm methods to forecast the house prices in China. SVM has proven to be one of the best algorithms in both classification and regression in lots of applications. In the study, genetic algorithm (GA) has been used instead of grid algorithm to optimize the parameters of the SVM since, GA is more time efficient and the G-SVM is developed. The results of the study revealed that the G-SVM method giving more accuracy than the Grey Algorithm (GM) which is used in the past to predict the house prices (Gu et al. 2011).

Machine learning algorithms has used to forecast the real estate prices by many researchers and Wang et al. (2014) used SVM to forecast the real estate price with particle swarm optimization (PSO) in the Chongqing city in China. One of the reasons to choose SVM is its ability to conquer the 'Curse of dimensionality'. To identify the parameters of SVM, PSO method is used instead of GA and grid algorithm since it is easy to enforce. The actual data was divided into train and test data for the modelling. The study shown that the PSO-SVM model is performing better than the BP neural network methods used by other researchers.

PSO has also used by Alfiyatin et al. (2017) for predicting the house prices to help the builders to decide the selling price of the house and to help the buyers to set the right

time to buy the house. PSO and regression analysis was implemented on the houses data in the Malang city of Indonesia within 2014-2017. Hedonic regression was chosen as the regression prediction model and PSO to select the appropriate features. The error prediction values are found to be higher for the regression model compared to the PSO-regression model. Hence the study proved that the combination of PSO with regression can give the minimum prediction error.

Support Vector Machine Regression (SVR) has used by several researchers to predict the real estate and housing prices due to its efficiency and application. A SVR model was used by Li et al. (2009) to understand the possibility of predicting real estate prices in China from 1998-2008. The results of the model are compared with the Back Propagation Neural Network (BPNN) model to analyse the performance of the model. Several indicators like CPI, loan interest rate, real estate price, real estate investment, income etc. were used to forecast the real estate prices. The analysis demonstrated that the SVM as regression model works better than the BPNN model for real estate forecasting based on the MAE, MAPE and RMSE. The study also proved that the SVR method is an efficient approach to forecast the real estate price (Li et al. 2009).

SVM regression model with Gaussian kernel was utilized by Miao et al. (2021) to predict the property prices in Boston area. The aim of the study was to help the people to estimate the price of the house based on the properties of the house. They selected the most important features using the decision tree with ID3 algorithm, divided the data to train and test data and SVM regression is employed on the data with the gaussian kernel to predict the prices and compared the model with different regression methods to analyse the performance. The study proved that the SVM regression with gaussian kernel has more efficiency than the KNN, decision tree and SVM regression with linear kernel. But the model still has some drawbacks which are mentioned by the authors and the important one is that some of the factors that impact the house prices cannot be measured such as the cultural tolerance of the neighbours and so on.

Jiao Yang Wu (2017) also performed a study to analyse the house price prediction using SVR based on the housing sales data of Kings County, USA with an aim to help the buyers and sellers. Feature selection methods like Random Forest selector, Lasso ridge and Recursive Feature Extraction and the feature extraction method PCA is done prior to building the model. Further to this, parameter tuning, and transformation techniques have been used to improve the accuracy of the model. But the results from

the experiments shown that there is not much difference in the performance of the model with feature extraction and feature selection. The Radial Basis Function (RBF) kernel with SVR was found to be the best one among the performed combinations. The author also pointed out that in future other machine learning models like XGBoost and other feature engineering methods can be applied on the data (Wu 2017).

Hedonic Regression is one of the estimation and prediction method preferred by the researchers when it comes to prices. It is used in most of the scenarios when a price variable is to be considered. Property prices in Croatia is studied by Kunovac and Zagreb (2019) using Hedonic regression based on the data collected from different sources. One of the goals of the research was to propose how the hedonic models can be used for the evaluation of residential property. The hedonic model built allows the evaluation of some attributes of the property such as age, location and so on. The results of the analysis shown that the micro location of the property should also be considered to the hedonic models and to commonly used other models to improve the prediction of residential house prices (Kunovac and Zagreb 2019).

Abdulai and Owusu-Ansah (2011) used hedonic regression to determine the house price determinants in Liverpool, The United Kingdom over a period of 18 years from 1990-2008. They also analysed how the past and present buyers valued the property features. The regression model was able to explain almost 75% of variation in the housing prices and all the variables in the dataset was found to be statistically significant. Another thing they found was that the price of the new properties is almost double than the old properties and the detached houses are more expensive than the flats. When the past buyers before 1999 focused more on the number of bedrooms, bathrooms and detached houses, the buyers after 2000 more value the number of floors, gardens, and showers (Abdulai and Owusu-Ansah 2011).

Selim (2008) identified the house price determinants in Turkey using hedonic regression using the 2004 household survey data. Environmental factors were not included in the data for analysis and natural logarithm of the house price is treated as the dependent variable. To estimate the hedonic model, ordinary least square method is utilized. The heteroscedasticity present in the model was eliminated using the White's heteroscedasticity consistent coefficient covariance matrix. And the results of the analysis shown most of the variables to be significant and the house prices seems to be higher for houses with more rooms. The variables such as water system, pool, type of

the house, number of rooms, size of house, locational attributes and building structure are the found to be the most significant variables that influence the house prices (Selim 2008).

Limsombunc et al. (2004) compared the hedonic regression with Artificial Neural Network (ANN) model on house price prediction based on a sample dataset of 200 New Zealand houses. The age of the house, size of house, bedrooms, bathrooms, and other features were considered for the experiment. The heteroscedasticity consistent coefficient covariance matrix and weighted least squares method were used instead of the ordinary least square method to eliminate the heteroscedasticity issues. However, the heteroscedasticity problem was not completely removed. The results from the study revealed that even though the hedonic model was able to explain almost 70% of variance in the model, it did not outperform the neural network model. The authors mentioned that the small dataset and the lack of environmental features may be some reasons for the poor performance of the hedonic model (Limsombunc et al. 2004).

Following researchers have used supervised and semi-supervised ML models for predicting property and real estate prices (Ihre 2019; Phan 2019; Quang et al. 2020; Mu et al. 2014; Ho et al. 2020; Chen et al. 2017; Gu et al. 2011; Wang et al. 2014; Pow and Janulewicz 1995; Alfiyatin et al. 2017; Li et al. 2009) and have employed the RMSE, MAE, MSE, MAPE, and R Squared performance metrices for evaluating the models. Either regression or classification supervised learning techniques are utilized by most scholars for prediction. There are certain algorithms which are preferred by several scholars such as the RF, DT, LR, K-NN, XG-Boost, SVM and ANN algorithm (Ja'afar et al. 2021).

Factors affecting the house prices have been identified by several researchers and Rahman et al. (2012) explored the factors affecting the house prices in the Chinese city, Hangzhou. The paper used the Ordinary Least Square (OLS) method by taking time series data of several variables from 1990-2009. It is noticed that household's income, lagged sum of annual real estate development investment, the urbanization rate, and the foreign direct investment in Hangzhou positively and considerably affect Hangzhou's housing price. The saving deposits of urban residents in Hangzhou have significant negative effect on the housing price. These results were in line with the theoretical predictions. They also mentioned that they could not include some important variables in the model due to lack of data such as the interest rate, expected return on housing

investment, housing tax and capital gain tax, land release data, etc. Hence, they suggested that future research will be helpful by including these variables and with a long-time frame, to obtain better research results (Rahman et al. 2012).

Antonina Mavrodiy (2005) analysed the determinants of real estate prices in the Kiev city of Ukraine using regression methods and confirmed that different micro and macro factors are affecting the property prices. From the analysis the found out that the GDP, income level, population rate is having a direct impact on the property prices. On the other hand, the interest rate is having a negative impact on the real estate market. He also found out that different geographical factors such as the location of the property, different facilities surrounding the property such as schools, metro station, hospital are having higher effect on the property prices along with the property size and flooring details (Antonina Mavrodiy 2005).

Different factors responsible for the housing prices in China are evaluated by (Gao et al. 2018) by using the theory of interpretative structural model (ISM). They selected 50 real estate residential price factors to build housing price factors binary relation on set and quantitative analysis matrix based on the document analysis of the residential real estate prices. ISM model for residential real estate prices were build using standardized methods and the results showed that, property management companies, brand value, population structure, education welfare state, political stability, disposable income growth, economic growth, land policy, real estate tax policy, living facilities, convenient living and other factors, building properties, location are the fundamental factors affecting the real estate prices and they have significant effect on determining the prices (Gao et al. 2018).

From the previous studies the most preferred algorithm is the RF. It is one of the ensemble methods used in machine learning and it is generally utilized for predicting the real estate prices due to its less error compared to other algorithms (Shinde and Gawande 2018). Another most chosen algorithm is SVM, and it is applied in both regression and classification problems. Since SVM can better deal with bias and can use high dimensional data, it is used as recognition in classification problems (Ja'afar et al. 2021). DT, XG-Boost and LR are also preferred by the researchers because, compared to other algorithms, DT requires less time for data preparation while pre-processing and data normalization and scaling is not required for it. Both classification and regression problems can be solved using the DT (K 2019). In XG-Boost algorithm, it has the in-

built ability to deal with the missing values and is faster than the gradient boosting machine (GBM). It is also referred as the regularized form of GBM because it has in-built Lasso regression (L1) and Ridge regression (L2) which reduces the overfitting (Kumar 2019). Linear regression is considered as one of the simplest algorithms and the over fitting problem in modelling can be avoided by using the regularization and cross validation techniques. It also works well with systems that has less computational power and has a noticeably lower time complexity (Waseem 2022). From the previous studies (Gao et al. 2018; Antonina Mavrodiy 2005; Rahman et al. 2012) it is evident that the location, the property size, and its other facilities are having a profound impact on the property prices.

CHAPTER 3

LIFE CYCLE OF PROJECT

The project is based on the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology to indicate Machine Learning aspects and ensure a successful execution of the project. The whole life cycle of the project consists of six stages: data collection, data preparation, data exploration, data modelling, evaluation, and deployment.

- Data Collection: The data was collected from the Residential Property Price Register website, which is available to the public for download. The data has the property selling details of Ireland from 2010 - 2022 with the address, post codes, county, prices, and several other attributes.
- Data Preparation: Data preparation involves the cleaning of the data including removing the data recorded in the Irish language, dealing with the missing values, and adding extra data such as province and latitude/longitude details to the available data.
- Data Exploration: Exploratory data analysis and summary statistics were performed to explore the patterns in the data. Data visualizations were performed on the data using different plots to understand the trend of the house pricing.
- Modelling: Statistical and machine learning techniques were employed on the data to predict the house prices. Dependent variable was selected from the data prior to modelling.
- Evaluation: To evaluate the models, performance metrices like MSE, RMSE and MAE were used and along with this, graphs were used to identify the patterns and outliers in the data.
- Deployment: In deployment stage, the review of the project is performed, and future improvements are suggested.

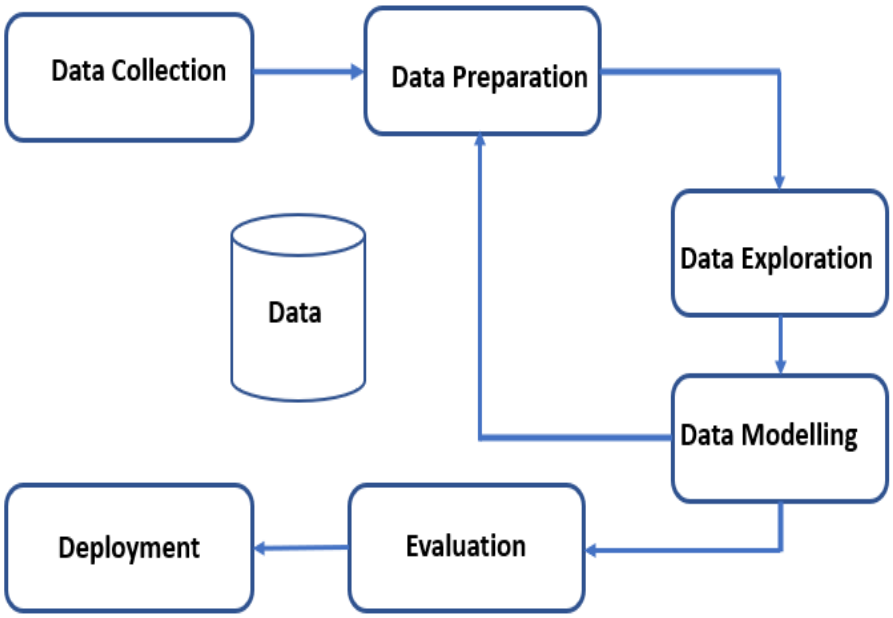


Figure 3. 1 Life Cycle of Project based on CRISP-DM

CHAPTER 4

DATA EXPLORATION

4.1 Data Collection

The data is collected from the Residential Property Price Register page of PSRA website which provides the residential property sales data of Ireland since January 2010. The data is collected as csv files for the ease of the analysis. Prior to this, an attempt was made to collect data from daft.ie, which is a property sales application of Ireland. Unfortunately, it failed since they were not willing to provide their data to a third person. The data from Kaggle and CSO was found to be insufficient due its size and incomplete information. Finally, the data is collected from the https://propertypriceregister.ie. website which has sufficient size and information for the analysis.

4.2 Data Preparation

Data wrangling is performed on the collected data and outliers and missing values are treated for further analysis. To expand the actual data, province, latitude & longitude, and location columns were added. The province details were collected from the 'IrelandTownList' website as a csv file, and it is then merged to the actual dataset. To access the latitude and longitude, an API is needed, and it was created through the https://opencagedata.com/ which is a free and secured website for creating free and paid API keys. The first option for creating API key was Google API but it is paid and not affordable, so I couldn't use it. Python Geopy was also tried for getting the latitude and longitudes however, Jupyter Notebook was not able to process all the data and hence that attempt was also in vain. Finally, API is created through 'opencagedata' website, and the latitude and longitude of each county are accessed and merged to the data. This is because, only 2000 requests can be processed in a single a day using the free API and the dataset has more than 500000 rows which makes it impossible to process the entire data. Another column 'location' is added to the data which contains the details of the place of sale either as 'Dublin' or 'Outside Dublin' for the analysis. Unemployment, income, and homelessness data were supposed to be added to the actual data for further analysis unfortunately, these data were not available for all the years since 2010 and as per county and thus those details were not considered for analysis.

The combined data was then cleaned using Python pandas library and missing values are treated. There are many missing values in the data, but they are kept same since removing them will affect the entire analysis and modelling. The duplicates were checked, datatypes were corrected and Irish text in the data were converted to English using Python. The address values were changed to string title as some of the values were in uppercase. Additional columns were created for date variable with month and year for further analysis and visualization.

4.3 Data Description

The dataset contains the details of the residential property sales of Ireland from 2010 - 2022. It was collected from the Residential Property Prices Register website and has 516586 rows × 9 columns. The data has the date of the sale, the price of the property, county, and other related information about the sale. After removing duplicates and adding the additional columns there were 515792 rows × 15 columns. Table 3 describe the variables in the dataset. The variables month, year, province, location, latitude, and longitude were additionally created for the analysis.

Type of Variables			
Variable Name	**Category**	**Type**	**Description**
date_of_sale	Date	Datetime	The date of the property sale
address	Nominal Categorical	String	The address of the sold property
postal_code	Nominal Categorical	String	The postal code of the sold property
county	Nominal Categorical	String	The county name of the sold property
price	Continuous Numerical	Float	The price of the sold property

FMP	Nominal Categorical	String	The information about whether the sold property price is full market price or not.
VAT_exclusive	Nominal Categorical	String	The information about whether the sold property price is VAT exclusive or not.
property_description	Nominal Categorical	String	The type of property.
property_size_description	Nominal Categorical	String	The size of the property.
province	Nominal Categorical	String	The province name of the sold property
lat	Continuous Numerical	Float	Latitude of the property.
lon	Continuous Numerical	Float	Longitude of the property.
location	Nominal Categorical	String	The information about whether the sold property is in Dublin or outside Dublin.
year	Discrete Numerical	Integer	The year in which property is sold.
month	Discrete Numerical	Integer	The month in which property is sold.

Table 4. 1 Variables in the dataset

4.4 Data Splitting and Scaling

To analyse the property prices in Dublin County, the cleaned dataset was divided into a new one with the details of county Dublin based on the location variable in the final dataset and there were 164027 observations and 15 variables in the new Dublin dataset. The data splitting was done to mainly analyse the effect of Dublin postcodes in the property prices. Data scaling was performed using standard scaler method since it works well for regression problems. Unfortunately, scaled data seemed to give less accurate results than non-scaled data for some of the algorithms and hence scaling was performed only for SVM regression. Apart from this, resampling was performed prior to SVM modelling and, this is because, the entire data was not getting executed for the SVM algorithm.

CHAPTER 5

IMPLEMENTATION METHODS

To analyse and predict the property prices different statistical and machine learning techniques were utilized in the research and this section will give an overview of those methods. The proposed work is implemented using scikit learn, a machine learning and statistical modelling tool in Python and for building NN model, Keras library has been used.

5.1 Scikit Learn

Scikit learn is a useful Python library for machine learning and it contains different tools for machine learning and statistical modelling including classification, regression, and clustering. It is developed upon the Scientific Python (SciPy) which must be installed before using scikit learn (Brownlee 2014).

The stepwise execution of a model using scikit learn is as follows (Andrade 2021).

1. Import the required libraries.
2. Load the required dataset.
3. Split the data into train and test data.
4. Fit the model into the data.
5. Predict the dependent variable for the test data.

5.2 Keras libaray

Keras is a powerful and easy-to-use free open-source Python library which is used for developing and evaluating deep learning models. It is part of the TensorFlow library and it helps to define and train neural network models in just a few lines of code (Javatpoint 2010).

Some of the advantages of Keras are:

- It is very easy to understand and integrate the faster deployment of network models.
- Most of the AI companies are focussed on using Keras and this gives Keras vast community support.
- Keras supports multi backend which means TensorFlow, CNTK or Theano can be used as backend according to the user requirement.

- Keras can be deployed in many devices such as, Raspberry pi, Web browser with .js support and Cloud engine etc.
- It supports data parallelism. So, it can be trained on multiple GPUs at the same time for speeding up the process (Javatpoint 2010).

5.3 Linear Regression

Linear regression is a supervised machine learning algorithm which identifies the best fit linear line between the dependent and independent variable. In other words, it identifies the linear relationship between the dependent and the independent variable. There are two types of linear regression, simple linear regression, where there is only one independent variable and multiple linear regression, where there are multiple independent variables (Deepanshi 2021).

Simple linear regression equation is given by, $y = b_0 + b_1 x$, where b_0 is the intercept, b_1 is slope, x is the independent variable and y is the dependent variable.

Multiple linear regression equation is given by, $y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots . + b_n x_n$, where b_0 is the intercept, b_1,b_2,b_3…,b_n are slopes of the independent variables x_1,x_2,x_3,x_4…,x_n and y is the dependent variable.

The basic assumptions for linear regression are as follows,

1. There should be a linear relationship between the independent and dependent variable. This can be verified using scatter plot between the variables.
2. The independent and dependent variables should be normally distributed. This can be identified using the KDE plots and histograms.
3. The spread of the residuals should be constant for all values of independent variable, which can be verified using the residual plot.
4. There should be no correlation between the independent variables and the error terms should be normally distributed.
5. There should be no autocorrelation.

When the assumptions are violated, it leads to the decrease in accuracy which in turn affects the predictions and makes the high errors (Deepanshi 2021).

5.4 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification and regression problems. The main goal of the SVM is to find a hyperplane in an n-dimensional space where n is the number of features that classifies the data points with maximum distance. SVM is built using the support vectors, which are the data points that are close to the hyperplane and are used to maximize the distance (Gandhi 2018; Ray 2017).

SVM can be of two types, Linear SVM and Non-linear SVM.

***Linear SVM*:** Linear SVM is used for linear data. If a dataset can be divided into two classes using a single straight line, them such data is known as linearly separable data and SVM linear classification is used for such data (SVM Algorithm - Javatpoint 2010).

***Non-linear SVM*:** Non-linear SVM is used for non-linear data. If a data cannot be divided into classes using a straight line, that data is a non-linear data and SVM non-linear classifier is used for such data (SVM Algorithm - Javatpoint 2010.).

Hyperplane in SVM: There are different lines to separate the classes in n-dimensional space and the best line or boundary that separate the classes is the hyperplane of SVM. The dimension of the hyperplane depends on the features in the dataset i.e., if there are two features the hyperplane will be a straight line and if there are three features then the hyperplane will be two-dimension plane (SVM Algorithm - Javatpoint 2010) .

Support Vectors in SVM: The datapoints that are nearest to the hyperplane and that influence the position of the hyperplane are known as support vectors and hence the name support vector machine.

5.5 Decision Tree

Decision Tree (DT) is a supervised machine learning algorithm which can be utilized for both classification and regression problems. It is a tree like structure with root node and then branches into solutions as like in a tree. There are root nodes, decision nodes and leaf nodes in a DT where, root nodes are the beginning node of the DT and the decision nodes are the nodes which we get after splitting the root nodes and the leaf

nodes, where further splitting is not possible. In DT, the overfitting can be reduced by pruning, the cutting down of some nodes (Saini 2021; Amrutha 2022).

Root Node: It is the starting point of the decision trees. It represents the dataset, and it is then divided into two or more homogenous sets. It is also known as the parent node.

Leaf Node: It is the final output node, and the tree cannot be separated further after getting the leaf node.

Splitting: It is the process of dividing the root node into sub-nodes based on the given criteria.

Branch/Sub Tree: It is the tree formed by dividing the tree.

Pruning: It is the process of getting rid of the unnecessary branches from the tree.

Child Node: The nodes other than the root node are called the child nodes (javatpoint 2010).

5.6 Random Forest

It is a supervised learning algorithm used for both classification and regression problems. It has number of decision trees on subsets and takes the average of it to predict the accuracy. The higher the number of trees in the model, the lower the chance of overfitting (Yiu 2019).

Assumptions of RF: The RF joins multiple trees to predict the class of the data. This makes some decision trees to predict the correct output and others to not. But when combined all the trees predict the correct output. Hence, there are some assumptions for the RF to predict the better output.

- There should be some actual values in the feature variables of the dataset for the classifier to predict the correct results.
- There should be low correlations between the predictions from each tree.

RF takes less time than other algorithms and it can predict outputs with higher accuracy for larger datasets and even with large number of missing values (javatpoint 2010).

5.7 XG-Boost

Extreme Gradient Boosting or XG-Boost is an ensemble learning method. It is an extension of gradient boosted decision trees and are designed to improve the performance of the model. XG-Boost has the regularized learning feature, which helps to reduce the overfitting (analyticsvidhya 2018).

The boosting ensemble technique has mainly 3 steps,

1. A primary model F0 is defines for predicting the target dependent y. This will be related to the residual (y-F0)
2. New model h1 is fit to the residuals from the step1.
3. The boosted version (F1) of F0 is obtained by combining F0 and h1. Similarly, new models are created after each residual to improve the performance. The mean squared error of each new model will be lower than previous model. The residuals can be minimized through m iterations (analyticsvidhya 2018).

$$F_1(x) < -F_0(x) + h_1(x)$$
$$F_m(x) < -F_{m-1}(x) + h_m(x)$$

5.8 Hedonic Regression

Hedonic regression is a regression method which analyses the impact of different attributes on the price of a good. In hedonic regression, the dependent variable will be the price of the good and the independent variables will be the attributes of the good that influence the price. It uses the ordinary least squares and other techniques to estimate how the attributes affect the price of real estate like house.

Hedonic regression function is defined as, $p_i = j(c_i)$, where *p* is the price of good *i*, and c_i is the vector of attributes related to the good. The attributes can be location, structure of the property, environmental properties, and accessibility to the property. The hedonic regression is mostly used to estimate the property prices in real estate industry (investopedia 2021; CFI 2022).

The main uses of hedonic models are:

- The hedonic regression method can be used to calculate property and other asset values based on actual choice of the customer.
- It is probably the most efficient method for making use of the available data.

- The use of hedonic regression methods in real estate is effective, as the property sales data is easily available, and other secondary data to get descriptive variables can be accessed or generated readily.
- Hedonic pricing model is versatile, it can be adapted to other market goods and services and environmental qualities (CFITeam 2022).

5.9 Neural Network Regression

Regression ANNs are used to predict an output variable as a function of the inputs. In ANN regression problems the dependent variable should be numeric. Neural networks consist of nodes and each node has corresponding activation function that defines the output of the node with the set of inputs. By changing the last activation function the NN classification problem can be changed to a regression model and the appropriate library to build the NN ML is the 'Keras' library (Ajay Ohri 2022).

To ensure the nonlinearity, the output of neuron is mapped to different values in the NN regression and single or a range of parameters can be selected for predicting output using NN regression. The outputs of neural network related to each other, and these neurons help in predicting future values as well as mapping a relationship between dependent and independent variables. A multilayer perceptron (MLP) is a class of a feedforward artificial neural network (ANN). MLP models are the most basic deep neural network and is composed of a series of fully connected layers (Ajay Ohri 2022).

5.10 ANOVA Test

Analysis of Variance (ANOVA) is a statistical test which helps to find out whether the difference between different groups of data is statistically significant. It also helps us to understand the relationship between the dependent and independent variables (Kenton 2022). One-way ANOVA is the basic form of ANOVA and there are other variations which can be used in different situations.

- Two-way ANOVA
- Factorial ANOVA
- Welch's F-test ANOVA
- Ranked ANOVA
- Games-Howell pairwise tests.

ANOVA works by evaluating the levels of variance within the groups of data through samples taken from each group. If there is lot of difference in the variance of the data groups, then there is chance for the mean of the selected group to be different from the population. ANOVA also deals with the sample size and the difference between the sample means. All these components are linked into a F value, which can then be analysed to give a probability (p-value) of whether the differences between sample groups are statistically significant or not. One-way ANOVA compares the effects of independent variable on dependent variables and two-way ANOVA does the same with multiple independent variables (Qualitrics 2022b).

CHAPTER 6

EVALUATION METHODS

This section explains the different evaluation methods used in the project to analyse the performance of the statistical as well as ML models. The performance metrics used to evaluate the ML models were:

6.1 Root Mean Squared Value

Root Mean Squared Error is the square root of the average of the squared difference between the actual value and the predicted value of the regression model (Bajaj Aayush 2022).

RMSE = $\sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - y_{j*})^2}$

$y_j - actual\ values$

$y_{j*} - predicted\ values$

$N - number\ of\ observations$

Some of the key advantages of RMSE are:

- It preserves the differentiable property of MSE.
- It manages the penalization of smaller errors made by MSE by square rooting the value.
- Error interpretation can be done easily since the scale is same as the random variable.
- It is less prone to outliers (Bajaj Aayush 2022).

6.2 Mean Absolute Value

Mean Absolute Error is the average of the difference between the ground truth and the predicted values (Bajaj Aayush 2022).

MAE = $\frac{1}{N}\sum_{j=1}^{N}|y_j - y_{j*}|$

$y_j - actual\ values$

$y_{j*} - predicted\ values$

$N - number\ of\ observations$

Key information related to MAE are:

- MAE is more robust towards outliers than MSE, since it does not amplify errors.
- It gives an understanding of how the predictions are away from the actual output.
- It does not give us an information about the direction of the error since it uses absolute value of the residual, i.e., whether the model is under-predicting or over-predicting the data.
- MAE is non-differentiable compared to MSE, which is differentiable.
- Error interpretation perfectly line up with the original degree of the variable (Bajaj Aayush 2022).

6.3 Mean Squared Error

Mean squared error is the most popular metric used for regression problems. It calculates the average of the squared difference between the actual value and the predicted value of the regression model (Bajaj Aayush 2022).

MSE = $\frac{1}{N}\sum_{j=1}^{N}(y_j - y_{j*})^2$

Some key information related to MSE are:

- MSE can be optimized better.
- It penalizes even small errors by squaring them, which leads to an overestimation of how bad the model is.
- Error interpretation should be done by considering the squaring factor.
- It is more prone to outliers than other metrics.
- The value of MSE is always positive.
- MSE value close to zero will represent better quality of the regression model (Bajaj Aayush 2022).

6.4 R-Squared Error

R-Squared error works as a post metric, which means it is a metric that is calculated using other metrics. It explains how much variation in actual variable is being explained by the variation regression line. It is calculated using the sum of squared errors (Bajaj Aayush 2022).

Variance of y, SE (y∗) = $\sum_{j=1}^{N}\left(y_j - y_{j*}\right)^2$

Percentage of variation described the regression line: $\frac{SE(line)}{SE(y*)}$

The percentage of variation described the regression line: $1-\frac{SE(line)}{SE(y*)}$

R- Squared error = $1- \frac{SE(line)}{SE(y*)}$

Key points related to R squared error are:

- If the sum of Squared Error of the regression line is small then R^2 will be close to 1 which is the ideal value. This means that the regression line was able to capture 100% of the variance in the target variable.
- On the contrary, if the sum of squared error of the regression line is high then the R^2 will be close to 0. This means that the regression line was not able to capture any variance in the target variable.
- The range of R^2 is $(-\infty,1)$ because the ratio of squared errors of the regression line and mean can exceed the value 1 if the squared error of regression line is too high i.e., if greater than squared error of the mean (Bajaj Aayush 2022).
- R-Squared is also known as the standardized version of MSE(Bajaj Aayush 2022).

The evaluation metrices used for the statistical models were:

6.5 Coefficient Std Error

The standard deviation of an estimate is called the standard error. The standard error of the coefficient determines how accurately the model estimates the coefficient's unknown value. The coefficient standard error is always positive (Vineet Jaiswal 2018).

6.6 Coefficient-t value

t value = estimate/std error

The higher the t value will be helpful in analysis as this would indicate if we could reject the null hypothesis and it is used to determine the p value (Vineet Jaiswal 2018).

6.7 Coefficient Pr(>\|t\|)

Individual p value for each variable to accept or reject the null hypothesis. Lower the p value allow us to reject null hypothesis (Vineet Jaiswal 2018).

6.8 Residual Standard Error

It is the average error of the model, how well the model is predicting the data on average (Vineet Jaiswal 2018).

6.9 Multiple R-squared and Adjusted R-squared

It is always between 0 to 1, higher values indicate better percentage of variation in the dependent variable that is explained by variation in the explanatory variables. This is used to estimate how good the model can explain the variance and when the number of variables is increased, then the values of R-squared will also increase (Vineet Jaiswal 2018).

6.10 F-statistic

F-statistic shows the relationship between predictor and response variables. Higher the value will give more reasons to reject null hypothesis (Vineet Jaiswal 2018).

6.11 p-value

A p-value is a statistical measurement used to justify a hypothesis against observed data. It measures the probability of achieving the observed results, assuming that the null hypothesis is true. The lower the p-value, the greater the statistical significance of the observed difference (Brian Beers 2022). The overall p-value based on the F-statistic. If the value is less than 0.05 for a confidence interval of 95, then the overall model is significant (Vineet Jaiswal 2018). The p-value depends on the confidence interval chosen by the user.

CHAPTER 7

RESULTS AND DISCUSSIONS

Statistical analyses, ML models and visualizations were implemented to explore the patterns in the data and to analyse the attributes contributing to the property prices. This section describes the outcome of the analyses performed.

7.1 Visualizations

Matplotlib, Seaborn and Plotly libraries were used to analyse the underlying patterns of the data. Univariate, bivariate, and multivariate visualizations were generated for better understanding and to explore how the variables are related to each other.

Figures 7.1.1 – 7.1.10 shows the information about the distribution of observations on each variable in the dataset. Count plot was used to analyse the frequency of observations in the variable and log of the price variable is chosen for better visualization and understanding. This is because the price variable has higher values, and this makes the plot difficult to interpret. From the figures it is evident that, there are outliers in some of the variables. Figures 7.1.1 and 7.1.2 reveals that, some of the properties were sold at higher prices and others at very lower prices and most of the sold properties are in Leinster and the smallest number of properties are sold in Ulster. Figures 7.1.3 and 7.1.5 shows that almost half of the observations account for Dublin. From Figures 7.1.4 and 7.1.7, it is evident that the property selling was higher in the year 2019 and least in 2012. The plots also shows that most of the houses are being sold in the month of December. Another observation from the figures is that most sold prices were not full market price and was VAT inclusive. The univariate figures also provide information about the size and type of the sold properties. As shown in Figures 7.1.9 and 7.1.10, most of the sold properties were second hand, rather than new houses. And the highest sold houses were with size greater than or equal to 38 Sqm and less than 125 Sqm, and least sold houses were with size less than 38 Sqm.

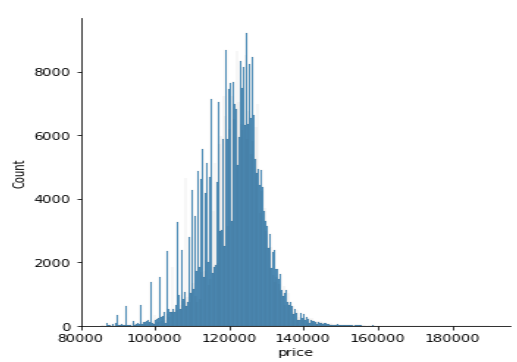


Figure 7.1. 1 Univariate for Price

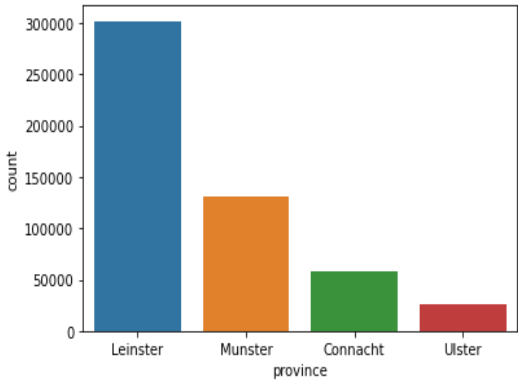


Figure 7.1. 2 Univariate for Province

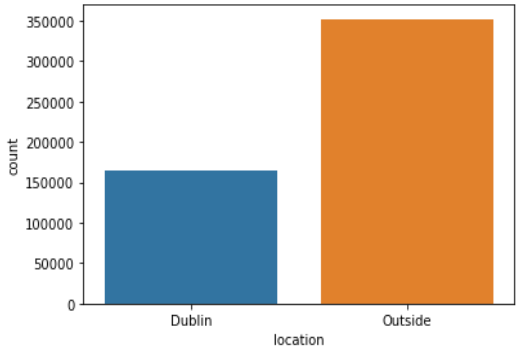


Figure 7.1. 3 Univariate for Location

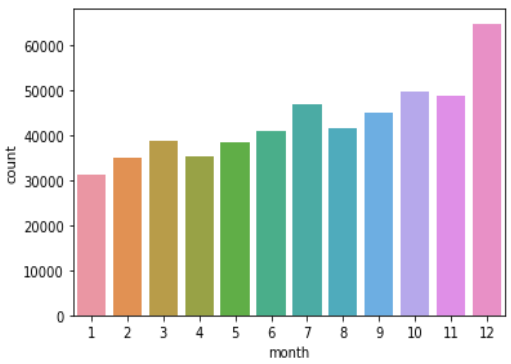


Figure 7.1. 4 Univariate for Month

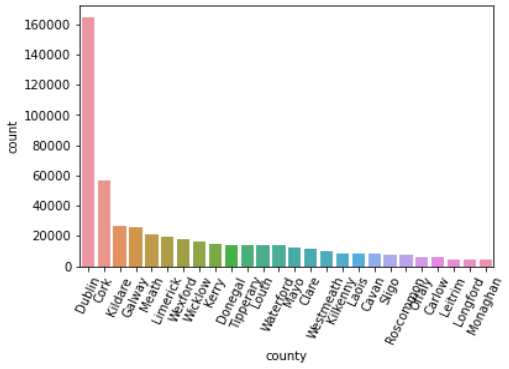


Figure 7.1. 5 Univariate for County

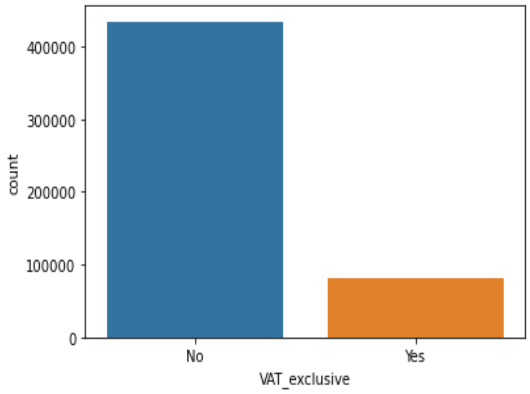


Figure 7.1. 6 Univariate for VAT exclusive

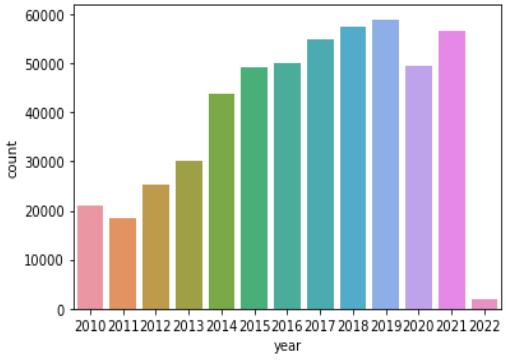


Figure 7.1. 7 Univariate for Year

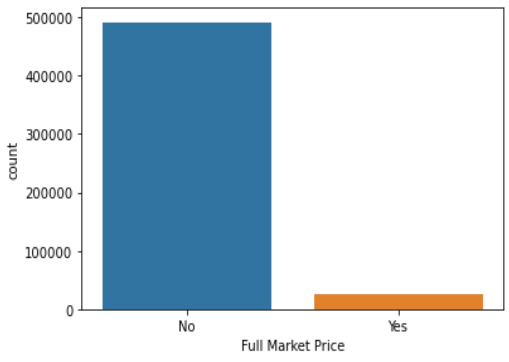


Figure 7.1. 8 Univariate for Full Market Price

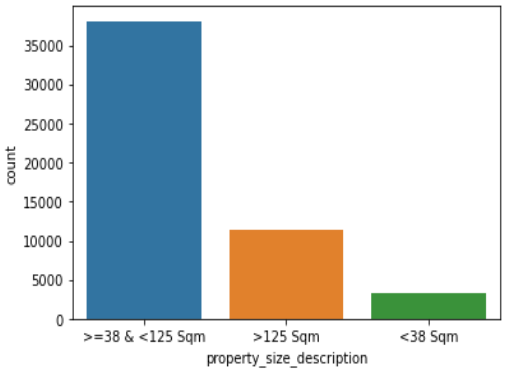


Figure 7.1. 9 Univariate for property size

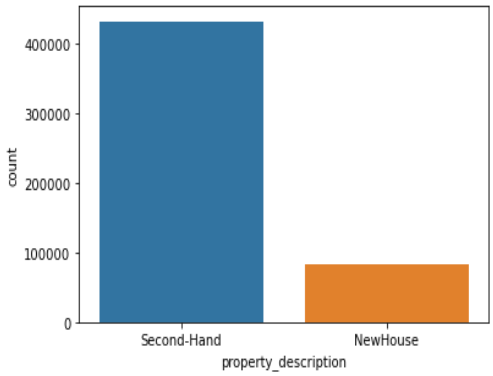


Figure 7.1. 10 Univariate for property description

Figures 7.1.11 - 7.1.18 displays the relationship between two variables in the dataset. Line plots, count plots and box plots were generated to analyse the relationship between the variables. Figure 7.1.11 shows that the house price shows a decrease in the year 2013 and then shows significant increase. This indicates that the house prices have been increasing considerably from about 2013 to 2022. The count plots reveals that higher number of properties were sold outside of Dublin but, when the whole observations are considered, sold properties are higher in Dublin than other counties and the second-hand properties are sold higher than the new properties. This may be because of the higher living conditions and job opportunities in Dublin. Apart from this the plots also shows that number of sold houses are decreasing each year. This may be a result of the higher house prices. The box plots show the outliers in each variable, the IQR of each data and the spread of data points. Figures 7.1.14, 7.1.15, 7.1.17 and 7.1.18 displays the same information as the count plot and it shows that most of the sold properties in County Dublin and most of them were second hand. Likewise, most of the sold houses were with size greater than or equal to 38 Sqm and least sold houses were with size less than 38 Sqm.

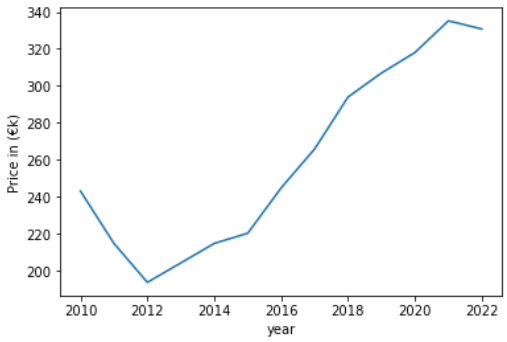


Figure 7.1. 11 Year-price bivariate

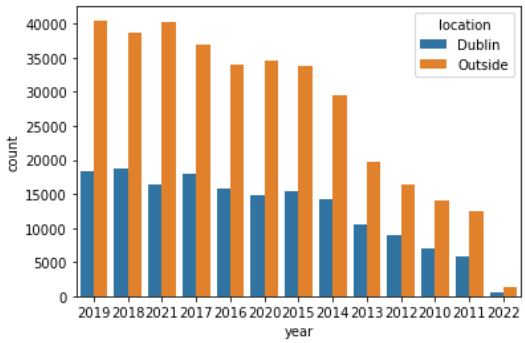


Figure 7.1. 12 Year-location bivariate

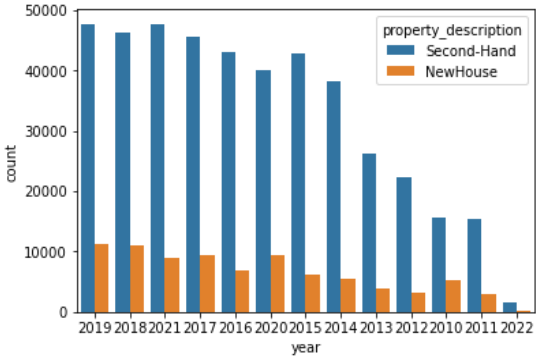


Figure 7.1. 13 Year-property description bivariate

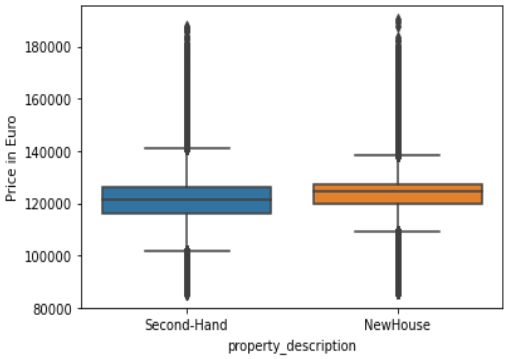


Figure 7.1. 16 Property description-price bivariate

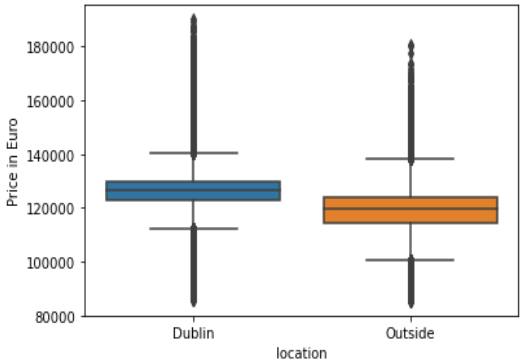


Figure 7.1. 14 location-price bivariate

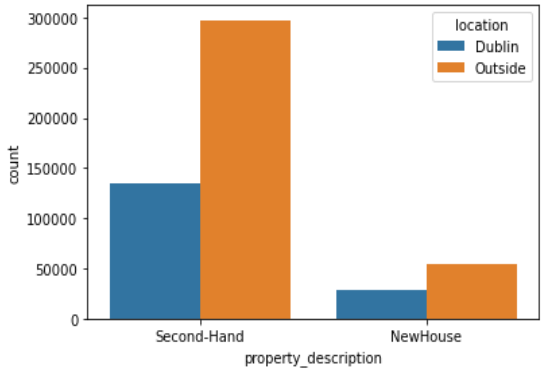


Figure 7.1. 17 location-property type bivariate

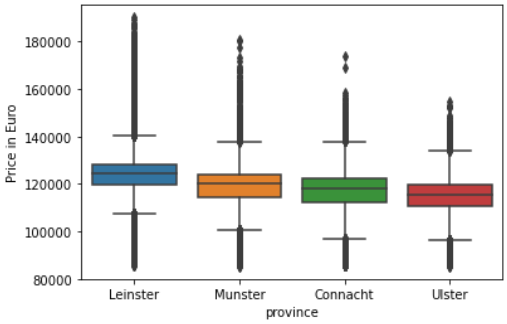


Figure 7.1. 15 Province-price bivariate

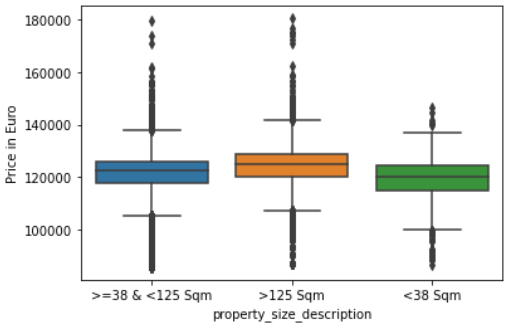


Figure 7.1. 18 Price-property size bivariate

Multivariate plots were generated to understand the relationship among several variables in the data. Line plots and box plots were implemented for this purpose. Line plots were used to analyse the data with respect to the year frame and box plots to analyse the spread, IQR and outliers in the data. Figures 7.1.19, 7.1.20 and 7.1.21 shows that the house price has been increasing significantly over the years in each county and province. It also shows that the house prices are very higher in Dublin than other counties. The box plots in figures 7.1.22, 7.1.23 and 7.1.24 reveals that there are outliers in the data

and the highest sold houses are second hand and had a size of greater than or equal to 38 Sqm.

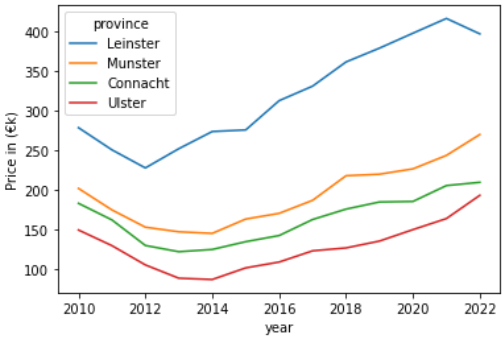


Figure 7.1. 19 Multivariate for Provinces

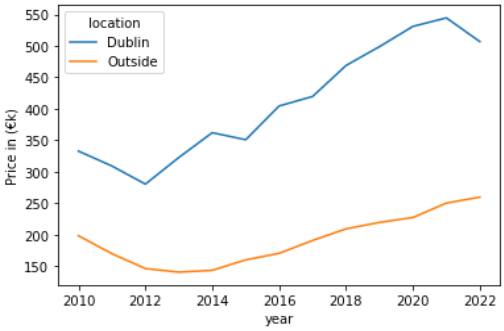


Figure 7.1. 20 Multivariate for location

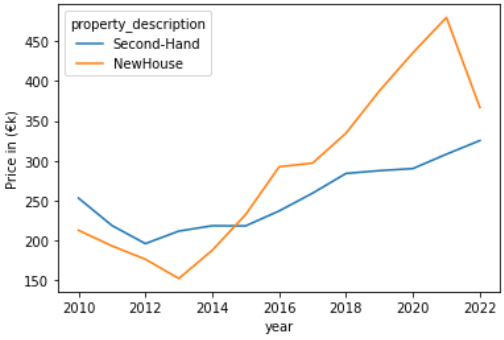


Figure 7.1. 21 Multivariate for property type

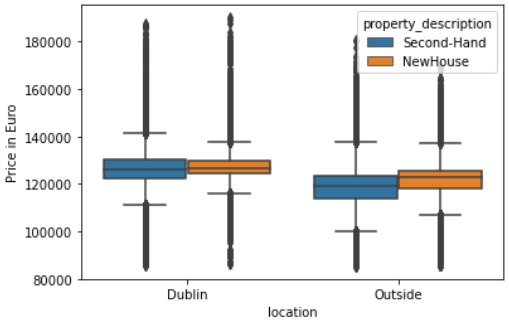


Figure 7.1. 22 Multivariate for property type and location

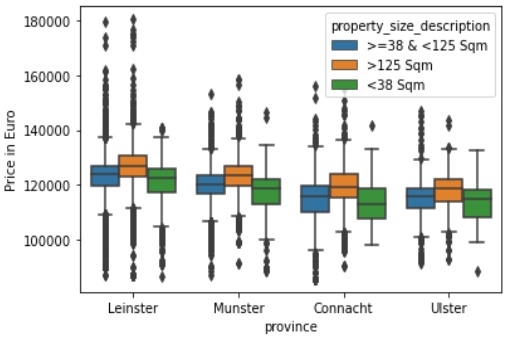


Figure 7.1. 23 Multivariate for property sizes and province

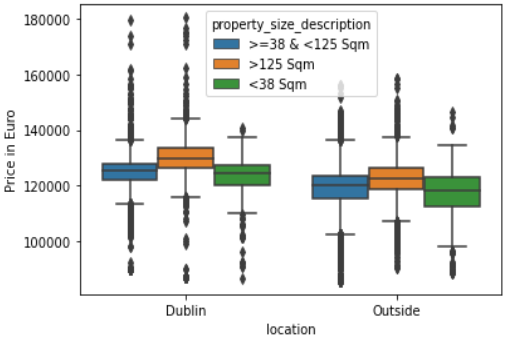


Figure 7.1. 24 Multivariate for location and property sizes

Further plots were made to understand the data for later analysis. The maximum price of sold-out property in each county were plotted. It is revealed that, Dublin has the sold properties with highest prices around 182 million Euro and the least sold price is in county Offaly with 1.4 million Euro.

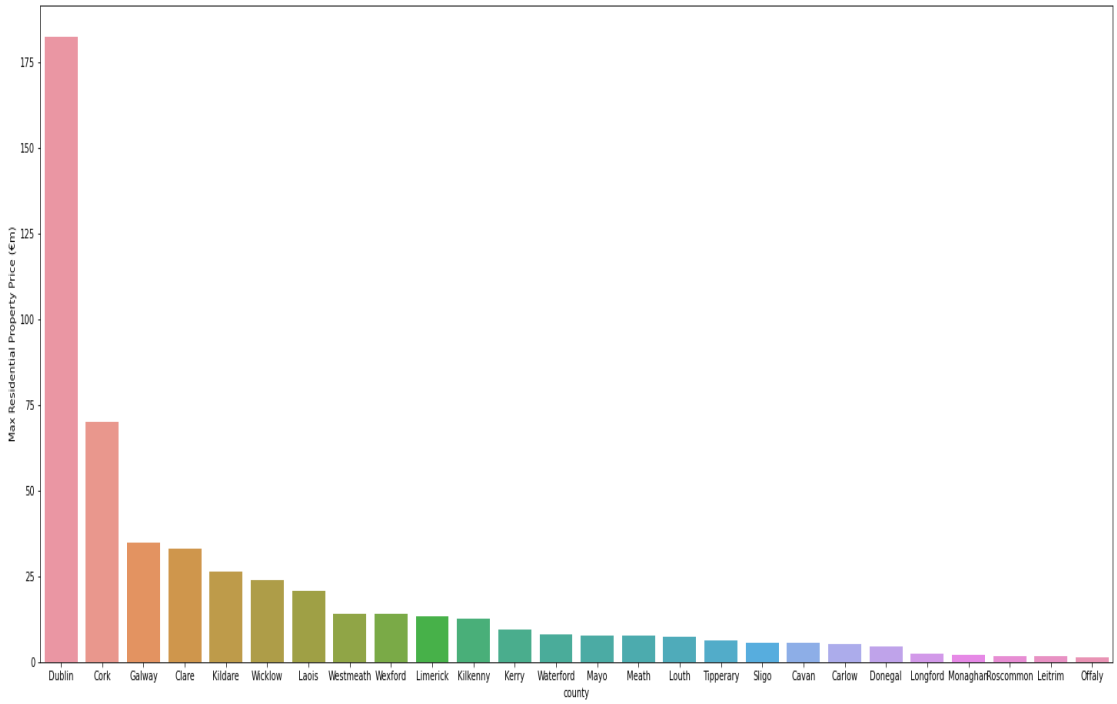


Figure 7.1. 25 Maximum property prices in counties.

Similarly, the minimum sold prices in each county is plotted and it shows that, county Cork has the minimum property price of 5000 Euro and county Carlow has the minimum property price of 7000 Euro.

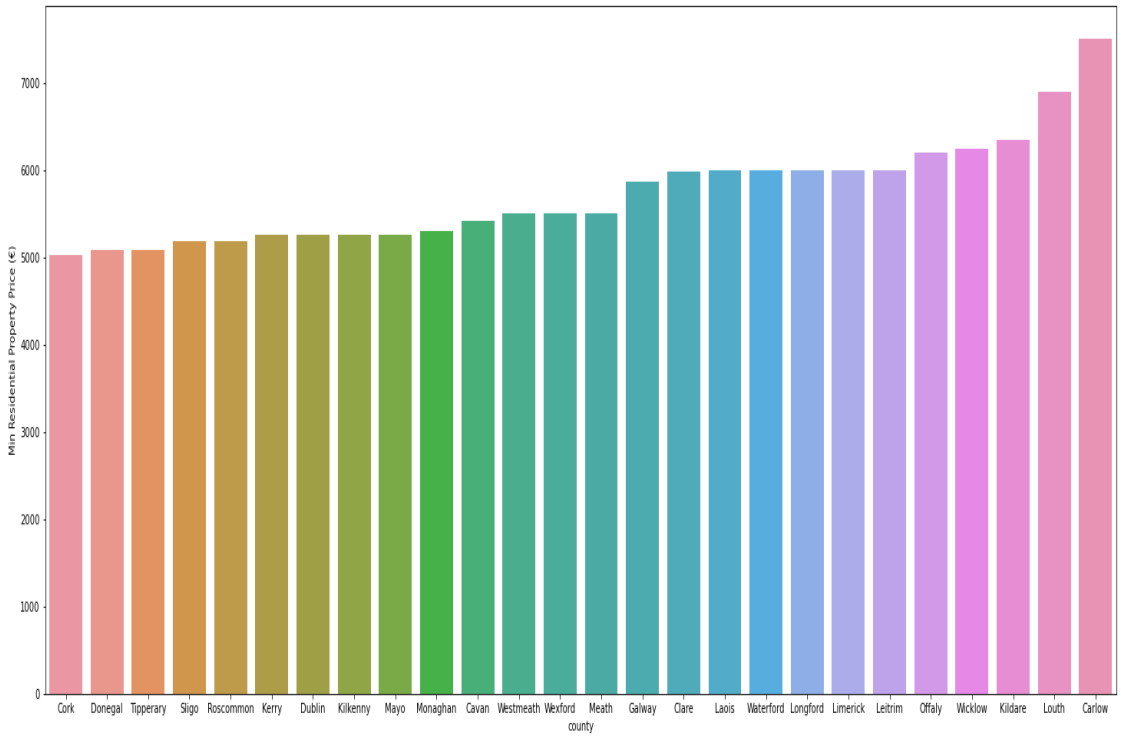


Figure 7.1. 26 Minimum property prices in counties

To understand the relation between the size and type of the property and price, a bar chart is plotted between the 'property_size_description' variable and the median of the prices and 'property_description' variable and median of the prices. From that, it is understood that the houses with greater than 125 square meters are having higher prices than the others and new houses are having higher prices than the second-hand houses.

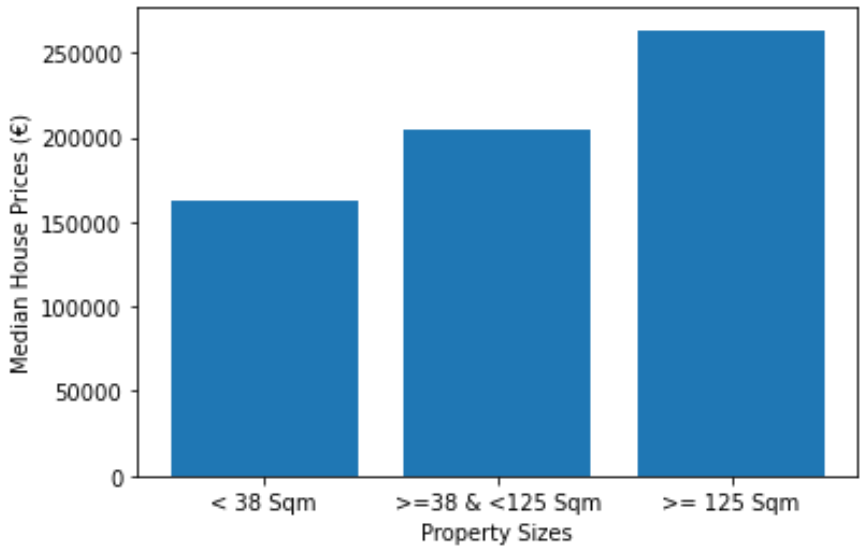


Figure 7.1. 27 Property sizes and median property prices

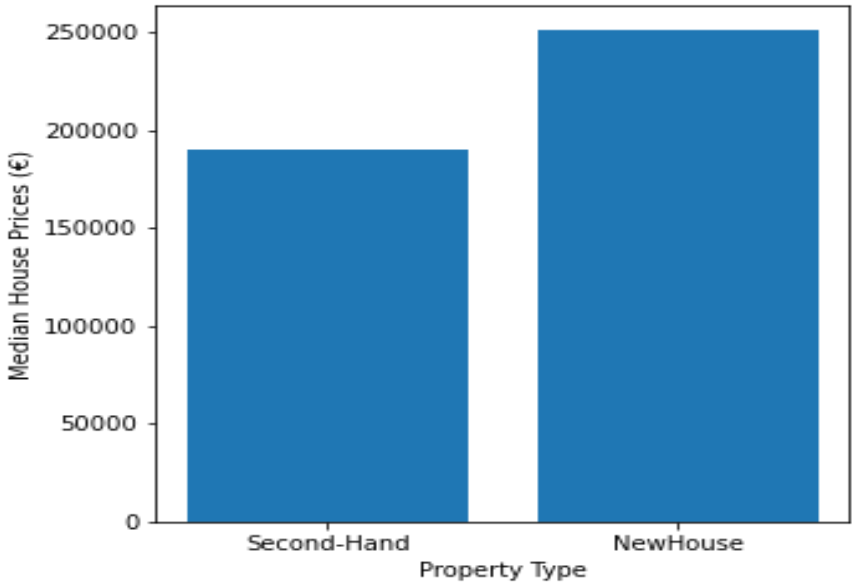


Figure 7.1. 28 Property types and median property prices

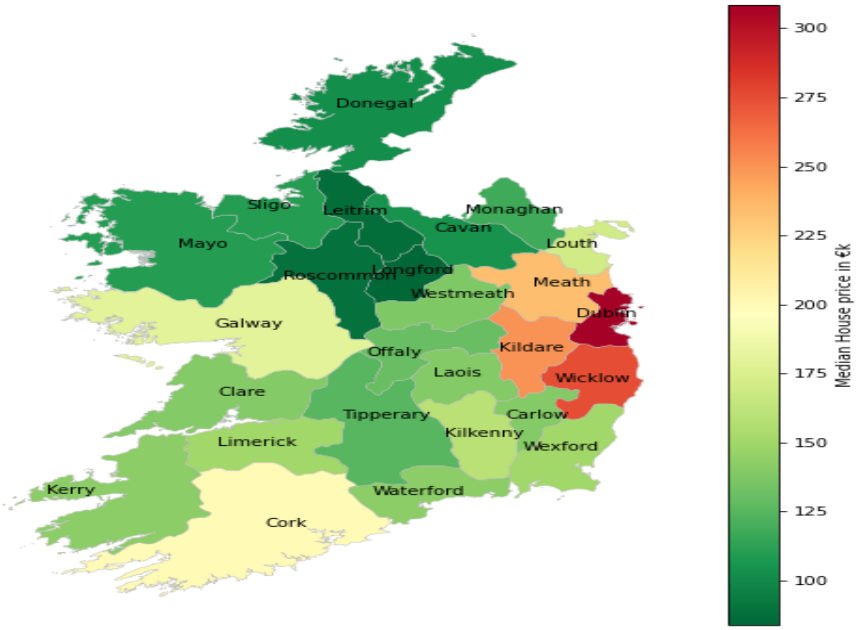


Figure 7.1. 29 Property prices and median property prices in counties

From the above choropleth, it is evident that property prices in Dublin are far higher than other counties. The counties Roscommon, Longford, Offlay and Leitrim are having considerably lower property prices when compared to Dublin. So, to analyse the price of properties inside Dublin a bar chart and choropleth is plot between the Dublin post codes and price. It shown that, within Dublin, the post code area Dublin 6 has the highest property prices and Dublin 10 has the lowest property prices. This implies that the post codes are also having an impact in property prices.

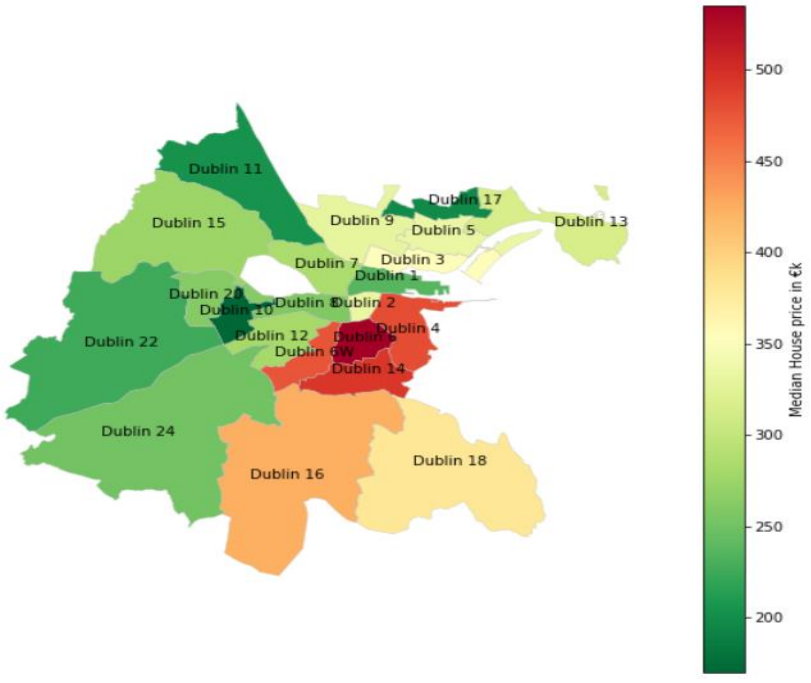


Figure 7.1. 30 Postal codes and median property prices

7.2 Statistical Analysis

For statistical analyses, a random sample of 500 observations are selected from the total population to get an accurate plots and results for ANOVA. The number of samples chosen was based on an online calculation that considered 95% confidence level and 5% margin of error (Qualitrics 2022a). The entire dataset is used to implement the MLR and SLR models for interpret the coefficients and to estimate the significant variables. ANOVA tests were performed as a part of initial analysis to understand the relationship between the dependent and independent variables. Price variable is chosen as the dependent variable and log of the price variable was applied to meet the assumptions of the tests.

7.2.1 ANOVA Test

The assumptions to be met for two-way ANOVA are as follows:

1. Observations should be sampled independently.
2. The variance of data in the different groups should be the same.
3. Each sample should be normally distributed.

A two-way model ANOVA is generated excluding the property size variable since it has many missing values.

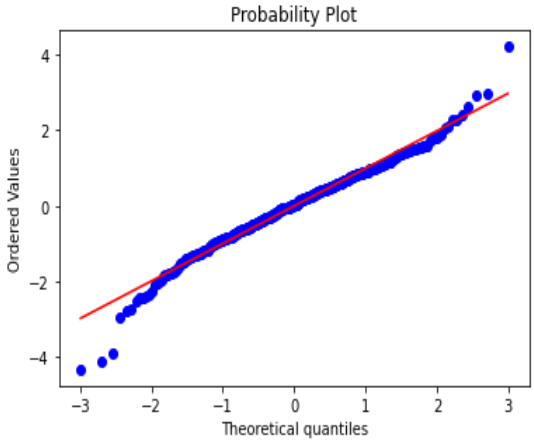


Figure 7.2.1. 1 Residuals Vs Fitted values Plot

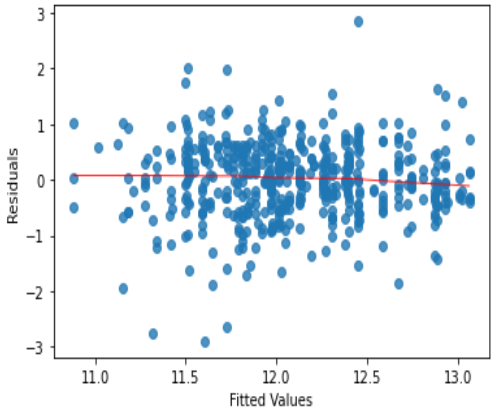


Figure 7.2.1. 2 Probability Plot

From the above figures it is implied that there is constant variance for the data points, and more than half of the data points are following the normality line in the

probability plot. So, the model is generated to predict the significance of the variables. From figure 7.2.1.3, it is revealed that that all the p values are less than 0.05 which shows that the variables are statistically significant.

	sum_sq	df	F	PR(>F)
C(year)	28.335014	12.0	5.027107	6.635126e-08
C(location)	26.370388	1.0	56.142591	3.248441e-13
C(property_description)	1.221570	1.0	2.600723	1.074695e-01
C(province)	12.478412	3.0	8.855518	1.007166e-05
Residual	226.397233	482.0	NaN	NaN

Figure 7.2.1. 3 ANOVA results for full model

Interpretation of the model as per *Figure 7.2.1.3.* and *Figure 7.2.1.4*

First Hypothesis:	Second Hypothesis:
H_0: μ_{2010} = μ_{2011} = μ_{2012} = μ_{2013} = μ_{2014}= μ_{2015} = μ_{2016} = μ_{2017}= μ_{2018} = μ_{2019}= μ_{2020}= μ_{2021} = μ_{2022} (All means are equal) Where μi is the sample mean for year i. H_1: Not all the means of the sample population is same. (at least on mean is different so, there is an effect with year on price) P-value = $6.6x10^{-8}$< 0.05, hence, reject H_0. Inference: Year is statistically significant; it is indicated that a change in year will affect the price or log price significantly.	H_0: $\mu_D = \mu_{OD}$ H_1: $\mu_D \neq \mu_{OD}$ Where: μ_D is the sample population mean for the log price when the property location is Dublin, μ_{OD} is the sample population mean for the log price when the property location is outside Dublin. P-value = $3.2x10^{-13}$< 0.05, hence, reject H_0. Inference: Location is statistically significant. So, changing the location will impact the price or log price significantly.

Third Hypothesis:	Fourth Hypothesis:
H_0: $\mu_{new} = \mu_{used}$ H_1: $\mu_{new} \neq \mu_{used}$ Where: μ_{new} is the sample population mean for the log price when the property is new, $\mu_{used}$is the sample population mean for the log price when the property second-hand. P-value = 1.07x 10^{-1}< 0.05, hence, reject H_0. Inference: Property type is statistically significant; it shows that changing the type will impact the price or log price significantly.	H_0: $\mu_{leinster} = \mu_{munster} = \mu_{ulster} = \mu_{connacht}$ H_1: Not all the means of the sample population is same. (at least on mean is different so, there is an effect with province on price) P-value = = 1.0x 10^{-6}<0.05. hence reject H_0. Inference: Province is statistically significant; it shows that changing the province will impact the price or log price significantly.

Table 7.2.1. 1 Two-way ANOVA interpretation

OLS Regression Results

Dep. Variable:	log_price	R-squared:	0.325
Model:	OLS	Adj. R-squared:	0.301
Method:	Least Squares	F-statistic:	13.66
Date:	Tue, 07 Jun 2022	Prob (F-statistic):	9.86e-32
Time:	17:07:14	Log-Likelihood:	-511.39
No. Observations:	500	AIC:	1059.
Df Residuals:	482	BIC:	1135.
Df Model:	17		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	12.6110	0.198	63.656	0.000	12.222	13.000
C(year)[T.2011]	-0.3512	0.199	-1.761	0.079	-0.743	0.041
C(year)[T.2012]	-0.4083	0.205	-1.996	0.046	-0.810	-0.006
C(year)[T.2013]	-0.4326	0.213	-2.029	0.043	-0.851	-0.014
C(year)[T.2014]	-0.3595	0.178	-2.023	0.044	-0.709	-0.010
C(year)[T.2015]	-0.6780	0.181	-3.751	0.000	-1.033	-0.323
C(year)[T.2016]	-0.1402	0.178	-0.787	0.431	-0.490	0.210
C(year)[T.2017]	-0.2228	0.175	-1.273	0.204	-0.567	0.121
C(year)[T.2018]	0.0757	0.175	0.432	0.666	-0.268	0.420
C(year)[T.2019]	0.0641	0.170	0.377	0.707	-0.270	0.399
C(year)[T.2020]	-0.0649	0.175	-0.370	0.711	-0.409	0.279
C(year)[T.2021]	0.1175	0.176	0.669	0.504	-0.227	0.462
C(year)[T.2022]	0.1833	0.423	0.433	0.665	-0.649	1.015
C(location)[T.Outside]	-0.6178	0.082	-7.493	0.000	-0.780	-0.456
C(property_description)[T.Second-Hand]	-0.1356	0.084	-1.613	0.107	-0.301	0.030
C(province)[T.Leinster]	0.3349	0.109	3.073	0.002	0.121	0.549
C(province)[T.Munster]	0.0953	0.111	0.854	0.393	-0.124	0.314
C(province)[T.Ulster]	-0.2971	0.156	-1.906	0.057	-0.604	0.009

Omnibus:	35.163	Durbin-Watson:	1.927
Prob(Omnibus):	0.000	Jarque-Bera (JB):	85.965
Skew:	-0.349	Prob(JB):	2.15e-19
Kurtosis:	4.907	Cond. No.	30.3

Figure 7.2.1. 4 Model summary for two-way ANOVA

7.2.2 Hedonic Pricing using MLR

The assumptions to be met for MLR are as follows:

1. There should be linear relationship between the dependent and independent variables.
2. The residuals should be normally distributed.
3. There should be no collinearity.

A MLR model is generated excluding the property size variable since it has many missing values. Square root transform is applied on the log value of price variable to deal with the outliers and make it symmetrical for better interpretation of the plots. However, the log price without the square root transform is fed to the model to get accurate results. A MLR model will be considered as Hedonic pricing model when the

dependent variable is price or log value of price. So here the MLR model is acting as a Hedonic model since our dependent variable is the log value of the price of the property.

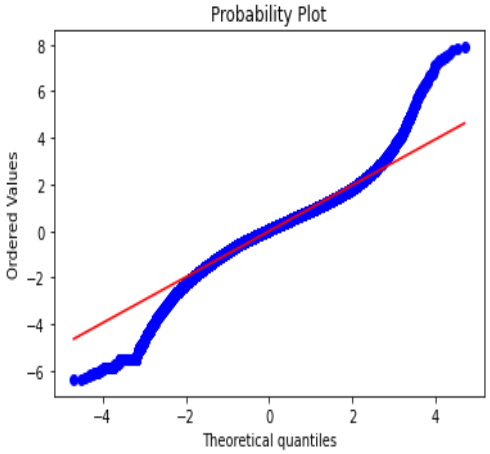


Figure 7.2.2. 1 Residuals Vs Fitted values Plot

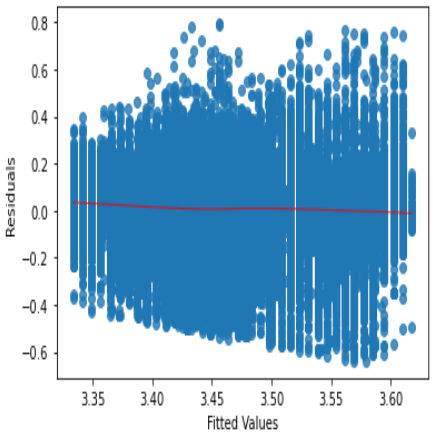


Figure 7.2.2. 2 Probability Plot

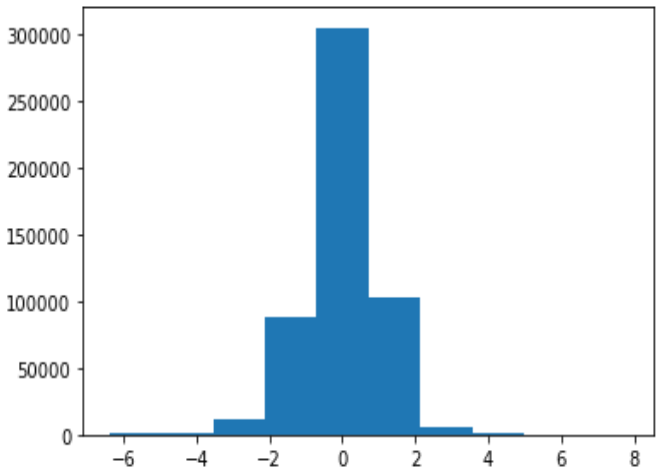


Figure 7.2.2. 3 Histogram for MLR

From the above figures it is implied that there is constant variance for the data points, and more than half of the data points are following the normality line in the probability plot. In addition to this, the histogram is almost symmetrical, so the model is generated to predict the significance of the variables. Figure 7.2.2. 4 shows the model summary and from that only 29 percentage of the variance in the data is being explained by the model. The standard error coefficients are positive with lower values. The p value of all variables except one variable is less than 0.05 which means the variables are significant.

OLS Regression Results

Dep. Variable:	log_price	R-squared:	0.228
Model:	OLS	Adj. R-squared:	0.228
Method:	Least Squares	F-statistic:	8472.
Date:	Mon, 05 Sep 2022	Prob (F-statistic):	0.00
Time:	13:17:46	Log-Likelihood:	-5.5626e+05
No. Observations:	516586	AIC:	1.113e+06
Df Residuals:	516567	BIC:	1.113e+06
Df Model:	18		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	21.1564	0.164	128.865	0.000	20.835	21.478
C(year)[T.2011]	-0.1552	0.007	-21.638	0.000	-0.169	-0.141
C(year)[T.2012]	-0.3209	0.007	-48.358	0.000	-0.334	-0.308
C(year)[T.2013]	-0.3800	0.006	-59.456	0.000	-0.393	-0.368
C(year)[T.2014]	-0.2768	0.006	-46.307	0.000	-0.288	-0.265
C(year)[T.2015]	-0.2015	0.006	-34.342	0.000	-0.213	-0.190
C(year)[T.2016]	-0.1019	0.006	-17.407	0.000	-0.113	-0.090
C(year)[T.2017]	0.0032	0.006	0.560	0.576	-0.008	0.015
C(year)[T.2018]	0.0755	0.006	13.171	0.000	0.064	0.087
C(year)[T.2019]	0.1213	0.006	21.244	0.000	0.110	0.133
C(year)[T.2020]	0.1523	0.006	26.007	0.000	0.141	0.164
C(year)[T.2021]	0.2444	0.006	42.535	0.000	0.233	0.256
C(year)[T.2022]	0.3149	0.017	18.347	0.000	0.281	0.349
C(province)[T.Leinster]	-0.1093	0.006	-17.434	0.000	-0.122	-0.097
C(province)[T.Munster]	-0.1061	0.006	-18.707	0.000	-0.117	-0.095
C(province)[T.Ulster]	-0.9669	0.008	-118.106	0.000	-0.983	-0.951
C(property_description)[T.Second-Hand]	-0.1311	0.003	-48.631	0.000	-0.136	-0.126
lat	-0.1225	0.003	-40.968	0.000	-0.128	-0.117
lon	0.3081	0.002	141.366	0.000	0.304	0.312

Omnibus:	34379.354	Durbin-Watson:	1.872
Prob(Omnibus):	0.000	Jarque-Bera (JB):	133193.650
Skew:	-0.239	Prob(JB):	0.00
Kurtosis:	5.441	Cond. No.	8.92e+03

Figure 7.2.2. 4 Model summary for MLR

Interpretation of the model coefficients as per Figure 7.2.2. 4.

y = 21.1- 0.15 year_2011 – 0.33 year_2012 - 0.38 year_2013 - 0.27 year_2014 - 0.20 year_2015 - 0.10 year_2016 + 0.003 year_2017 + 0.08 year_2018 +0.12 year_2019 + 0.15 year_2020 + 0.24 year_2021 + 0.31 year_2022 - 0.11 province_leinster - 0.11 province_munster - 0.97 province_ulster - 0.13 second_hand – 0.12 lat + 0.30 lon.

- 21.5 is the estimated intercept i.e., the mean average selling price of the house when all predictor variables are zero. This can also be interpreted as, the expected price (log price) when we have 2010 as the year, Connacht as province, with property type as new.
- Since the dependent variable is on log scale, the coefficient is exponentiated, then subtracted one from this number, and multiplied it by 100. This gives the percent increase (or decrease) in the response for every one-unit increase in the independent variable (Clay Ford 2018).
- Going from year 2010 to 2011, the average increase in property price will be -14%, keeping all other variables constant.
- Going from year 2010 to 2012, the average increase in property price will be -28%, keeping all other variables constant.
- Going from year 2010 to 2013, the average increase in property price will be -31.6%, keeping all other variables constant.
- Going from year 2010 to 2014, the average increase in property price will be -23.7%, keeping all other variables constant.
- Going from year 2010 to 2015, the average increase in property price will be -18.1%, keeping all other variables constant.
- Going from year 2010 to 2016, the average increase in property price will be -9.5%, keeping all other variables constant.
- Going from year 2010 to 2017, the average increase in property price will be 0.03%, keeping all other variables constant.
- Going from year 2010 to 2018, the average increase in property price will be 8.3%, keeping all other variables constant.
- Going from year 2010 to 2019, the average increase in property price will be 12.8%, keeping all other variables constant.

- Going from year 2010 to 2020, the average increase in property price will be 16.2%, keeping all other variables constant.
- Going from year 2010 to 2021, the average increase in property price will be 27.1%, keeping all other variables constant.
- Going from year 2010 to 2022, the average increase in property price will be 36.3%, keeping all other variables constant.
- Moving from Connacht to province_leinster, the average increase in property price will be -10.4%, keeping all other variables constant.
- Moving from Connacht to province_munster, the average increase in property price will be -10.4%, keeping all other variables constant.
- Moving from Connacht to province_ulster, the average increase in property price will be -62%, keeping all other variables constant.
- Moving from new to second_hand, the average increase in property price will be -12.2%, keeping all other variables constant.
- For 1 unit increase in latitude, the average increase in property price will be -11.3%, keeping all other variables constant
- For 1 unit increase in longitude, the average increase in property price will be 35%, keeping all other variables constant
- The property price is having positive increase after year 2017, it shows that the price has been increased significantly after 2017.

All p-values are < 0.05, except for year_2017 which is 0.08 this implies that the variables year, province, and property_description are statistically significant.

Table 7.2.2. 1 MLR Results interpretation

7.2.3 SLR on Dublin Data

To analyse the impact of Post codes on property prices, a new dataset was created from the original dataset with the details of County Dublin. The assumptions to be met for SLR are as follows:

1. There should be linear relationship between the dependent and independent variable.
2. The residuals should be normally distributed.
3. There should be no relationship between the residuals and the independent variable.

A SLR model is generated using the price and the post codes. Square root transform is applied on the log value of price variable to deal with the outliers and make it symmetrical. However, the log price without transform is fed to the model to get accurate results.

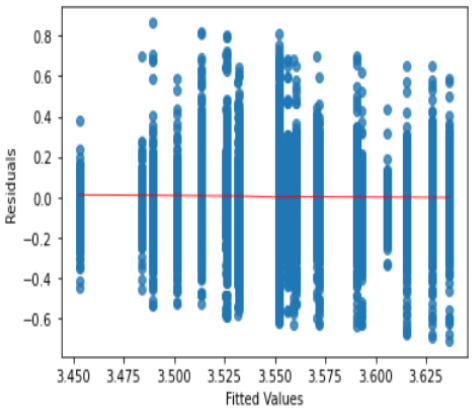


Figure 7.2.3. 1 Residuals Vs Fitted values Plot

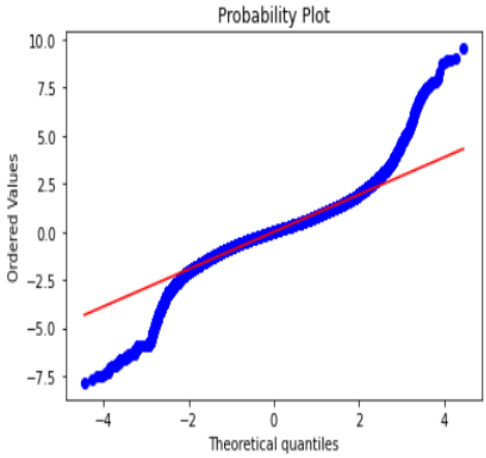


Figure 7.2.3. 2 Probability Plot

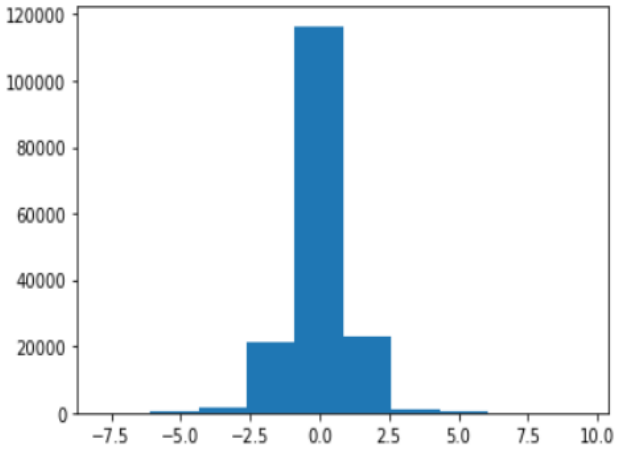


Figure 7.2.3. 3 Histogram for SLR

From the Figures 7.2.3.1 and 7.2.3.2 it is indicated that there is constant variance for the data points, and more than half of the data points are following the normality line in the probability plot. In addition to this, the histogram is almost symmetrical, so the model is generated to predict the significance of the variables. Figure 7.2.3.4 shows the model summary and from that only 19 percentage of the variance in the data is being explained by the model. The standard error coefficients are positive with lower values. The p value of all variables is less than 0.05 which means the variables are statistically significant.

OLS Regression Results

Dep. Variable:	log_price	R-squared:	0.192
Model:	OLS	Adj. R-squared:	0.192
Method:	Least Squares	F-statistic:	1099.
Date:	Fri, 05 Aug 2022	Prob (F-statistic):	0.00
Time:	11:01:20	Log-Likelihood:	-90954.
No. Observations:	96825	AIC:	1.820e+05
Df Residuals:	96803	BIC:	1.822e+05
Df Model:	21		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	12.3552	0.011	1119.318	0.000	12.334	12.377
C(postal_code)[T.Dublin 10]	-0.4243	0.020	-20.879	0.000	-0.464	-0.384
C(postal_code)[T.Dublin 11]	-0.1709	0.014	-12.223	0.000	-0.198	-0.143
C(postal_code)[T.Dublin 12]	0.1221	0.014	8.604	0.000	0.094	0.150
C(postal_code)[T.Dublin 13]	0.3297	0.014	23.336	0.000	0.302	0.357
C(postal_code)[T.Dublin 14]	0.7217	0.014	50.397	0.000	0.694	0.750
C(postal_code)[T.Dublin 15]	0.1308	0.012	10.538	0.000	0.106	0.155
C(postal_code)[T.Dublin 16]	0.5594	0.014	38.753	0.000	0.531	0.588
C(postal_code)[T.Dublin 17]	-0.2118	0.022	-9.843	0.000	-0.254	-0.170
C(postal_code)[T.Dublin 18]	0.5427	0.013	40.758	0.000	0.517	0.569
C(postal_code)[T.Dublin 2]	0.4033	0.018	22.324	0.000	0.368	0.439
C(postal_code)[T.Dublin 20]	0.0825	0.021	3.880	0.000	0.041	0.124
C(postal_code)[T.Dublin 22]	-0.0914	0.016	-5.742	0.000	-0.123	-0.060
C(postal_code)[T.Dublin 24]	0.0399	0.013	2.995	0.003	0.014	0.066
C(postal_code)[T.Dublin 3]	0.4124	0.015	28.371	0.000	0.384	0.441
C(postal_code)[T.Dublin 4]	0.8159	0.014	59.150	0.000	0.789	0.843
C(postal_code)[T.Dublin 5]	0.3172	0.015	21.288	0.000	0.288	0.346
C(postal_code)[T.Dublin 6]	0.8800	0.015	60.569	0.000	0.852	0.908
C(postal_code)[T.Dublin 6w]	0.6512	0.025	26.119	0.000	0.602	0.700
C(postal_code)[T.Dublin 7]	0.1673	0.014	12.044	0.000	0.140	0.195
C(postal_code)[T.Dublin 8]	0.0895	0.014	6.522	0.000	0.063	0.116
C(postal_code)[T.Dublin 9]	0.2975	0.014	21.498	0.000	0.270	0.325

Omnibus:	16165.422	Durbin-Watson:	1.511
Prob(Omnibus):	0.000	Jarque-Bera (JB):	311906.290
Skew:	-0.192	Prob(JB):	0.00
Kurtosis:	11.784	Cond. No.	27.5

Figure 7.2.3. 4 Model summary for SLR

Interpretation of the model coefficients as per Figure 7.2.3.4

y = 12.36 - 0.42 Dublin10 – 0.17 Dublin11 + 0.12 Dublin12 + 0.33 Dublin13 + 0.72 Dublin14 + 0.13 Dublin15 + 0.56 Dublin16 – 0.21 Dublin17 + 0.54 Dublin18 + 0.40 Dublin2 + 0.08 Dublin20 - 0.09 Dublin22 + 0.03 Dublin24 + 0.41Dublin3 + 0.81 Dublin4 + 0.32 Dublin5 + 0.88 Dublin6 + 0.65 Dublin6w + 0.16 Dublin7 + 0.09 Dublin8 + 0.3 Dublin9.

- 12.36 is the estimated intercept i.e., the mean average selling price of the house when all predictor variables are zero. So, this can also be interpreted as, the expected price (log price) when we have Dublin 1 as the postcode.
- Since the dependent variable is on log scale, the coefficient is exponentiated, then subtracted one from this number, and multiplied it by 100. This gives the percent increase (or decrease) in the response for every one-unit increase in the independent variable (Clay Ford 2018).
- When going from Dublin1 to Dublin10, the average increase in property price will be -34.3%, keeping all other variables constant.
- When going from Dublin1 to Dublin11, the average increase in property price will be -15.6%, keeping all other variables constant.
- When going from Dublin1 to Dublin12, the average increase in property price will be 12.8%, keeping all other variables constant.
- When going from Dublin1 to Dublin13, the average increase in property price will be 39.1%, keeping all other variables constant.
- When going from Dublin1 to Dublin14, the average increase in property price will be 99.5%, keeping all other variables constant.
- When going from Dublin1 to Dublin15, the average increase in property price will be 13.9%, keeping all other variables constant.
- When going from Dublin1 to Dublin16, the average increase in property price will be 75%, keeping all other variables constant.
- When going from Dublin1 to Dublin17, the average increase in property price will be -19%, keeping all other variables constant.
- When going from Dublin1 to Dublin18, the average increase in property price will be 72%, keeping all other variables constant.

- When going from Dublin1 to Dublin2, the average increase in property price will be 49.2%, keeping all other variables constant.
- When going from Dublin1 to Dublin20, the average increase in property price will be 8.3%, keeping all other variables constant.
- When going from Dublin1 to Dublin22, the average increase in property price will be -8.6%, keeping all other variables constant.
- When going from Dublin1 to Dublin24, the average increase in property price will be 4.1%, keeping all other variables constant.
- When going from Dublin1 to Dublin3, the average increase in property price will be 50.7%, keeping all other variables constant.
- When going from Dublin1 to Dublin4, the average increase in property price will be 99%, keeping all other variables constant.
- When going from Dublin1 to Dublin5, the average increase in property price will be 37.8%, keeping all other variables constant.
- When going from Dublin1 to Dublin6, the average increase in property price will be 99.1%, keeping all other variables constant.
- When going from Dublin1 to Dublin6w, the average increase in property price will be 91.5%, keeping all other variables constant.
- When going from Dublin1 to Dublin7, the average increase in property price will be 18.5%, keeping all other variables constant.
- When going from Dublin1 to Dublin8, the average increase in property price will be 9.4%, keeping all other variables constant.
- When going from Dublin1 to Dublin9, the average increase in property price will be 3%, keeping all other variables constant.

All p-values are < 0.05 this implies that the post codes statistically significant and influences the property prices.

Table 7.2.3. 1 SLR Results interpretation

7.2.4 MLR on Dublin Data

The assumptions to be met for MLR are as follows:

1. There should be linear relationship between the dependent and independent variables.
2. The residuals should be normally distributed.
3. There should be no collinearity.

A MLR model is generated excluding the property size variable since it has many missing values. Square root transform is applied on the log value of price variable to deal with the outliers and make it symmetrical. However, the log price without transform is fed to the model to get accurate results.

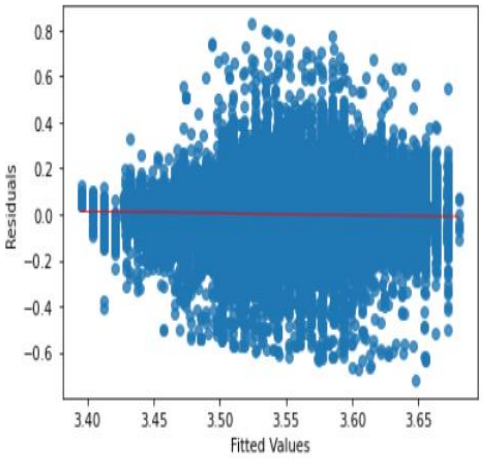


Figure 7.2.4. 1 Residuals Vs Fitted values Plot

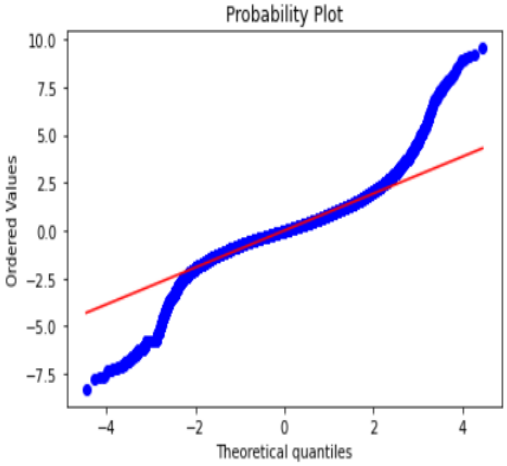


Figure 7.2.4. 2 Probability Plot

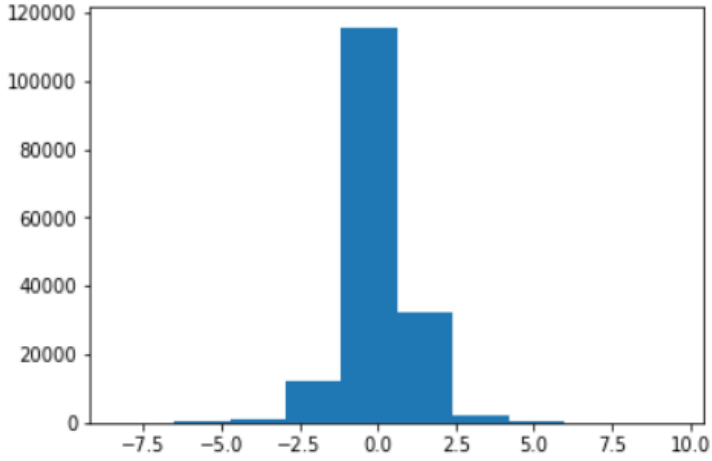


Figure 7.2.4. 3 Histogram

The figures 7.2.4.1 and 7.2.4.2 shows that there is constant variance for the data points, and more than half of the data points are following the normality line in the probability plot. In addition to this, the histogram is almost symmetrical, so the model is generated to predict the significance of the variables. Figure 7.2.4.4 shows the model summary and from that only 29 percentage of the variance in the data is being explained by the model. The standard error coefficients are positive with lower values. The p value of all variables except one variable is less than 0.05 which means the variables are statistically significant.

OLS Regression Results

Dep. Variable:	log_price	**R-squared:**	0.281
Model:	OLS	**Adj. R-squared:**	0.281
Method:	Least Squares	**F-statistic:**	1114.
Date:	Thu, 04 Aug 2022	**Prob (F-statistic):**	0.00
Time:	18:58:22	**Log-Likelihood:**	-85320.
No. Observations:	96825	**AIC:**	1.707e+05
Df Residuals:	96790	**BIC:**	1.710e+05
Df Model:	34		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0042	5.72e-06	738.484	0.000	0.004	0.004
C(year)[T.2011]	-0.1707	0.018	-9.306	0.000	-0.207	-0.135
C(year)[T.2012]	-0.3598	0.017	-21.785	0.000	-0.392	-0.327
C(year)[T.2013]	-0.3115	0.014	-21.745	0.000	-0.340	-0.283
C(year)[T.2014]	0.0155	0.013	1.161	0.246	-0.011	0.042
C(year)[T.2015]	0.0426	0.013	3.282	0.001	0.017	0.068
C(year)[T.2016]	0.1505	0.013	11.594	0.000	0.125	0.176
C(year)[T.2017]	0.2476	0.013	19.247	0.000	0.222	0.273
C(year)[T.2018]	0.3112	0.013	24.226	0.000	0.286	0.336
C(year)[T.2019]	0.3415	0.013	26.628	0.000	0.316	0.367
C(year)[T.2020]	0.3530	0.013	26.979	0.000	0.327	0.379
C(year)[T.2021]	0.4218	0.013	32.555	0.000	0.396	0.447
C(year)[T.2022]	0.4276	0.033	12.810	0.000	0.362	0.493
C(postal_code)[T.Dublin 10]	-0.4472	0.019	-23.316	0.000	-0.485	-0.410
C(postal_code)[T.Dublin 11]	-0.1914	0.013	-14.493	0.000	-0.217	-0.166
C(postal_code)[T.Dublin 12]	0.0959	0.013	7.152	0.000	0.070	0.122
C(postal_code)[T.Dublin 13]	0.2814	0.013	21.027	0.000	0.255	0.308
C(postal_code)[T.Dublin 14]	0.6939	0.014	51.312	0.000	0.667	0.720
C(postal_code)[T.Dublin 15]	0.0859	0.012	7.312	0.000	0.063	0.109
C(postal_code)[T.Dublin 16]	0.5269	0.014	38.632	0.000	0.500	0.554
C(postal_code)[T.Dublin 17]	-0.2279	0.020	-11.206	0.000	-0.268	-0.188
C(postal_code)[T.Dublin 18]	0.5082	0.013	40.264	0.000	0.483	0.533
C(postal_code)[T.Dublin 2]	0.3986	0.017	23.379	0.000	0.365	0.432
C(postal_code)[T.Dublin 20]	0.0559	0.020	2.781	0.005	0.016	0.095
C(postal_code)[T.Dublin 22]	-0.1576	0.015	-10.484	0.000	-0.187	-0.128
C(postal_code)[T.Dublin 24]	-0.0353	0.013	-2.797	0.005	-0.060	-0.011
C(postal_code)[T.Dublin 3]	0.3954	0.014	28.810	0.000	0.368	0.422
C(postal_code)[T.Dublin 4]	0.7969	0.013	61.177	0.000	0.771	0.822
C(postal_code)[T.Dublin 5]	0.2962	0.014	21.047	0.000	0.269	0.324
C(postal_code)[T.Dublin 6]	0.8627	0.014	62.887	0.000	0.836	0.890
C(postal_code)[T.Dublin 6w]	0.7735	0.024	32.492	0.000	0.727	0.820
C(postal_code)[T.Dublin 7]	0.1446	0.013	11.018	0.000	0.119	0.170
C(postal_code)[T.Dublin 8]	0.0656	0.013	5.067	0.000	0.040	0.091
C(postal_code)[T.Dublin 9]	0.2663	0.013	20.359	0.000	0.241	0.292
C(property_description)[T.Second-Hand]	0.0161	0.006	2.710	0.007	0.004	0.028
lat	0.2252	0.000	738.484	0.000	0.225	0.226
lon	-0.0264	3.58e-05	-738.484	0.000	-0.026	-0.026

Omnibus:	16685.870	**Durbin-Watson:**	1.696
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	391302.189
Skew:	0.042	**Prob(JB):**	0.00
Kurtosis:	12.848	**Cond. No.**	4.08e+15

Figure 7.2.4. 4 Model Summary for MLR

Interpretation of the model coefficients as per Figure 7.2.4.4.

y = 0.0042 - 0.17 year_2011 – 0.36 year_2012 - 0.31year_2013 + 0.015 year_2014 + 0.04 year_2015 + 0.15 year_2016 + 0.24year_2017 + 0.31 year_2018 +0.34 year_2019 + 0.35 year_2020 + 0.42 year_2021 + 0.43 year_2022 - 0.44 Dublin10 – 0.19 Dublin11 + 0.09 Dublin12 + 0.28 Dublin13 + 0.69 Dublin14 + 0.08 Dublin15 + 0.52 Dublin16 – 0.22 Dublin17 + 0.51 Dublin18 + 0.4 Dublin2 + 0.05 Dublin20 - 0.15 Dublin22 - 0.03 Dublin24 + 0.39 Dublin3 + 0.8 Dublin4 + 0.3 Dublin5 + 0.86 Dublin6 + 0.77 Dublin6w + 0.14 Dublin7 + 0.06 Dublin8 + 0.27 Dublin9 + 0.02 second_hand + 0.22 lat – 0.03 lon.

- 0. 0042 is the estimated intercept i.e., the mean average selling price of the house when all predictor variables are zero. So, this can also be interpreted as, the expected price (log price) when we have 2010 as the year, Dublin 1 as postcode, with type new property.
- Since the dependent variable is on log scale, the coefficient is exponentiated, then subtracted one from this number, and multiplied it by 100. This gives the percent increase (or decrease) in the response for every one-unit increase in the independent variable (Clay Ford 2018).
- Going from year 2010 to 2011, the average increase in property price will be -15.6%, keeping all other variables constant.
- Going from year 2010 to 2012, the average increase in property price will be -30.2%, keeping all other variables constant.
- Going from year 2010 to 2013, the average increase in property price will be -26.7%, keeping all other variables constant.
- Going from year 2010 to 2014, the average increase in property price will be 1.5%, keeping all other variables constant.
- Going from year 2010 to 2015, the average increase in property price will be 4%, keeping all other variables constant.
- Going from year 2010 to 2016, the average increase in property price will be 16%, keeping all other variables constant.
- Going from year 2010 to 2017, the average increase in property price will be 28.4%, keeping all other variables constant.

- Going from year 2010 to 2018, the average increase in property price will be 36.3%, keeping all other variables constant.
- Going from year 2010 to 2019, the average increase in property price will be 38.3%, keeping all other variables constant.
- Going from year 2010 to 2020, the average increase in property price will be 37.2%, keeping all other variables constant.
- Going from year 2010 to 2021, the average increase in property price will be 52.2%, keeping all other variables constant.
- Going from year 2010 to 2022, the average increase in property price will be 53.7%, keeping all other variables constant.
- Going from Dublin1 to Dublin10, the average increase in property price will be -36.2%, keeping all other variables constant.
- Going from Dublin1 to Dublin11, the average increase in property price will be -18.1%, keeping all other variables constant.
- Going from Dublin1 to Dublin12, the average increase in property price will be 9.4%, keeping all other variables constant.
- Going from Dublin1 to Dublin13, the average increase in property price will be 32.3%, keeping all other variables constant.
- Going from Dublin1 to Dublin14, the average increase in property price will be 99.4%, keeping all other variables constant.
- Going from Dublin1 to Dublin15, the average increase in property price will be 13.9%, keeping all other variables constant.
- Going from Dublin1 to Dublin16, the average increase in property price will be 8.32%, keeping all other variables constant.
- Going from Dublin1 to Dublin17, the average increase in property price will be -20.5%, keeping all other variables constant.
- Going from Dublin1 to Dublin18, the average increase in property price will be 66.5%, keeping all other variables constant.
- Going from Dublin1 to Dublin2, the average increase in property price will be 49.2%, keeping all other variables constant.
- Going from Dublin1 to Dublin20, the average increase in property price will be 5.1%, keeping all other variables constant.

- Going from Dublin1 to Dublin22, the average increase in property price will be -14.8%, keeping all other variables constant.
- Going from Dublin1 to Dublin24, the average increase in property price will be -3%, keeping all other variables constant.
- Going from Dublin1 to Dublin3, the average increase in property price will be 47.7%, keeping all other variables constant.
- Going from Dublin1 to Dublin4, the average increase in property price will be 98.4%, keeping all other variables constant.
- Going from Dublin1 to Dublin5, the average increase in property price will be 35%, keeping all other variables constant.
- Going from Dublin1 to Dublin6, the average increase in property price will be 99.1%, keeping all other variables constant.
- Going from Dublin1 to Dublin6w, the average increase in property price will be 98%, keeping all other variables constant.
- Going from Dublin1 to Dublin7, the average increase in property price will be 15%, keeping all other variables constant.
- Going from Dublin1 to Dublin8, the average increase in property price will be 6.1%, keeping all other variables constant.
- Going from Dublin1 to Dublin9, the average increase in property price will be 31%, keeping all other variables constant.
- Going from new to second-hand, the average increase in property price will be 0.02, keeping all other variables constant.
- For 1 unit increase in latitude, the average increase in property price will be 2%, keeping all other variables constant
- For 1 unit increase in longitude, the average increase in property price will be -3%, keeping all other variables constant

Almost all the p-values are < 0.05 this implies that the post codes statistically significant and influences the property prices.

Table 7.2.4. 1 MLR Results interpretation for Dublin

7.3 ML Models

Machine learning algorithms were used to analyse and predict the property prices. The dataset had 515792 rows × 15 columns. The dataset was divided into training and

test data with 80 percentage training and 20 percentage of test data. The training data consists of 412633 observations and the test data has 103159 observations. The ML models are mainly classified as three supervised, semi supervised, and reinforced models. Here for predicting the house prices supervised ML algorithms are considered since the data has labels.

Python and Jupyter notebook were used for implementing the models using the scikit and keras library. The ML modelling consists of data ingestion, data cleaning, exploratory data analysis, feature engineering and finally machine learning. The data is cleaned prior to the modelling and the NaN values and insignificant columns are removed to perform the modelling. Since the dependent variable is numeric, ML regression algorithms were used to analyse the data. Dummy variables were created for the categorical variables in the data to make it appropriate to fed to the model. A correlation matrix was plotted to understand the relationship of the variables and to identify the correlations in the data. There were no strong correlations in the data and hence it is acceptable to continue with modelling without altering any variables.

Figure 7.3. 1 Correlation Matrix

The standard development cycle for ML is followed in the project to develop and evaluate the models. Standard scaler was chosen for pre-processing and scaling the data prior to modelling. LR, RF, DCT, XG-Boost, SVR and NN algorithms were utilized for building the regression models. At first the models were implemented using the scaled

data and the performance metrices were evaluated and after that data without scaling is fed to the models to analyse the performance. It is found that the scaled data gives less performance for some models compared to unscaled data and hence some models were implemented without data scaling. Hence, SVM regressor was performed with scaling and other algorithms were implemented without scaling data. Apart from this, resampling was performed prior to SVM modelling. This is because the SVM modelling with the entire data was not getting executed and thus resampled data is used. RMSE, MSE, MAE and r squared error were utilized for evaluating the performance of the models to identify the best fit model. The RMSE, MSE and MAE values were found to be large values, this is because we are dealing with real and economic data and there is chance for the performance metrics of economic data to be high values.

7.3.1 Linear Regression Results

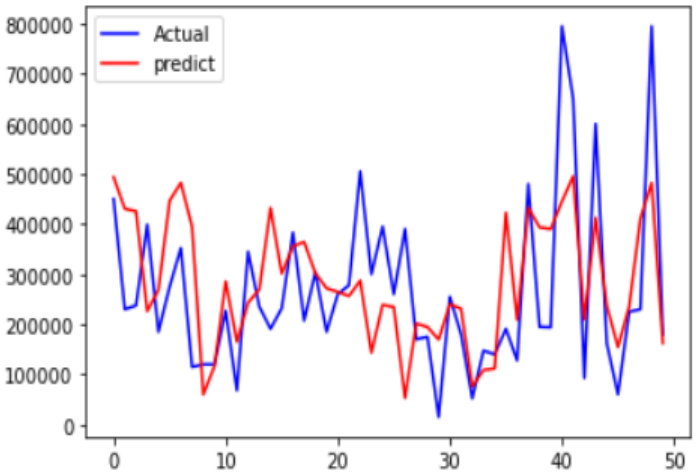


Figure 7.3.1. 1 Actual Vs Predicted Values in Linear Regression

The figure 7.3.1.1 shows the zoomed portion of line plot of the fifty actual values and predicted values using LR. The actual values are plotted in blue colour and the predicted values in red colour. The predicted values are different from the actual values in almost all areas. The RMSE value of the model is 803454 and the MAE is 135139 while the r squared value is relatively low with only 0.02. The r squared measures how close the data are fitted to the regression line and for the data fitted closer to the regression line, r squared value will be higher and for lower r squared values, the data will be scattered around the regression line. In LR algorithm, the r squared is considerably low and hence most of the data is dispersed around the regression line as shown in figure 7.3.1.2.

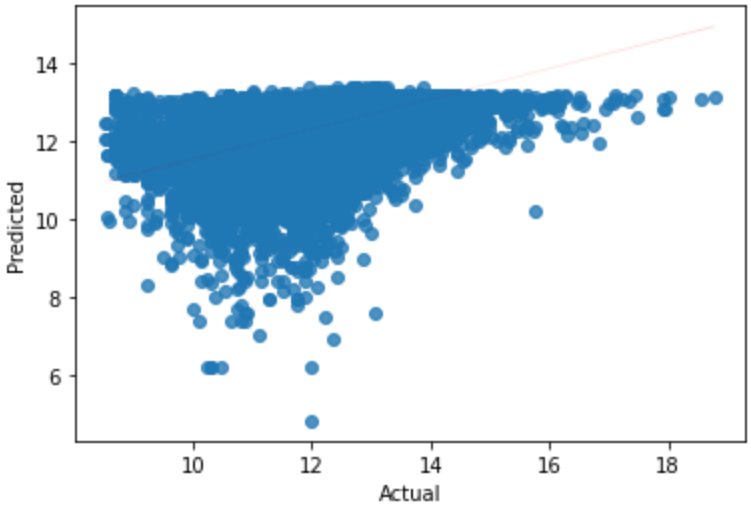


Figure 7.3.1. 2 Scatter plot of predicted values by Linear Regression

Linear Regression Model			
RMSE	**MSE**	**MAE**	**R Squared**
803454.106	645538500662.903	135139.365	0.023

Table 7.3.1. 1 Performance metrices of Linear Regression

7.3.2 Decision Tree Regression Results

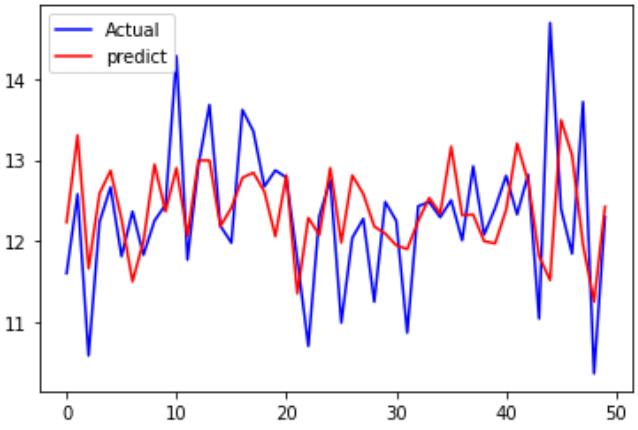


Figure 7.3.2. 1 Actual Vs Predicted Values in DCT Regression

The figure 7.3.2.1 shows the line plot of the first fifty actual values and predicted values using DT. The actual values are plotted in blue colour and the predicted values in red colour. The predicted values are different from the actual values in some areas. The RMSE value of the DCT model is 871447 and the MAE is 137785 which is higher while the r squared value is 0.01. The r squared measures how close the data are fitted to the regression line and for the data fitted closer to the regression line, r squared value will be higher and for lower r squared values, the data will be scattered around the regression

line. In DT algorithm, the r squared is very low and hence most of the data is scattered around the regression line as shown in figure 7.3.2.2.

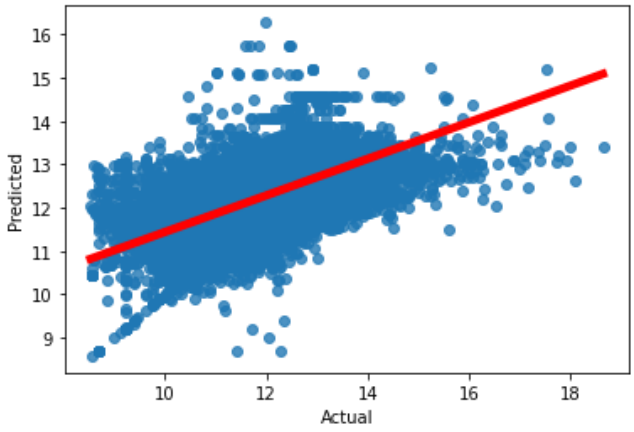


Figure 7.3.2. 2 Scatter plot of predicted values by DCT Regression

Decision Tree Regression Model			
RMSE	**MSE**	**MAE**	**R Squared**
913160	833862387387.05	139920.77	0.01

Table 7.3.2. 1 Performance metrices of Decision Tree Regression

7.3.3 Support Vector Machine Regression Results

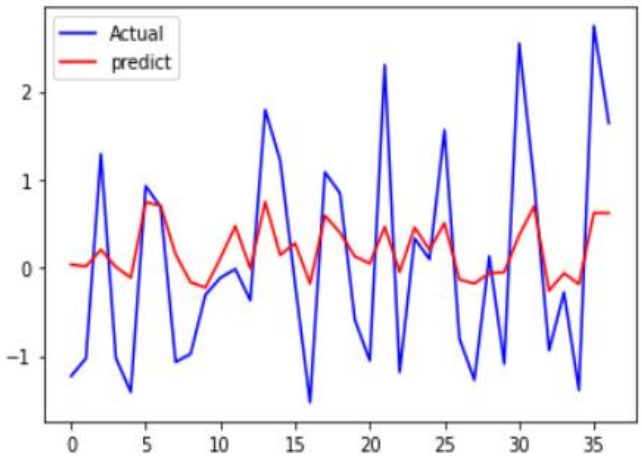


Figure 7.3.3. 1 Actual Vs Predicted Values in SVM Regression

The figure 7.3.3.1 shows the line plot of the actual values and predicted values using SVM regression. The actual values are plotted in blue colour and the predicted values in red colour. The predicted values are very smaller from the actual values in all points. The RMSE value of the model is 0.786 and the MAE is 0.617 which is comparatively good, and the r squared value is 0.452 which is reasonably a better value. This may be

because of the use of resampled data. The r squared measures how close the data are fitted to the regression line and for the data fitted closer to the regression line, r squared value will be higher and for lower r squared values, the data will be scattered around the regression line. In SVM algorithm, the r squared is comparatively high and hence most of the data is following around the regression line as shown in figure 7.3.3.2. Even though the points are plotted closer to the line, some of the predicted values are far beyond the actual values in SVM regression.

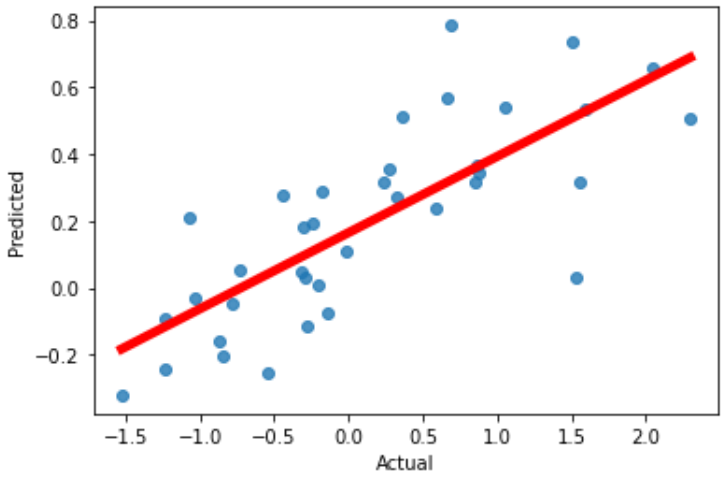


Figure 7.3.3. 2 Scatter plot of predicted values by SVM Regression

Support Vector Regression Model			
RMSE	**MSE**	**MAE**	**R Squared**
0.786	0.617	0.629	0.452

Table 7.3.3. 1 Performance metrices of Support Vector Regression

7.3.4 XG-Boost Regression Results

The figure 7.3.4.1 shows the zoomed portion of line plot of the fifty actual values and predicted values using XG-Boost. The actual values are plotted in blue colour and the predicted values in red colour. The predicted values are very closer to the actual values in all points for the XG-Boost. The RMSE value of the model is 653997 and the MAE is 134917which is higher values while the r squared value is 0.010158 which is comparatively lower. The r squared measures how close the data are fitted to the regression line and for the data fitted closer to the regression line, r squared value will be higher and for lower r squared values, the data will be scattered around the regression line. In XG-Boost algorithm, the r squared is comparatively low. Most of the actual data

and the predicted values are overlapping each other and the values are scattered around the regression line as shown in figure 7.3.3.2. Even though the points are scattered around the line, the prediction values are almost inline the actual values in XG-Boost.

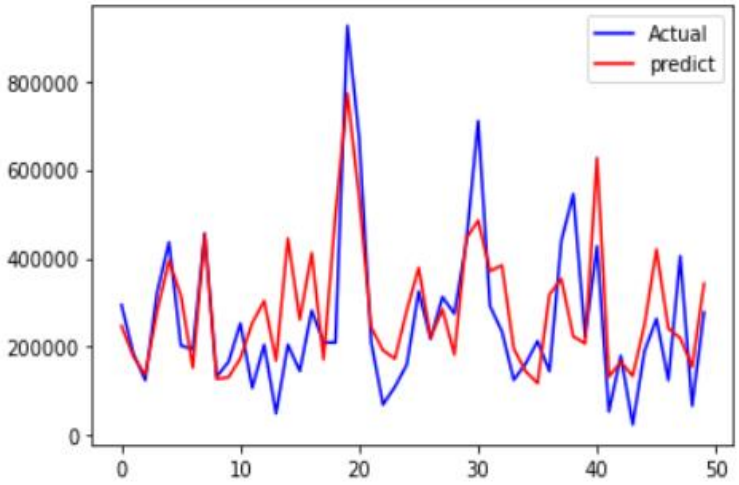


Figure 7.3.4. 1 Actual Vs Predicted Values in XG-Boost Regression

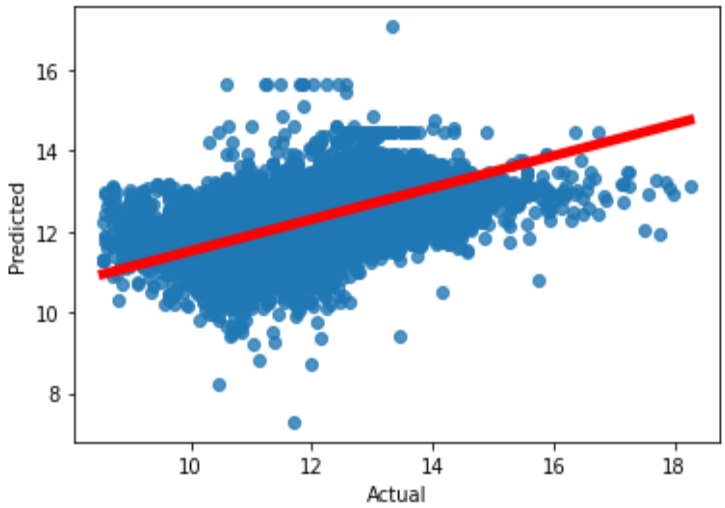


Figure 7.3.4. 2 Scatter plot of predicted values by XG-Boost Regression

XG-Boost Regression Model			
RMSE	**MSE**	**MAE**	**R Squared**
653997.82	427713148859.75	134917.819	0.010158

Table 7.3.4. 1 Performance metrices of XG-Boost Regression

7.3.5 Random Forest Regression Results

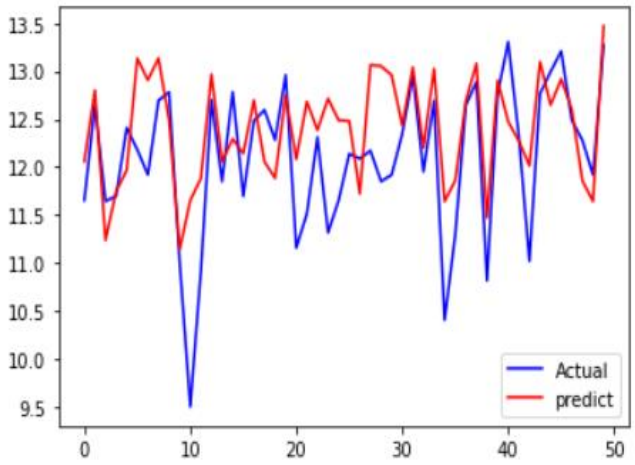


Figure 7.3.5. 1 Actual Vs Predicted Values in RF Regression

The figure 7.3.5.1 shows the line plot of the 50 actual values and predicted values using RF. The actual values are plotted in blue colour and the predicted values in red colour. Some of the predicted values are very closer to the actual values in the RF model. The RMSE value of the model is 1094282 and the MAE is 140099 which is comparatively higher values and the r squared value is also very low which is 0.004. The r squared measures how close the data are fitted to the regression line and for the data fitted closer to the regression line, r squared value will be higher and for lower r squared values, the data will be scattered around the regression line. In RF algorithm, the r squared is comparatively low. Most of the actual data and the predicted values are overlapping each other and the values are scattered around the regression line as shown in figure 7.3.5.2. Even though the points are scattered around the line, some of the predicted values are equal to the actual values in RF.

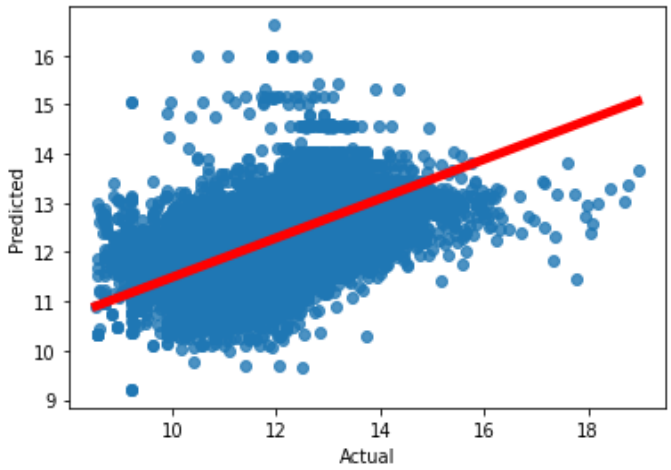


Figure 7.3.5. 2 Scatter plot of predicted values by RF Regression

Random Forest Regression Model			
RMSE	**MSE**	**MAE**	**R Squared**
1094282.69	1197454605589.61	140099.601	0.004

Table 7.3.5. 1 Performance metrices of Random Forest Regression

7.3.6 Neural Network Regression Results

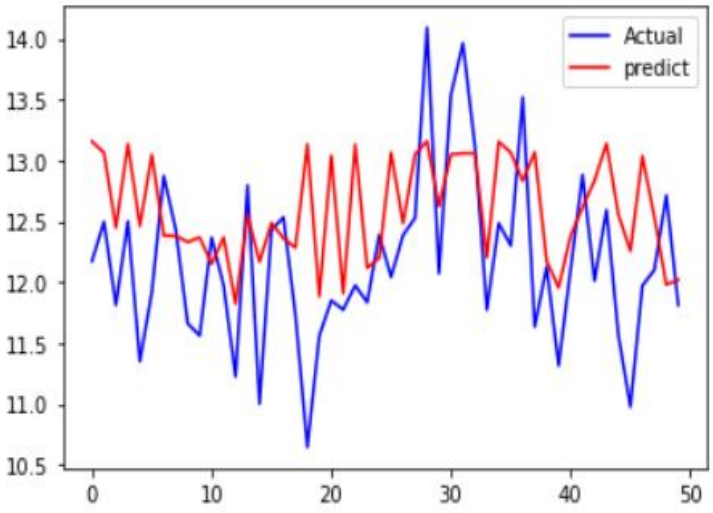


Figure 7.3.6. 1 Actual Vs Predicted Values in NN Regression

The figure 7.3.6.1 shows the line plot of some of the actual values and predicted values using RF. The actual values are plotted in blue colour and the predicted values in red colour. In the figure it shows that the predicted values are different from the actual values where some are higher, and some are lower than the actual values. The RMSE value of the model is 728057 and the MAE is 153320 which is comparatively higher values and the r squared value is also very low which is 0.02. The r squared measures how close the data are fitted to the regression line and for the data fitted closer to the regression line, r squared value will be higher and for lower r squared values, the data will be scattered around the regression line. In NN algorithm, the r squared is comparatively low. Most of the actual data and the predicted values are overlapping each other and the values are scattered around the regression line as shown in figure 7.3.6.2.

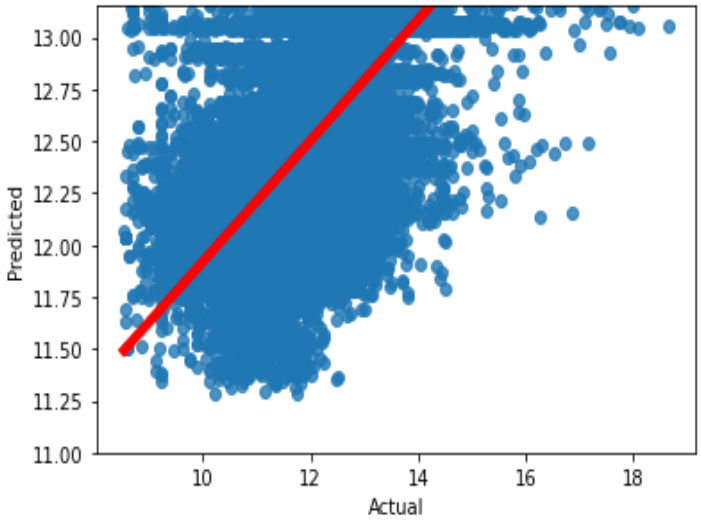


Figure 7.3.6. 2 Scatter plot of predicted values by NN Regression

NN Regression Model			
RMSE	**MSE**	**MAE**	**R Squared**
728057.406	530067586324.427	153320.551	0.022

Table 7.3.6. 1 Performance metrices of NN Regression

7.3.7 Selection of ML algorithm

The performance metrices of the different ML algorithms employed are analysed and after that the R squared, RMSE and, MAE were chosen as the criteria to decide the better model. This is because, the R squared can tell how well a model can predict the response variable in percentage and it captures the fraction of variance of actual values captured by the regression model. It also tends to give a better picture of the quality of the regression model. On the other hand, RMSE can be dealt with high error values and can tell the average deviation between the predicted house price made by the model and the actual house price. Compared to MSE, MAE is a more direct representation of sum of error terms. MSE gives larger penalization to big error by squaring it and is highly biased for larger values while MAE treats all errors the same**.** Since MAE is more robust with outliers it is also considered while evaluating the models.

Comparison of Regression Models				
Model	**RMSE**	**MSE**	**MAE**	**R Squared**
LR	803454.106	645538500662.903	135139.365	0.023
DCT	913160	833862387387.05	139920.77	0.01
SVM	0.786	0.617	0.629	0.452
XG-Boost	653997.82	427713148859.75	134917.819	0.010158
RF	1094282.69	1197454605589.61	140099.601	0.004
NN	728057.406	530067586324.427	153320.551	0.022

Table 7.3.7. 1 Comparison of performance metrices of regression models

By comparing the metrices, it is revealed that the XG-Boost model has the best RMSE and MAE values among the six algorithms performed but the R squared value is lower. NN is also showing comparatively good values of RMSE, MAE compared to the other algorithms and its R squared value is better than XG-Boost. Hence, NN regression and XG-Boost algorithms seems to be accurate for property price analysis. SVM regression has shown lower values for RMSE, MAE, and R squared, and this may be because of the resampling performed. Hence it can also be considered as a good algorithm for the property price analysis.

The actual and the predicted values of each model has been plotted to understand how well the model predicts apart from the results of the performance metrices. Only 50 data points were chosen from the data for plotting to understand the data more precisely. And it shows that the XG-Boost algorithm's prediction is better with the available data compared to other models. All the other models seem to be predicting some of the data when compared to XG-Boost.

K-Fold cross validation is applied on the best models by dividing the data into different groups. This is performed as the train and test data does not always have the same variation as the original data and hence it will affect the performance of the model when using the train test split method. In cross validation this problem is solved by dividing the data into multiple groups other than just two. K-fold cross validation is

utilized here, which will shuffle the data and divide the data into the desired number of folds. In this method one-fold will be used for testing and the other folds for training which will reduce the chances of underfitting as we are using most of the data for training, and significantly reduces overfitting as most of the data is also being used in validation set. K-fold cross validation on the SVM regression model produced a r-squared value of 0.4 in the fourth fold, which shows the model performance can be improved by this method. The same technique was applied to XG-Boost regression also and it produced 93100 as the average RMSE value for all the folds. This also implies that by K-fold cross validation the overall performance can be improved.

CHAPTER 8

ETHICAL CONSIDERATIONS

This section describes the ethical challenges faced in the different stages of the research as a data scientist and the ethical concerns with the data. To avoid the ethical issues, the data was collected from a website which is accessible to the public for download and use and no personal information were used for analysis. The ethical challenges faced are as follows.

8.1 Data storage, security, and responsible data stewardship

Even though the data does not contain any personal information, to restrict the loss of the data from the system and to avoid the misuse of it especially the API key, a folder is created in Google drive and one drive to save the details related to the dissertation. The folder is secured with passcode to avoid the misuse of it. Anti-virus packages are installed in machines to reduce the risk of hacking and virus.

8.2 Data hygiene and data relevance

To add additional data to the dataset, details are obtained from genuine websites with proper referencing. The API key to get the latitude is taken from a website which follows GDPR and sends requests as HTTPS.

8.3 Identifying and addressing ethically harmful data bias

The dataset has a column with postal code, and it contains the postal code of Dublin only. So, if I use that column in my modelling there may be a chance of bias towards Dublin than other places. Even though it doesn't affect any individuals and doesn't create any harms, but for the better modelling and to reduce bias, I may drop that column.

The ethical issues related to data are also considered in the research and they are,

8.4 Harms to Privacy and Security

To avoid the issues with privacy and security, no personal information is using in the research. The data is collected from an open website, and the data is free to public for downloading and using. Reuse and download of the data are permitted with

acknowledging the website and the data cannot be used in a misleading way. The data does not contain any personal information and hence there is no ethical harms related to privacy and security (Vallor et al.).

8.5 Harms to Fairness and Justice

Since no personal data is being used, there is no issue related to discrimination and bias related to gender or age and hence there is no issue related to sexism, racism, and ableism. Since the data is considering the whole Ireland and the properties sold and not about people, there is no other ethical issues related to fairness and justice also (Vallor et al.).

8.6 Harms to Transperancy and Autonomy

To avoid the transparency issues, the data is collected from the free website, and anyone can visit the data without any issues. The research is not using any humans for the analysis or any personal data and hence there is no issues related to transparency and autonomy and no human rights have been violated (Vallor et al.).

CHAPTER 9

CONCLUSION AND FUTUREWORK

Property prices in Ireland were analysed using different statistical methods, ML algorithms and visualizations to identify the factors effecting the property prices, to predict the property prices and to analyse the relationship between the different parameters. Different plots were created using seaborn and matplotlib for the entire dataset which is downloaded from the PSRA website and ANOVA test was generated for the random sample size of 500 samples from the entire data. MLR model was utilized to generate the hedonic pricing model to determine the influence of different factors. The results from the ANOVA test and MLR model implies that the attributes 'year', 'province', 'property description', 'latitude', 'longitude' and 'location' has significant effect on the property prices. From the visualizations it is revealed that the property prices are higher in Dublin compared to other counties and more houses are sold in Dublin also. The post codes of the county Dublin are also having significant effect on the property prices where Dublin 14 is having higher property prices and Dublin 10 is having lower prices. The SLR model implemented for the Dublin data confirmed this and revealed that the post codes are having a significant effect on the property prices. Different ML algorithms were used to identify the accurate model for the house price prediction and the XG-Boost regression and NN was found to the best models based on the RMSE and MAE values while XG-Boost model was found to be predicting the prices more precisely. The SVM regression performed on resample data was also found to be a better model for analysing the property prices based on its performance metrices.

As the data was limited and the access to other environmental and geographical data was limited, more accurate results and analysis could not be done. In future, if the data regarding the unemployment rate, geographical data such as flood rate of the area, population of the area in each year, and safety index of the area is available, the prediction could be done based on that data which makes the prediction more valuable and helpful to the customers. In addition to this the proposed model has high scalability which can be used to analyse the house prices of other countries with similar population and landscapes and properties.

APPENDIX A

In [ ]:
```
import pandas as pd
import os
import warnings
warnings.filterwarnings('ignore')
os.chdir("E:\Ginu_StudyMaterials\Sem2\Dissertation\Data")
from sklearn.metrics import mean_squared_error # for calculating the cost function
from sklearn.ensemble import RandomForestRegressor # for building the model
from sklearn.metrics import classification_report
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from opencage.geocoder import OpenCageGeocode
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from numpy import asarray
from xgboost import XGBRegressor
from numpy import absolute
from pandas import read_csv
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedKFold
from xgboost import XGBRegressor
from sklearn.metrics import accuracy_score
from sklearn import metrics
from sklearn.model_selection import GridSearchCV
from sklearn.neural_network import MLPRegressor
from sklearn.linear_model import Lasso
from sklearn.model_selection import KFold
```

In [ ]:
```
# reading the dataset
property_prices = pd.read_csv("PPR_ALL_v1.csv", na_values =("N/A", "NA", "--", " "),
```

In [ ]:
```
# renaming the columns
property_prices.rename({'Date of Sale (dd/mm/yyyy)':'date_of_sale', 'Address':'addres
```

In [ ]:
```
# reading the dataset to get the province list
town_list = pd.read_csv("ie_towns_sample.csv", na_values =("N/A", "NA", "--", " "))
```

In [ ]:
```
#taking the necessary columns only
province = town_list[['county', 'province']]
```

In [ ]:
```
# checking the unique province names
province['province'].unique()
```

In [ ]:
```
# dropping the duplicates
province_list = province.drop_duplicates(subset= ['county'], keep='first')
```

In [ ]:
```
# getting the api key to access the lattitude and longitude
key = '40d783cbf75143b48b8528d1804a3ccd'  # get api key from:  https://opencagedata.c

geocoder = OpenCageGeocode(key)
```

In [ ]:
```
list_lat = []   # create empty lists

list_long = []

for index, row in province_list.iterrows(): # iterate over rows in dataframe


    City = row['county']
    State = row['province']
    query = str(City)+','+str(State)
    #loc = row['temp_add']
    #query = str(loc)

    results = geocoder.geocode(query)
    lat = results[0]['geometry']['lat']
    long = results[0]['geometry']['lng']

    list_lat.append(lat)
    list_long.append(long)

# create new columns from lists

province_list['lat'] = list_lat

province_list['lon'] = list_long
```

In [ ]:
```
# dataframe with latitude and longitude
#province_list
```

In [ ]:
```
# merging the two data into one
df_merge_col = pd.merge(property_prices, province_list, on='county', how='left')
```

In [ ]:
```
# dropping the duplicates
df = df_merge_col.drop_duplicates()
```

In [ ]:
```
# converting the uppercase strings to title case
df['address'] = df['address'].str.title()
```

In [ ]:
```
df = df.assign(location=df["county"])
```

In [ ]:
```
# Split the location between Dublin and outside Dublin
df['location'] = df['location'].map({
 "Cork": "Outside", "Galway": "Outside", "Kildare": "Outside", "Meath": "Outside", "L
 "Wexford": "Outside", "Wicklow": "Outside", "Kerry": "Outside", "Donegal": "Outside"
 "Tipperary": "Outside", "Louth": "Outside", "Mayo": "Outside", "Clare": "Outside", "
 "Cavan": "Outside", "Sligo": "Outside", "Kilkenny": "Outside", "Laois": "Outside", "
 "Offaly": "Outside", "Carlow": "Outside", "Leitrim": "Outside", "Longford": "Outside
 "Dublin": "Dublin"})
```

In [ ]:
```
# converting strings in Irish to English
df['property_description'] = df['property_description'].replace(['Teach/Árasán Cónait
```

In [ ]:
```
# converting strings in Irish to English
df['property_size_description'] = df['property_size_description'].replace(['n?os l? r
```

In [ ]:
```
# converting strings in Irish to English
df['county'] = df['county'].replace(['Baile ?tha Cliath','Ní Bhaineann'],['Dublin',''
```

In [ ]:
```
# converting strings in Irish to English
df['property_description'] = df['property_description'].replace(['Teach/Árasán Cónait
```

In [ ]:
```
# converting strings in Irish to English
df['postal_code'] = df['postal_code'].replace(['Baile Átha Cliath 3','Baile Átha Clia
```

In [ ]:
```
# converting strings in Irish to English
df['property_description'] = df['property_description'].replace(['Second-Hand Dwellir
```

In [ ]:
```
# changing the date to pandas datetime format

df["date_of_sale"] = pd.to_datetime(df["date_of_sale"], format = '%d/%m/%Y')
```

In [ ]:
```
# adding columns month and year

df['year'] = df["date_of_sale"].dt.year
df['month'] = df["date_of_sale"].dt.month
```

In [ ]:
```
# verifying the datatypes of the data
df.info()
```

In [ ]:
```
#checking the null values
df.isna().sum()
```

In [ ]:
```
#saving the dataset to a new file for using it for dublin analysis.
df.to_csv("PRP_FOR_DUB.csv", index=False)
```

LR

In [ ]:
```
df1 = df.copy()
```

In [ ]:
```
df1.info()
```

In [ ]:
```
# plotting the correlation matrix to verify the correlation
plt.figure(figsize=(10,10))
mask=np.zeros_like(df1.corr(),dtype=np.bool)
mask[np.triu_indices_from(mask)]=True
sns.heatmap(data=df1.corr(),annot=True,square=True,mask=mask,cmap="RdBu_r",linewidths
plt.title("Correlation of Property prices data")
```

In [ ]:
```
# dropping the columns that are not co nsidering for the modelling.
df1.drop(columns = ['postal_code', 'property_size_description'], inplace=True)
```

In [ ]:
```
# checking the missing values
df1.isna().sum()
```

In [ ]:
```
# getting dummy values for the categorical variables
X1 = pd.get_dummies(df1[[ 'county', 'FMP', 'VAT_exclusive','property_description', 'p
```

In [ ]:
```
# merging two dataframes to get the final data for LR model
x3 = df1[['date_of_sale','price']]
x2=pd.concat([df1,X1], axis =1)
```

LR with scaling

In [ ]:
```
x3 = x2.copy()
x = x3.drop(columns = ['date_of_sale','address', 'price','county','FMP','VAT_exclusiv
y = x3[['price']]
x=x.values
y=y.values
```

In [ ]:
```
# Scale train and test sets with StandardScaler

data_train, data_test, target_train, target_test = train_test_split( x, y,test_size=0
X_train_std = StandardScaler().fit_transform(data_train)
X_test_std = StandardScaler().fit_transform(data_test)
reg = LinearRegression().fit(X_train_std, target_train)
reg.score(X_test_std, target_test)
```

In [ ]:
```
# Make predictions using the testing set
data_y_pred = reg.predict(X_test_std)
print(data_y_pred)
```

In [ ]:
```
# RMSE (Root Mean Square Error)
rmse = float(format(np.sqrt(mean_squared_error(target_test, data_y_pred)),'.3f'))
print("\nRMSE:\n",rmse)

# MSE (Mean Square Error)
mse = float(format((mean_squared_error(target_test, data_y_pred)),'.3f'))
print("\nMSE:\n",rmse)

# r squared
r = float(format((r2_score(target_test, data_y_pred)),'.3f'))
print("\nr squared:\n",rmse)

#Mean absolute error
mae = float(format((mean_absolute_error(target_test, data_y_pred)),'.3f'))
print("\nMAE:\n",rmse)
```

LR without scaling

In [ ]:
```
data_train, data_test, target_train, target_test = train_test_split( x, y,test_size=0
```

In [ ]:
```
# Fitting the model
reg = LinearRegression().fit(data_train, target_train)
reg.score(data_test, target_test)
```

In [ ]:
```
# Make predictions using the testing set
data_y_pred = reg.predict(data_test)
print(data_y_pred)
```

In [ ]:
```
# The coefficients
print('Coefficients: \n', reg.coef_)
```

In [ ]:
```
# RMSE (Root Mean Square Error)
rmse = float(format(np.sqrt(mean_squared_error(target_test, data_y_pred)),'.3f'))
print("\nRMSE:\n",rmse)

# MSE (Mean Square Error)
mse = float(format((mean_squared_error(target_test, data_y_pred)),'.3f'))
print("\nMSE:\n",rmse)

# r squared
r = float(format((r2_score(target_test, data_y_pred)),'.3f'))
print("\nr squared:\n",rmse)

#Mean absolute error
mae = float(format((mean_absolute_error(target_test, data_y_pred)),'.3f'))
print("\nMAE:\n",rmse)
```

In [ ]:
```
## LR Prediction
```

In [ ]:
```
y_pred = reg.predict(data_test)
```

In [ ]:
```
df_preds = pd.DataFrame({'Actual': target_test.squeeze(), 'Predicted': y_pred.squeeze
print(df_preds)
```

In [ ]:
```
# predicted values regression plot

sns.regplot(x=target_test,y=y_pred,ci=95,color ='blue', line_kws={"color": "red"});
plt.xlabel('y_test')
plt.ylabel('Predicted')
plt.show()
```

In [ ]:
```
sns.regplot(np.log(target_test),np.log(y_pred),truncate = True, line_kws={"color": "r
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()
```

In [ ]:
```
plt.plot(np.log(target_test),color='blue')
plt.plot(np.log(y_pred),color='red')
plt.show()
```

In [ ]:
```
# actual and predicted vales zoomed plot
plot_target = target_test[:50]
plot_ypred = y_pred[:50]

plt.plot(plot_target,color = 'blue',label='Actual')
plt.plot(plot_ypred,color = 'red',label='predict')
plt.legend()
plt.show()
```

DCT

In [ ]:
```
# x and y data
x4 = x2.copy()
X = x4.drop(columns = ['date_of_sale','address', 'price','county','FMP','VAT_exclusiv
y = x4['price']
```

In [ ]:
```
# train and test splitting
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state
```

In [ ]:
```
# fitting the model
model = DecisionTreeRegressor(random_state=44)
model.fit(X_train, y_train)
```

In [ ]:
```
predictions = model.predict(X_test)
print(predictions)
```

In [ ]:
```
# The coefficients
print('Coefficients: \n', reg.coef_)

rmse = float(format(np.sqrt(mean_squared_error(y_test, predictions)),'.3f'))
print("RMSE:",rmse)

# The mean squared error
print("Mean squared error: %.2f"
      % mean_squared_error(y_test, predictions))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % r2_score(y_test, predictions))

print("MAE : %.2f" % mean_absolute_error(y_test, predictions))
```

In [ ]:
```
df_preds_dc = pd.DataFrame({'Actual': y_test.squeeze(), 'Predicted': predictions.sque
print(df_preds_dc)
```

In [ ]:
```
plt.plot(np.log(y_test),color='blue')
plt.plot(np.log(predictions),color='red')
plt.show()
```

In [ ]:
```
# actual and predicted values plot
plot_target = y_test[:50]
plot_ypred = predictions[:50]

plt.plot(np.log(plot_target),color = 'blue',label='Actual')
plt.plot(np.log(plot_ypred),color = 'red',label='predict')
plt.legend()
plt.show()
```

In [ ]:
```
sns.regplot(x=y_test,y=predictions,ci=None,color ='blue');
plt.xlabel('y_test')
plt.ylabel('Predicted')
plt.show()
```

In [ ]:
```
# regression plot with predicted values
sns.regplot(np.log(y_test),np.log(predictions),truncate = True,  line_kws={"color": "
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()
```

In [ ]:
```
#from sklearn.tree import plot_tree
#import matplotlib.pyplot as plt
#plt.figure(figsize=(10,8), dpi=150)
#plot_tree(model, feature_names=X.columns);
```

In [ ]:
```
# printing the column names
#for col in x2.columns:
    #print(col)#
```

SVM

In [ ]:
```
## Full data is not working for SVM so resampled data and performed svm
```

In [ ]:
```
data = x2.copy()

df1 = data.set_index('date_of_sale')
```

In [ ]:
```
# downsampling the data
df11 = df1.resample('M').mean().reset_index()
```

In [ ]:
```
X_l = df11.drop(columns = ['date_of_sale', 'price'],axis=1)
y_p = df11[['price']]
```

In [ ]:
```
StdS_X = StandardScaler()
StdS_y = StandardScaler()
X_l = StdS_X.fit_transform(X_l)
y_p = StdS_y.fit_transform(y_p)
```

In [ ]:
```
# train test splitting
xtrain,xtest,ytrain,ytest=train_test_split(X_l,y_p)
```

In [ ]:
```
# import the model
# create the model object
#regressor = SVR(kernel = 'rbf')
# fit the model on the data
#regressor.fit(X_l, y_p)
```

In [ ]:
```
regressor=SVR(kernel='rbf',epsilon=1.0)
regressor.fit(xtrain,ytrain)
pred=regressor.predict(xtest)
#print(regressor.score(xtest,ytest))
#print(r2_score(ytest,pred))
```

In [ ]:
```
df_preds_svr = pd.DataFrame({'Actual': ytest.squeeze(), 'Predicted': pred.squeeze()})
```

In [ ]:
```
# zoomed version of actual vs predicted values plot
plot_target = ytest[:50]
plot_ypred = pred[:50]

plt.plot((plot_target),color = 'blue',label='Actual')
plt.plot((plot_ypred),color = 'red',label='predict')
plt.legend()
plt.show()
```

In [ ]:
```
sns.regplot((ytest),(pred),ci=None,  line_kws={"color": "red","alpha":1,"lw":5})
plt.xlabel('Actual')
```

```
plt.ylabel('Predicted')
plt.show()
```

In [ ]:
```
# RMSE (Root Mean Square Error)
rmse = float(format(np.sqrt(mean_squared_error(ytest, pred)),'.3f'))
print("\nRMSE:\n",rmse)

# MSE (Mean Square Error)
rmse = float(format((mean_squared_error(ytest, pred)),'.3f'))
print("\nMSE:\n",rmse)

# r squared
rmse = float(format((r2_score(ytest, pred)),'.3f'))
print("\nr squared:\n",rmse)

#Mean absolute error
rmse = float(format((mean_absolute_error(ytest, pred)),'.3f'))
print("\nMAE:\n",rmse)
```

In [ ]:
```
# k-fold CV (using all the 13 variables)
#lm = LinearRegression()
scores = cross_val_score(regressor, xtrain, ytrain, scoring='r2', cv=5)
scores
```

In [ ]:
```
# create a KFold object with 5 splits
folds = KFold(n_splits = 5, shuffle = True, random_state = 100)
scores = cross_val_score(regressor, xtrain, ytrain, scoring='r2', cv=folds)
scores
```

XGBoost

In [ ]:
```
d1 = x2.copy()
X = d1.drop(columns = ['date_of_sale','address', 'price','county','FMP','VAT_exclusiv
y = d1[['price']]
X=X.values
y=y.values
```

In [ ]:
```
d2 = x2.copy()
X = d2.drop(columns = ['date_of_sale','address', 'price','county','FMP','VAT_exclusiv
Y = d2[['price']]
X=X.values
Y=Y.values
```

In [ ]:
```
seed = 7
test_size = 0.2
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=test_size, random
```

In [ ]:
```
model = XGBRegressor()
model.fit(X_train, y_train)
```

In [ ]:
```
# make predictions for test data
y_pred = model.predict(X_test)
predictions = [round(value) for value in y_pred]
# evaluate predictions
accuracy = accuracy_score(y_test, predictions)
print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

In [ ]:
```
print("r2 score : ",metrics.r2_score(y_test, y_pred))

print("MSE : ",metrics.mean_squared_error(y_test, y_pred))

print("MAE: ",metrics.mean_absolute_error(y_test, y_pred))

rmse = float(format(np.sqrt(mean_squared_error(y_test, y_pred))))
print("RMSE:",rmse)
```

In [ ]:
```
# actual vs predicted values plot
plot_target = y_test[:50]
plot_ypred = y_pred[:50]

plt.plot((plot_target),color = 'blue',label='Actual')
plt.plot((plot_ypred),color = 'red',label='predict')
plt.legend()
plt.show()
```

In [ ]:
```
# regression plot with predicted values
sns.regplot(np.log(y_test),np.log(y_pred),truncate=True,  line_kws={"color": "red","a
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()
```

In [ ]:
```
# cross validation
# create a KFold object with 5 splits
folds = KFold(n_splits = 5, shuffle = True, random_state = 100)
scores = cross_val_score(model, X_train, y_train, scoring='r2', cv=folds)
scores
```

In [ ]:
```
def rmse(score):
    rmse = np.sqrt(-score)
    print(f'rmse= {"{:.2f}".format(rmse)}')
```

In [ ]:
```
folds = KFold(n_splits = 10, shuffle = True, random_state = 100)
score = cross_val_score(model, X_train, y_train, scoring='neg_mean_squared_error', cv
#scores


print(f'Scores for each fold: {score}')
```

In [ ]:
```
rmse(score.mean())
```

In [ ]:
```
rmse = np.sqrt(-score)
print("Average RMSE: {}".format(np.mean(rmse)))
```

RF

In [ ]:
```
d3 = x2.copy()
X = d3.drop(columns = ['date_of_sale','address', 'price','county','FMP','VAT_exclusiv
y = d3['price']
X=X.values
y=y.values
```

In [ ]:
```
# Splitting the dataset into training and testing set (80/20)
```

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_sta
```

In [ ]:
```
# Initializing the Random Forest Regression model with 10 decision trees
model = RandomForestRegressor(n_estimators = 10, random_state = 0)
```

In [ ]:
```
# Fitting the Random Forest Regression model to the data
model.fit(x_train, y_train)
```

In [ ]:
```
# Predicting the target values of the test set
y_pred = model.predict(x_test)
```

In [ ]:
```
# RMSE (Root Mean Square Error)
rmse = float(format(np.sqrt(mean_squared_error(y_test, y_pred)),'.3f'))
print("\nRMSE:\n",rmse)

# MSE (Mean Square Error)
rmse = float(format((mean_squared_error(y_test, y_pred)),'.3f'))
print("\nMSE:\n",rmse)

# r squared
rmse = float(format((r2_score(y_test, y_pred)),'.3f'))
print("\nr squared:\n",rmse)

#Mean absolute error
rmse = float(format((mean_absolute_error(y_test, y_pred)),'.3f'))
print("\nMAE:\n",rmse)
```

In [ ]:
```
plot_target = y_test[:50]
plot_ypred = y_pred[:50]

plt.plot(np.log(plot_target),color = 'blue',label='Actual')
plt.plot(np.log(plot_ypred),color = 'red',label='predict')
plt.legend()
plt.show()
```

In [ ]:
```
sns.regplot(np.log(y_test),np.log(y_pred),truncate=True,  line_kws={"color": "red","a
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()
```

NN

In [ ]:
```
d6 = x2.copy()
X = d6.drop(columns = ['date_of_sale','address', 'price','county','FMP','VAT_exclusiv
y = d6['price']
X=X.values
y=y.values
```

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y,random_state=1, test_size=0.
```

In [ ]:
```
print(X_train.shape); print(X_test.shape)
```

In [ ]:
```
reg = MLPRegressor(hidden_layer_sizes=(43,43,43),activation="relu" ,random_state=1, m
```

In [ ]:
```
y_pred=reg.predict(X_test)
```

In [ ]:
```
# RMSE (Root Mean Square Error)
rmse = float(format(np.sqrt(mean_squared_error(y_test, y_pred)),'.3f'))
print("\nRMSE:\n",rmse)

# MSE (Mean Square Error)
rmse = float(format((mean_squared_error(y_test, y_pred)),'.3f'))
print("\nMSE:\n",rmse)

# r squared
rmse = float(format((r2_score(y_test, y_pred)),'.3f'))
print("\nr squared:\n",rmse)

#Mean absolute error
rmse = float(format((mean_absolute_error(y_test, y_pred)),'.3f'))
print("\nMAE:\n",rmse)
```

In [ ]:
```
plot_target = y_test[:50]
plot_ypred = y_pred[:50]

plt.plot(np.log(plot_target),color = 'blue',label='Actual')
plt.plot(np.log(plot_ypred),color = 'red',label='predict')
plt.legend()
plt.show()
```

In [ ]:
```
sns.regplot(np.log(y_test),np.log(y_pred),truncate=True,  line_kws={"color": "red","a
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()
```

In [ ]:
```

```

APPENDIX B

In [ ]:
```
import pandas as pd
import os
import numpy as np
import plotly as py
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
from plotly.offline import init_notebook_mode
import statsmodels.api as sm
from statsmodels.formula.api import ols
import statistics
import scipy.stats as stats
from scipy.stats import kendalltau
from scipy.stats import spearmanr
from scipy.stats import pearsonr
from statsmodels.graphics.regressionplots import plot_partregress_grid
os.chdir("E:\Ginu_StudyMaterials\Sem2\Dissertation\Data")
```

In [ ]:
```
data = pd.read_csv("PRP_FOR_DUB.csv", na_values =("N/A", "NA", "--", " "), encoding =
data
```

In [ ]:
```
values=["Dublin"]
dub_data = data[data["location"].isin(values)]
dub_data
```

In [ ]:
```
dub_data.to_csv("PRP_Dublin.csv", index=False)
```

Statistical Analyses

In [ ]:
```
#### MLR
```

In [ ]:
```
for col in dub_data.columns:
    print(col)
```

In [ ]:
```
counties = dub_data['county'].unique()
counties
```

In [ ]:
```
rppr1 = dub_data.copy()
rppr1.drop(columns =['date_of_sale','address','VAT_exclusive','FMP','county','locatio
```

In [ ]:
```
#rppr1["location_Dublin"]=pd.get_dummies(rppr1["location"])["Dublin"]
```

In [ ]:
```
rppr1["property_new"]=pd.get_dummies(rppr1["property_description"])["NewHouse"]
```

In [ ]:
```
rppr1["postal_code_Dublin 14"]=pd.get_dummies(rppr1["postal_code"])["Dublin 14"]
rppr1["postal_code_Dublin 2"]=pd.get_dummies(rppr1["postal_code"])["Dublin 2"]
rppr1["postal_code_Dublin 13"]=pd.get_dummies(rppr1["postal_code"])["Dublin 13"]
rppr1["postal_code_Dublin 12"]=pd.get_dummies(rppr1["postal_code"])["Dublin 12"]
rppr1["postal_code_Dublin 4"]=pd.get_dummies(rppr1["postal_code"])["Dublin 4"]
rppr1["postal_code_Dublin 11"]=pd.get_dummies(rppr1["postal_code"])["Dublin 11"]
```

```
rppr1["postal_code_Dublin 9"]=pd.get_dummies(rppr1["postal_code"])["Dublin 9"]
rppr1["postal_code_Dublin 10"]=pd.get_dummies(rppr1["postal_code"])["Dublin 10"]
rppr1["postal_code_Dublin 15"]=pd.get_dummies(rppr1["postal_code"])["Dublin 15"]
rppr1["postal_code_Dublin 22"]=pd.get_dummies(rppr1["postal_code"])["Dublin 22"]
rppr1["postal_code_Dublin 5"]=pd.get_dummies(rppr1["postal_code"])["Dublin 5"]
rppr1["postal_code_Dublin 18"]=pd.get_dummies(rppr1["postal_code"])["Dublin 18"]
rppr1["postal_code_Dublin 6"]=pd.get_dummies(rppr1["postal_code"])["Dublin 6"]
rppr1["postal_code_Dublin 6w"]=pd.get_dummies(rppr1["postal_code"])["Dublin 6w"]
rppr1["postal_code_Dublin 17"]=pd.get_dummies(rppr1["postal_code"])["Dublin 17"]
rppr1["postal_code_Dublin 16"]=pd.get_dummies(rppr1["postal_code"])["Dublin 16"]
rppr1["postal_code_Dublin 8"]=pd.get_dummies(rppr1["postal_code"])["Dublin 8"]
rppr1["postal_code_Dublin 3"]=pd.get_dummies(rppr1["postal_code"])["Dublin 3"]
rppr1["postal_code_Dublin 1"]=pd.get_dummies(rppr1["postal_code"])["Dublin 1"]
rppr1["postal_code_Dublin 17"]=pd.get_dummies(rppr1["postal_code"])["Dublin 17"]
rppr1["postal_code_Dublin 20"]=pd.get_dummies(rppr1["postal_code"])["Dublin 20"]
```

In [ ]:
```
from numpy import sqrt
log_price = np.log(rppr1['price'])
transform  = sqrt(log_price)
```

In [ ]:
```
X = rppr1[["property_new","year","lat","lon","postal_code_Dublin 14","postal_code_Dub
X = sm.add_constant(X)
y = transform
#X.head(20)
```

In [ ]:
```
model_full_mlr = sm.OLS(y, X).fit()
```

In [ ]:
```
#fitted values
model_fitted_vals = model_full_mlr.fittedvalues
#model residuals
model_residuals = model_full_mlr.resid
#standardised residuals
model_norm_residuals = model_full_mlr.get_influence().resid_studentized_internal
```

In [ ]:
```
sns.regplot(x=model_fitted_vals,y=model_residuals,
 ci=False,lowess=True,
 line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.show()
```

In [ ]:
```
stats.probplot(model_norm_residuals, plot=sns.mpl.pyplot)
plt.show()

plt.hist(model_norm_residuals)
plt.show()
```

In [ ]:
```
from statsmodels.formula.api import ols
model_full_mlr1 = ols('log_price ~ lat+lon+C(year)+C(postal_code)+C(property_descript
model_full_mlr1.summary()
```

SLR

In [ ]:
```
rppr1 = dub_data.copy()
log_price = np.log(rppr1['price'])
transform  = sqrt(log_price)
```

In [ ]:
```
rppr1["postal_code_Dublin 14"]=pd.get_dummies(rppr1["postal_code"])["Dublin 14"]
rppr1["postal_code_Dublin 2"]=pd.get_dummies(rppr1["postal_code"])["Dublin 2"]
rppr1["postal_code_Dublin 13"]=pd.get_dummies(rppr1["postal_code"])["Dublin 13"]
rppr1["postal_code_Dublin 12"]=pd.get_dummies(rppr1["postal_code"])["Dublin 12"]
rppr1["postal_code_Dublin 4"]=pd.get_dummies(rppr1["postal_code"])["Dublin 4"]
rppr1["postal_code_Dublin 11"]=pd.get_dummies(rppr1["postal_code"])["Dublin 11"]
rppr1["postal_code_Dublin 9"]=pd.get_dummies(rppr1["postal_code"])["Dublin 9"]
rppr1["postal_code_Dublin 10"]=pd.get_dummies(rppr1["postal_code"])["Dublin 10"]
rppr1["postal_code_Dublin 15"]=pd.get_dummies(rppr1["postal_code"])["Dublin 15"]
rppr1["postal_code_Dublin 22"]=pd.get_dummies(rppr1["postal_code"])["Dublin 22"]
rppr1["postal_code_Dublin 5"]=pd.get_dummies(rppr1["postal_code"])["Dublin 5"]
rppr1["postal_code_Dublin 18"]=pd.get_dummies(rppr1["postal_code"])["Dublin 18"]
rppr1["postal_code_Dublin 6"]=pd.get_dummies(rppr1["postal_code"])["Dublin 6"]
rppr1["postal_code_Dublin 6w"]=pd.get_dummies(rppr1["postal_code"])["Dublin 6w"]
rppr1["postal_code_Dublin 17"]=pd.get_dummies(rppr1["postal_code"])["Dublin 17"]
rppr1["postal_code_Dublin 16"]=pd.get_dummies(rppr1["postal_code"])["Dublin 16"]
rppr1["postal_code_Dublin 8"]=pd.get_dummies(rppr1["postal_code"])["Dublin 8"]
rppr1["postal_code_Dublin 3"]=pd.get_dummies(rppr1["postal_code"])["Dublin 3"]
rppr1["postal_code_Dublin 1"]=pd.get_dummies(rppr1["postal_code"])["Dublin 1"]
rppr1["postal_code_Dublin 17"]=pd.get_dummies(rppr1["postal_code"])["Dublin 17"]
rppr1["postal_code_Dublin 20"]=pd.get_dummies(rppr1["postal_code"])["Dublin 20"]
```

In [ ]:
```
# Try SLR
import numpy as np
X = rppr1[["postal_code_Dublin 14","postal_code_Dublin 2","postal_code_Dublin 13","pc
X = sm.add_constant(X)
y = transform
model_slr = sm.OLS(y, X).fit()
```

In [ ]:
```
#fitted values
model_fitted_vals = model_slr.fittedvalues
#model residuals
model_residuals = model_slr.resid
#standardised residuals
model_norm_residuals = model_slr.get_influence().resid_studentized_internal
```

In [ ]:
```
sns.regplot(x=model_fitted_vals,y=model_residuals,
ci=False,lowess=True,
line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.show()
```

In [ ]:
```
stats.probplot(model_norm_residuals, plot=sns.mpl.pyplot)
plt.show()

plt.hist(model_norm_residuals)
plt.show()
```

In [ ]:
```
from statsmodels.formula.api import ols
model_full_mlr1 = ols('log_price ~ C(postal_code)', data=rppr1).fit()
model_full_mlr1.summary()
```

APPENDIX C

In [ ]:
```
import pandas as pd
import numpy as np
import plotly.express as px
import geopandas as gpd
from geopandas import GeoDataFrame
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:
```
import geopandas as gpd
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.axes_grid1 import make_axes_locatable
import plotly.graph_objects as go
```

In [ ]:
```
data = pd.read_csv('E://Ginu_StudyMaterials//Sem2//Dissertation//MapData//PRP.csv')
```

In [ ]:
```
data
```

In [ ]:
```
counties = data['county'].unique()
counties
```

In [ ]:
```
median_per_county = [data['price'][data['county']==county].median() for county in counties]

median_per_county = np.asarray(median_per_county)
median_per_county
```

In [ ]:
```
q=[]
for price in median_per_county:
    x = price/10**3
    q.append(x)
```

In [ ]:
```
county_price = pd.DataFrame(q)
```

In [ ]:
```
county_price['county'] = counties
```

In [ ]:
```
county_price.rename(columns={0:'price'},inplace=True)
```

In [ ]:
```
from opencage.geocoder import OpenCageGeocode
key = '40d783cbf75143b48b8528d1804a3ccd'  # get api key from:  https://opencagedata.com

geocoder = OpenCageGeocode(key)
```

In [ ]:
```
list_lat = []   # create empty lists

list_long = []

for index, row in county_price.iterrows(): # iterate over rows in dataframe


    City = row['county']
    #State = row['province']
    query = str(City)
    #loc = row['temp_add']
    #query = str(loc)

    results = geocoder.geocode(query)
    lat = results[0]['geometry']['lat']
    long = results[0]['geometry']['lng']

    list_lat.append(lat)
    list_long.append(long)

# create new columns from lists

county_price['lat'] = list_lat

county_price['lon'] = list_long
```

In [ ]:
```
county_price
```

In [ ]:
```
data1 = gpd.read_file('E://Ginu_StudyMaterials//Sem2//Dissertation//MapData//IRL_adm1.shp')
#pd.read_csv('E://Ginu_StudyMaterials//Sem2//Dissertation//MapData//IRL_roads.shp', encoding = 'unicode_escape')
```

In [ ]:
```
data1
```

In [ ]:
```
data1.rename({'NAME_1':'county' }, axis=1, inplace=True)
```

In [ ]:
```
df_merge_col = pd.merge(county_price, data1, on='county', how='left')

df_merge_col
```

In [ ]:
```
df_merge_col =df_merge_col[['price', 'county' , 'geometry', 'ID_1']]
```

In [ ]:
```
df_merge_col = df_merge_col.dropna(subset=['price', 'geometry']).set_index('ID_1')
df_merge_col
```

In [ ]:
```
from geopandas import GeoDataFrame

df_merge_col = GeoDataFrame(df_merge_col)
```

In [ ]:
```
# OPTIONAL: Display using geopandas
fig, ax = plt.subplots(1,1, figsize=(20,20))
divider = make_axes_locatable(ax)
tmp = df_merge_col.copy()
#tmp['price'] = tmp['price']*100 #To display percentages
cax = divider.append_axes("right", size="3%", pad=-1) #resize the colorbar
tmp.plot(column='price', ax=ax,cax=cax,  legend=True,
         legend_kwds={'label': "Median House price in "})

tmp.geometry.boundary.plot(color='#BABABA', ax=ax, linewidth=0.3) #Add some borders to the geometries
ax.axis('off')
```

In [ ]:
```
# OPTIONAL: Display using geopandas

tmp = df_merge_col.copy()
tmp['coords'] = tmp['geometry'].apply(lambda x: x.representative_point().coords[:])
tmp['coords']= [coords[0] for coords in tmp['coords']]

fig, ax = plt.subplots(1,1, figsize=(10,10))
#divider = make_axes_locatable(ax)

#tmp['price'] = tmp['price']*100 #To display percentages
#cax = divider.append_axes("right", size="3%", pad=-1) #resize the colorbar
tmp.plot(column='price', ax=ax,  legend=True, colormap = 'RdYlGn_r',
         legend_kwds={'label': "Median House price in €k"})

for idx, row in tmp.iterrows():
   plt.annotate(text=row['county'], xy=row['coords'], horizontalalignment='center', color='black')

tmp.geometry.boundary.plot(color='#BABABA', ax=ax, linewidth=0.3) #Add some borders to the geometries
ax.axis('off')
```

In [ ]:
```
import adjustText as aT
```

In [ ]:
```
tmp1 = df_merge_col.copy()

tmp1["rep"] = tmp1["geometry"].representative_point()
za_points = tmp1.copy()
za_points.set_geometry("rep", inplace = True)
```

In [ ]:
```
ax = tmp1.plot(figsize = (15, 12), color = "whitesmoke", edgecolor = "lightgrey", linewidth = 0.5)
texts = []

for x, y, label in zip(za_points.geometry.x, za_points.geometry.y, za_points["county"]):
    texts.append(plt.text(x, y, label, fontsize = 8))

aT.adjust_text(texts, force_points=0.3, force_text=0.8, expand_points=(1,1), expand_text=(1,1),
               arrowprops=dict(arrowstyle="-", color='grey', lw=0.5))
```

In [ ]:
```
data_dub = pd.read_csv('E://Ginu_StudyMaterials//Sem2//Dissertation//Data//PRP_Dublin.csv')
```

In [ ]:
```
codes = data['postal_code'].unique()
codes
```

In [ ]:
```
median_per_county = [data['price'][data['postal_code']==postal_code].median() for postal_code in codes]

median_per_county = np.asarray(median_per_county)
median_per_county
```

In [ ]:
```
q=[]
for price in median_per_county:
    x = price/10**3
    q.append(x)
```

In [ ]:
```
county_price = pd.DataFrame(q)
```

In [ ]:
```
county_price['postal_code'] = codes
```

In [ ]:
```
county_price['county'] = "Dublin"
```

In [ ]:
```
county_price.rename(columns={0:'price'},inplace=True)
```

In [ ]:
```
county_price.dropna()
```

In [ ]:
```
county_price['postal_code'] = county_price['postal_code'].replace(['Dublin 6w'],['Dublin 6W'])
```

In [ ]:
```
data1 = gpd.read_file('E://Ginu_StudyMaterials//Sem2//Dissertation//MapData//dublin_postcodes//Postcode_dissolve.
```

In [ ]:
```
data1
```

In [ ]:
```
data1.rename({'Yelp_postc':'postal_code' }, axis=1, inplace=True)
```

In [ ]:
```
df_merge = pd.merge(county_price, data1, on='postal_code', how='left')

df_merged = df_merge.copy()
```

In [ ]:
```
df_merged =df_merged[['price', 'county' ,'postal_code', 'geometry']]
```

In [ ]:
```
df_merged.dropna()
```

In [ ]:
```
#df_merged = df_merged.dropna(subset=['price', 'geometry']).set_index()
#df_merged
```

In [ ]:
```
from geopandas import GeoDataFrame

df_merged = GeoDataFrame(df_merged)
```

In [ ]:
```
df_merged1 = df_merged.dropna()
```

In [ ]:
```
# OPTIONAL: Display using geopandas
fig, ax = plt.subplots(1,1, figsize=(10,10))
divider = make_axes_locatable(ax)
tmp = df_merged.copy()
#tmp['price'] = tmp['price']*100 #To display percentages
cax = divider.append_axes("right", size="3%", pad=-1) #resize the colorbar
tmp.plot(column='price', ax=ax,cax=cax,  legend=True,
         legend_kwds={'label': "Median House price in "})

tmp.geometry.boundary.plot(color='#BABABA', ax=ax, linewidth=0.3) #Add some borders to the geometries
ax.axis('off')
```

In [ ]:
```
# OPTIONAL: Display using geopandas
```

```
from matplotlib import cm
#from colorspacious import cspace_converter


tmp = df_merged1.copy()
#tmp["geometry"] = tmp["geometry"].centroid
tmp['coords'] = tmp['geometry'].apply(lambda x: x.representative_point().coords[:])
tmp['coords']= [coords[0] for coords in tmp['coords']]

fig, ax = plt.subplots(1,1, figsize=(10,10))
#divider = make_axes_locatable(ax)

#tmp['price'] = tmp['price']*100 #To display percentages
#cax = divider.append_axes("right", size="3%", pad=-1) #resize the colorbar
tmp.plot(column='price', ax=ax,  legend=True, colormap = 'RdYlGn_r',
         legend_kwds={'label': "Median House price in €k"})

for idx, row in tmp.iterrows():
   plt.annotate(text=row['postal_code'], xy=row['coords'], horizontalalignment='center', color='black')

tmp.geometry.boundary.plot(color='#BABABA', ax=ax, linewidth=0.3) #Add some borders to the geometries
ax.axis('off')
```

In [ ]:

APPENDIX D

In [ ]:
```
import pandas as pd
import os
import numpy as np
import plotly as py
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
from plotly.offline import init_notebook_mode
import statsmodels.api as sm
from statsmodels.formula.api import ols
import statistics
import scipy.stats as stats
from scipy.stats import kendalltau
from scipy.stats import spearmanr
from scipy.stats import pearsonr
from statsmodels.graphics.regressionplots import plot_partregress_grid
os.chdir("E:\Ginu_StudyMaterials\Sem2\Dissertation\Data")
```

In [ ]:
```
property_prices = pd.read_csv("PPR_ALL_v1.csv", na_values =("N/A", "NA", "--", " "), encoding = 'unicode_escape')
property_prices
```

In [ ]:
```
property_prices.rename({'Date of Sale (dd/mm/yyyy)':'date_of_sale', 'Address':'address', 'Postal Code':'postal_co
```

In [ ]:
```
town_list = pd.read_csv("ie_towns_sample.csv", na_values =("N/A", "NA", "--", " "))
town_list
```

In [ ]:
```
province_list = town_list[['county', 'province']]
```

In [ ]:
```
province_list['province'].unique()
```

In [ ]:
```
df1 = province_list.drop_duplicates(subset= ['county'], keep='first')
df1
```

In [ ]:
```
df1['county'].unique()
```

In [ ]:
```
from opencage.geocoder import OpenCageGeocode
key = '40d783cbf75143b48b8528d1804a3ccd'  # get api key from:  https://opencagedata.com

geocoder = OpenCageGeocode(key)
```

In [ ]:
```
list_lat = []   # create empty lists

list_long = []

for index, row in df1.iterrows(): # iterate over rows in dataframe


    City = row['county']
    State = row['province']
    query = str(City)+','+str(State)
    #loc = row['temp_add']
    #query = str(loc)

    results = geocoder.geocode(query)
    lat = results[0]['geometry']['lat']
    long = results[0]['geometry']['lng']

    list_lat.append(lat)
    list_long.append(long)

# create new columns from lists

df1['lat'] = list_lat

df1['lon'] = list_long
```

In [ ]:
```
df_merge_col = pd.merge(property_prices, df1, on='county', how='left')

df_merge_col
```

In [ ]:
```
df_merge_col.drop_duplicates()
```

In [ ]:
```
df_merge_col['property_description'] = df_merge_col['property_description'].replace(['Teach/Árasán Cónaithe Athái
```

In [ ]:
```
df_merge_col['property_size_description'] = df_merge_col['property_size_description'].replace(['n?os l? n? 38 m?a
```

In [ ]:
```
df_merge_col['county'] = df_merge_col['county'].replace(['Baile ?tha Cliath','Ni Bhaineann'],['Dublin',''])
```

In [ ]:
```
df_merge_col['postal_code'] = df_merge_col['postal_code'].replace(['Baile Átha Cliath 3','Baile Átha Cliath 4','B
```

In [ ]:
```
df_merge_col['property_description'] = df_merge_col['property_description'].replace(['Second-Hand Dwelling house
```

In [ ]:
```
df_merge_col['address'] = df_merge_col['address'].str.title()
```

In [ ]:
```
#df_merge_col['price'] =df_merge_col['price'].str.replace('£','')
#df_merge_col['price'] = df_merge_col['price'].str.replace(',', '')
```

In [ ]:
```
# changing the date to pandas datetime format

pd.to_datetime(df_merge_col['date_of_sale'].astype(str), format ='%d/%m/%Y')
```

In [ ]:
```
cd_date1 = pd.to_datetime(df_merge_col['date_of_sale'].astype(str), format ='%d/%m/%Y')
#pd.to_datetime(df_merge_col['date_of_sale'].astype(str), format ='%d/%m/%Y')
df_merge_col['month_year'] = pd.to_datetime(df_merge_col['date_of_sale']).dt.to_period('M')

df_merge_col['year'] = cd_date1.dt.year
df_merge_col['month'] = cd_date1.dt.month
```

In [ ]:
```
df_merge_col = df_merge_col.assign(location=df_merge_col["county"])
```

In [ ]:
```
# Split the location between Dublin and outside Dublin
df_merge_col['location'] = df_merge_col['location'].map({
 "Cork": "Outside", "Galway": "Outside", "Kildare": "Outside", "Meath": "Outside", "Limerick": "Outside",
 "Wexford": "Outside", "Wicklow": "Outside", "Kerry": "Outside", "Donegal": "Outside", "Waterford": "Outside",
 "Tipperary": "Outside", "Louth": "Outside", "Mayo": "Outside", "Clare": "Outside", "Westmeath": "Outside",
 "Cavan": "Outside", "Sligo": "Outside", "Kilkenny": "Outside", "Laois": "Outside", "Roscommon": "Outside",
 "Offaly": "Outside", "Carlow": "Outside", "Leitrim": "Outside", "Longford": "Outside", "Monaghan": "Outside",
 "Dublin": "Dublin"})
df_merge_col.head()
```

In [ ]:
```
df_merge_col.info()
```

In [ ]:
```
# changing the datatype to int
df_merge_col['price'].astype('int64')
```

In [ ]:
```
# getting the maximum house prices
for county in df_merge_col['county'].unique():
    print('County {}; max house price €{:.0f}m'.format(county, (df_merge_col[df_merge_col['county'] == county]['p
```

In [ ]:
```
county = df_merge_col['county'].unique()
```

In [ ]:
```
max_house_prices = [(df_merge_col[df_merge_col['county'] == county]['price'].max())/10**6 for county in df_merge_
```

In [ ]:
```
max_house_prices
```

In [ ]:
```
county_price = pd.DataFrame(county)
county_price['max_house_prices'] = max_house_prices
county_price.rename(columns={0:'county'},inplace=True)
county_price
```

In [ ]:
```
fig, ax = plt.subplots(figsize=(25, 10))
plt.ylabel('Max Residential Property Price (€m)')
sns.barplot(x= county_price['county'], y=  max_house_prices,order=county_price.sort_values('max_house_prices', as
plt.show()
```

In [ ]:
```
# getting the minimum house prices
min_house_prices = [(df_merge_col[df_merge_col['county'] == county]['price'].min()) for county in df_merge_col['c
```

In [ ]:
```
county_price_min = pd.DataFrame(county)
county_price_min['min_house_prices'] = min_house_prices
county_price_min.rename(columns={0:'county'},inplace=True)
county_price_min
```

In [ ]:
```
fig, ax = plt.subplots(figsize=(25, 10))
plt.ylabel('Min Residential Property Price (€)')
sns.barplot(x=county_price_min['county'], y= min_house_prices, order=county_price_min.sort_values('min_house_pric
plt.show()
```

In [ ]:
```
property_sizes = np.delete(df_merge_col['property_size_description'].unique(), 0)
property_sizes
```

In [ ]:
```
median_per_property_size = [df_merge_col['price'][df_merge_col['property_size_description']==property_size].media
```

In [ ]:
```
median_per_property_size
```

In [ ]:
```
# Replacing the values with another string
property_sizes = list(map(lambda x: x.replace('greater than 125 sq metres', '> 125 Sqm'), property_sizes))
```

In [ ]:
```
property_sizes = list(map(lambda x: x.replace('greater than or equal to 38 sq metres and less than 125 sq metres'
```

In [ ]:
```
property_sizes = list(map(lambda x: x.replace('less than 38 sq metres', '< 38 Sqm'), property_sizes))
```

In [ ]:
```
property_sizes = list(map(lambda x: x.replace('greater than or equal to 125 sq metres', '>= 125 Sqm'), property_s
```

In [ ]:
```
property_sizes
```

In [ ]:
```
size_price = pd.DataFrame(property_sizes)
size_price
```

In [ ]:
```
size_price['median_per_property_size'] = median_per_property_size
size_price.rename(columns={0:'property_sizes'},inplace=True)
size_price
```

In [ ]:
```
fig, ax = plt.subplots(figsize=(10, 5))
plt.bar(size_price['property_sizes'], size_price['median_per_property_size'] )
#ax.set_xticklabels(labels=property_sizes,rotation=90)
plt.xlabel('Property Sizes')
#plt.xlabel(" Property Sizes, 0: <38 sqm, 1: =>38 sqm, 2: =>125 sqm")
plt.ylabel('Median House Prices (€)')
plt.show()
```

In [ ]:
```
property_ = (df_merge_col['property_description'].unique())
property_
```

In [ ]:
```
propertysize_house_prices = [(df_merge_col[df_merge_col['property_description'] == property_description]['price']
```

In [ ]:
```
propertysize_house_prices
```

In [ ]:
```
property_ = list(map(lambda x: x.replace('Second-Hand Dwelling house /Apartment', 'UsedHouse'), property_))
```

In [ ]:
```
property_ = list(map(lambda x: x.replace('New Dwelling house /Apartment', 'NewHouse'), property_))
```

In [ ]:
```
fig, ax = plt.subplots(figsize=(5, 5))
plt.bar(property_,propertysize_house_prices)
#ax.set_xticklabels(labels=property_,rotation=90)
```

```
plt.xlabel('Property Type')
plt.ylabel('Median House Prices (€)')
plt.show()
```

In [ ]:
```
df_merge_col[df_merge_col['county'] == 'Dublin']['postal_code'].unique()

for postal_code in df_merge_col[df_merge_col['county'] == 'Dublin']['postal_code'].unique():
      print('Median house price in {}: {:.2f}k'.format(postal_code, df_merge_col['price'][df_merge_col['postal_co
```

In [ ]:
```
post_codes = np.delete(df_merge_col[df_merge_col['county'] == 'Dublin']['postal_code'].unique(), 2)
post_codes
```

In [ ]:
```
# getting the prices based on post codes in dublin
median_per_postal_code = [df_merge_col['price'][df_merge_col['postal_code']==postal_code].median() for postal_coc

median_per_postal_code = np.asarray(median_per_postal_code)
median_per_postal_code
```

In [ ]:
```
#df_merge_col['price'] = df_merge_col['price'].astype(float)
q=[]
for price in median_per_postal_code:
    x = price/10**3
    q.append(x)
```

In [ ]:
```
q.pop(0)
q.pop(21)
post_codes =post_codes.tolist()
post_codes.pop(0)
post_codes.pop(21)
```

In [ ]:
```
print(post_codes)
```

In [ ]:
```
print(len(q))
```

In [ ]:
```
print(len(post_codes))
```

In [ ]:
```
post_price = pd.DataFrame(q)
```

In [ ]:
```
post_price['post_code'] = post_codes
```

In [ ]:
```
post_price = post_price.dropna()
```

In [ ]:
```
print(post_price)
```

In [ ]:
```
post_price.rename(columns={0:'price'},inplace=True)
```

In [ ]:
```
post_price
```

In [ ]:
```
fig, ax = plt.subplots(figsize=(22, 15))
sns.barplot(x = post_price['post_code'],y = post_price['price'], order=post_price.sort_values('price', ascending
ax.set_xticklabels(labels=post_price['post_code'],rotation=90)
#layout = { xaxis={'categoryorder':'total descending'})
plt.xlabel('Post Codes')
plt.ylabel('Median House Prices (€k)')
plt.show()
```

In [ ]:
```
counties = df_merge_col['county'].unique()
counties
```

In [ ]:
```
median_per_county = [df_merge_col['price'][df_merge_col['county']==county].median() for county in counties]

median_per_county = np.asarray(median_per_county)
median_per_county
```

In [ ]:
```
q=[]
for price in median_per_county:
    x = price
    q.append(x)
```

In [ ]:
```
county_price = pd.DataFrame(q)
```

In [ ]:
```
county_price['county'] = counties
```

In [ ]:
```
county_price.rename(columns={0:'price'},inplace=True)
```

Univariate Plots

In [ ]:
```
sns.displot(np.log(df_merge_col['price']))
plt.show()
```

In [ ]:
```
df_merge_col
```

In [ ]:
```
sns.countplot(x='VAT_exclusive', data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.countplot(x='FMP', data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.countplot(x='location', data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.countplot(x='county', data=df_merge_col, order = df_merge_col['county'].value_counts().index)
locs, labels = plt.xticks()
plt.setp(labels, rotation=65)
plt.show()
```

In [ ]:
```
sns.countplot(x='year', data=df_merge_col, order = df_merge_col['year'].value_counts().index)
plt.show()
```

In [ ]:
```
sns.countplot(x='month', data=df_merge_col, order = df_merge_col['month'].value_counts().index)
plt.show()
```

In [ ]:
```
sns.countplot(x='property_description', data=df_merge_col, palette = 'mako')
plt.xlabel('Property Type')
plt.show()
```

In [ ]:
```
df_merge_col['property_size_description'] = df_merge_col['property_size_description'].replace(['greater than or e
```

In [ ]:
```
sns.countplot(x='property_size_description', data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.countplot(x='province', data=df_merge_col, palette = 'mako')
#sns.color_palette("light:#5A9", as_cmap=True)
plt.show()
```

Bivariate Plots

In [ ]:
```
sns.boxplot(x='year', y='month', data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.barplot(x='year', y='price', data=df_merge_col, ci=None, palette = 'mako')
plt.xlabel('Year')
plt.ylabel('Price in Euro')
plt.show()
```

In [ ]:
```
sns.countplot(x='year',hue='location', data=df_merge_col, order = df_merge_col['year'].value_counts().index)
plt.show()
```

In [ ]:
```
sns.countplot(x='year',hue='property_description', data=df_merge_col, order = df_merge_col['year'].value_counts()
plt.show()
```

In [ ]:
```
sns.countplot(x='year',hue='property_size_description', data=df_merge_col, order = df_merge_col['year'].value_cou
plt.show()
```

In [ ]:
```
sns.lineplot(x='month', y=np.log(df_merge_col['price']), data=df_merge_col, ci=None)
plt.show()
```

In [ ]:
```
#  bivariate with Price
sns.boxplot(x='location', y=np.log(df_merge_col['price']), data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.boxplot(x='property_description', y=np.log(df_merge_col['price']), data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.boxplot(x='property_size_description', y=np.log(df_merge_col['price']), data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.boxplot(x='province', y=np.log(df_merge_col['price']), data=df_merge_col)
plt.show()
```

In [ ]:
```
# with location
sns.countplot(x='property_size_description',hue='location', data=df_merge_col, order = df_merge_col['property_siz
plt.show()
```

In [ ]:
```
sns.countplot(x='property_description',hue='location', data=df_merge_col, order = df_merge_col['property_descript
plt.show()
```

In [ ]:
```
# geogrpahy and price
plt.scatter(df_merge_col['lat'],np.log(df_merge_col['price']) ,edgecolors='r')
plt.xlabel('Latitude')
plt.ylabel('Price In Euro')
plt.show()
```

In [ ]:
```
plt.scatter(df_merge_col['lon'],np.log(df_merge_col['price']) ,edgecolors='r')
plt.xlabel('Longitude')
plt.ylabel('Price In Euro')
plt.show()
```

Multivariate

In [ ]:
```
p1 = sns.lineplot(data=df_merge_col, x="year", y="price",hue='province', ci=None)
p1.set_yscale('log')
```

In [ ]:
```
p1 = sns.lineplot(data=df_merge_col, x="year", y="price",hue='location', ci=None)
p1.set_yscale('log')
```

In [ ]:
```
p1 = sns.lineplot(data=df_merge_col, x="year", y="price",hue='property_description',ci=None)
p1.set_yscale('log')
```

In [ ]:
```
sns.boxplot(x='year', y=np.log(df_merge_col['price']), hue='location', data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.boxplot(x='year', y=np.log(df_merge_col['price']), hue='property_description', data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.boxplot(x='year', y=np.log(df_merge_col['price']), hue='property_size_description', data=df_merge_col)
```

```
plt.show()
```

In [ ]:
```
sns.boxplot(x='location', y=np.log(df_merge_col['price']), hue='property_size_description', data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.boxplot(x='province', y=np.log(df_merge_col['price']), hue='property_size_description', data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.boxplot(x='province', y=np.log(df_merge_col['price']), hue='property_description', data=df_merge_col)
plt.show()
```

In [ ]:
```
sns.boxplot(x='location', y=np.log(df_merge_col['price']), hue='property_description', data=df_merge_col)
plt.show()
```

Initail Analyses

MLR

In [ ]:
```
rppr1 = df_merge_col.copy()
rppr1.drop(columns =['month_year','date_of_sale','address','VAT_exclusive','FMP','postal_code','county'], inplace
```

In [ ]:
```
#pd.get_dummies(rppr1["location"])
rppr1["location_Dublin"]=pd.get_dummies(rppr1["location"])["Dublin"]
```

In [ ]:
```
#pd.get_dummies(rppr1["property_description"])
rppr1["property_new"]=pd.get_dummies(rppr1["property_description"])["NewHouse"]
```

In [ ]:
```
#pd.get_dummies(rppr1["province"])
#rppr1["provinces_Leinster"]=pd.get_dummies(rppr["province"])["Leinster"]
rppr1["provinces_connacht"]=pd.get_dummies(rppr1["province"])["Connacht"]
rppr1["provinces_Ulster"]=pd.get_dummies(rppr1["province"])["Ulster"]
rppr1["provinces_munster"]=pd.get_dummies(rppr1["province"])["Munster"]
```

In [ ]:
```
from numpy import sqrt
log_price = np.log(rppr1['price'])
transform  = sqrt(log_price)
```

In [ ]:
```
X = rppr1[["location_Dublin","property_new","provinces_connacht", "provinces_Ulster", "provinces_munster","year",
X = sm.add_constant(X)
y = transform
#X.head(20)
```

In [ ]:
```
model_full_mlr = sm.OLS(y, X).fit()
```

In [ ]:
```
#fitted values
model_fitted_vals = model_full_mlr.fittedvalues
#model residuals
model_residuals = model_full_mlr.resid
#standardised residuals
model_norm_residuals = model_full_mlr.get_influence().resid_studentized_internal
```

In [ ]:
```
sns.regplot(x=model_fitted_vals,y=model_residuals,
 ci=False,lowess=True,
 line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.show()
```

In [ ]:
```
stats.probplot(model_norm_residuals, plot=sns.mpl.pyplot)
plt.show()

plt.hist(model_norm_residuals)
plt.show()
```

In [ ]:
```
from statsmodels.formula.api import ols
```

```
model_full_mlr1 = ols('log_price ~ C(year)+C(province)+C(location)+C(property_description)+lat+lon', data=rppr1).
model_full_mlr1.summary()
```

In [ ]:
```
#model_full_mlr.summary()
```

A random sample of 9423 obsevrations is seleceted from the whole data for statistical analyses.

In [ ]:
```
# 9423 random sample
rppr_sub = df_merge_col.sample(n=9423, random_state=3)
rppr_sub
```

In [ ]:
```
#log_price = np.log(rppr_sub['price'])
```

SLR

In [ ]:
```
# Try SLR
X = rppr_sub["year"]
X = sm.add_constant(X)
y = np.log(rppr_sub['price'])
model_slr = sm.OLS(y, X).fit()
#fitted values
model_fitted_vals = model_slr.fittedvalues
#model residuals
model_residuals = model_slr.resid
#standardised residuals
model_norm_residuals = model_slr.get_influence().resid_studentized_internal
```

In [ ]:
```
sns.regplot(x=model_fitted_vals,y=model_residuals,
ci=False,lowess=True,
line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.show()
```

In [ ]:
```
stats.probplot(model_norm_residuals, plot=sns.mpl.pyplot)
plt.show()
```

In [ ]:
```
X = rppr_sub["lat"]
X = sm.add_constant(X)
y = rppr_sub['price']
model_slr = sm.OLS(y, X).fit()
#fitted values
model_fitted_vals1 = model_slr.fittedvalues
#model residuals
model_residuals1 = model_slr.resid
#standardised residuals
model_norm_residuals1 = model_slr.get_influence().resid_studentized_internal
```

In [ ]:
```
sns.regplot(x=model_fitted_vals1,y=model_residuals1,
ci=False,lowess=True,
line_kws={'color': 'red', 'lw': 1, 'alpha': 0.25})
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.ylim(0,2000000)
plt.show()
```

In [ ]:
```
stats.probplot(model_norm_residuals1, plot=sns.mpl.pyplot)
#plt.ylim(0,30)
plt.show()

plt.hist(model_norm_residuals1)
plt.show()
```

These are not meeting the assumptions. So I am not proceeding with SLR.

Dropping the address, latitude and longitude and date column since it is not using in following analysis.

ANOVA

In [ ]:

```
rppr = rppr_sub.copy()
rppr.drop(columns =['month_year', 'lat','lon','date_of_sale','address','VAT_exclusive','FMP','postal_code','count
```

In [ ]:
```
rppr
```

In [ ]:
```
pd.get_dummies(rppr["location"])
rppr["location_Dublin"]=pd.get_dummies(rppr["location"])["Dublin"]
```

In [ ]:
```
pd.get_dummies(rppr["property_description"])
rppr["property_new"]=pd.get_dummies(rppr["property_description"])["NewHouse"]
```

In [ ]:
```
#rpp = rppr[['price','year']]
from numpy import sqrt
log_price = np.log(rppr['price'])
transform = series = sqrt(log_price)
```

In [ ]:
```
log_price = np.log(rppr['price'])
```

In [ ]:
```
d1= pd.crosstab(index=rppr['location'], columns=rppr["year"], margins=True)
d1
```

In [ ]:
```
## property_size_description variable has lots of NaN values and hence it is not considered.
```

In [ ]:
```
#perform two-way ANOVA without interaction
model2t = ols('log_price~ C(year) + C(location) + C(property_description) + C(province)', data=rppr_sub).fit()
#fitted values
model_fitted_vals2 = model2t.fittedvalues
#model residuals
model_residuals2 = model2t.resid
#standardised residuals
model_norm_residuals2t = model2t.get_influence().resid_studentized_internal
```

In [ ]:
```
sns.regplot(x=model_fitted_vals2,y=model_residuals2,
ci=False,lowess=True,
line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.show()
```

In [ ]:
```
stats.probplot(model_norm_residuals2t, plot=sns.mpl.pyplot)
plt.show()
```

In [ ]:
```
sm.stats.anova_lm(model2t, typ=2)
```

In [ ]:
```
model2t.summary()
```

In [ ]:
```
#perform two-way ANOVA without interaction
model3 = ols('log_price~ C(year) + C(location) + C(property_description)',
 data=rppr_sub).fit()
#fitted values
model_fitted_vals3 = model3.fittedvalues
#model residuals
model_residuals3 = model3.resid
#standardised residuals
model_norm_residuals3 = model3.get_influence().resid_studentized_internal
```

In [ ]:
```
sns.regplot(x=model_fitted_vals3,y=model_residuals3,
ci=False,lowess=True,
line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.show()
```

In [ ]:
```
stats.probplot(model_norm_residuals3, plot=sns.mpl.pyplot)
plt.show()
```

In [ ]:
```
model3.summary()
```

APPENDIX E

Project Progress

by Ginu Varghese (https://mahara.dkit.ie/user/view.php?id=1 3318)

02/03/2022

Discussion points:

- Discussed the overview of project.
- Received the clarifications for doubts and tasks for next week from the supervisor.

Deliverables:

- Find out the extra data to be added to the existing dataset. -- Got 2 datasets to add. Couldn't find the one related to unemployment.
- Find the literature reviews done in the same area and identify the methods used by them. -- Found old papers.
- Modify the research questions. -- Done

Date of next meeting: 09/03/2022 at 1.15 pm.

09/03/2022

Discussion points:

- Discussed about the previous papers and the methods used in those papers.
- Discussed the datasets to merge and how to merge it
- Discussed about the Google API for latitude and longitudes.

Deliverables:

- Merge datasets
- Find more papers related to Ireland.
- Google API for latitude and longitudes. --- Couldn't read data to Jupyter. Found a code for Google API for latitudes and longitudes.

Date of next meeting: 16/03/2022 at 1.15 pm

16/03/2022

Discussion Points:

- Discussed about the dataset and how to solve the issue with reading the dataset.

Deliverables:

- Ethical consideration form – Filled
- Merging province dataset to original dataset. -- Done

Date of next meeting: 23/03/2022 at 1pm

23/03/2022

Discussion points:

- Discussed the problem with getting latitude and longitude through API. And suggested to look fordifferent ways.
- Suggested that price can be made category for analysis and make another column without removing the existing one.
- Discussed about the importance of literature review and asked to read more and more.

Tasks:

- Start cleaning and visualization

Next meeting: 30/03/2022 at 3pm

30/03/2022

Discussion points:

- Suggested to split the county column to Dublin and outside Dublin and add another column. Thiswill be useful for analysis.
- Discussed about the Latitude and longitude API and it is ok to do with that.
- Discussed about doing different model on the data.

Tasks:

- Make another column with Dublin and outside Dublin - done

- Try out different models - done there is some doubts and errors

Next meeting: 06/04/2022 at lpm

06/04/2022

Discussion Points:

- Discussed how to write literature review.
- Decided to implement both statistical and machine learning techniques.
- Supervisor suggested an article and asked to read it.

Deliverables:

- Read articles - done
- Make another column with Dublin and outside Dublin - done
    - Start writing and visualizing - not completed but started

Next meeting: 27/04/2022 after Easter holidays at 1.30 PM

27/04/2022

Discussion points:

- Discussed about the literature review.
- Discussed about the article sent by supervisor.
- Clarified some doubts.

Deliverables:

- Start writing and visualizing - Ongoing

Next meeting: 04705/2022 at 3 pm

04/05/2022

Discussion points:

- Supervisor asked to finalize the ML and statistical methods using in the thesis.
- Suggested to perform all models and include the best in the report.
- Clarified some doubts regarding the ML techniques.

Deliverables:

- Literature review - Ongoing

Next meeting: 13/05/2022 at 9.30 am

13/05/2022

Discussion points:

- Clarified the doubts regarding the literature review.
- Asked doubts regarding choropleth

Deliverables:

- Start doing interim report works -ongoing

Next meeting: 26/05/2022 at 10 am

26/05/2022

Discussion Points:

- Discussed about the literature review and report and the sections to be included in the report.
- Discussed about the initial analyses to be included in the report
- Discussed about the plotly map of Ireland

Deliverables:

- Start doing report - completed visualizations

Next meeting on: 03/06/2022 at 9 am in college

03/06/2022

Discussion Points:

- Discussed about the literature review and report and the feedback comments.
- Discussed about the initial analyses to be included in the report.
- Suggested a plan to complete report by Wednesday and send it for reviewing.

Deliverables:

- Start doing report - completed visualizations and Literature review

Next meeting on: 08/06/2022 at 3.30 pm via zoom

08/06/2022

Discussion Points:

- Discussed about the ML algorithms and its implementation.
- Suggested not to remove the null values since it may affect the modelling.
- Shown the report via screen share and discussed it.

Deliverables:

- Report - Send it to Supervisor for review

Next meeting on: 10/06/2022 at 2.40 pm via zoom

10/06/2022

Discussion Points:

- Clarified the doubts regarding the feedback comments.

Deliverables:

- Report - Completed

Next meeting on: 15/06/2022 at 9 via zoom

15/6/22

Provided feedback for the report and submission.

Marks for the interim report has been shared.

Added everything related to project to the Git hub.

22/6/22

Started working on the code to get the map of Ireland. Searched different websites regarding the same.

29/6/22

Got the code to make the choropleth of Ireland.

The map to plot Dublin post codes was showing some error.

Did some research on how to plot the post codes.

6/7/22

The shape file to plot the post codes was downloaded and the error was solved. Choropleth for Dublin post codes have been plotted. Started reading about hedonic pricing.

13/7/22

After researching about hedonic models, it was found that MLR models with price as dependent variableis the hedonic models. So started writing the results for MLR models which was already implemented for interim report.

20/7/22

Completed the writing for hedonic model.

Started implementing statistical models for Dublin data to analyze the price in Dublin. A new data has created for Dublin.

27/7/22

Completed the models for Dublin data and implemented SLR and MLR models.Started writing the interpretation for the models.

3/8/22

Completed the writing for Dublin data models and started researching research papers regarding thefactors influencing house prices to add to literature review. Found two papers and started reading it.

10/8/22

Read two papers regarding the factors influencing house prices and added them to the literature review.

17/8/22

Started working on ML models for the project and started with the LR and DCT.

24/8/22

Completed the LR, DCT, RF, XG-boost models and started writing the results section of report.

26/8/22

Completed the results section and then added one more model to the project ie. the NN model.

Updated the corresponding sections in the report and uploaded it to the git.

28/8/22

Updated the report and added the evaluation section and completed the report.

29/8/22

Added all the code and updated all files on GitHub.

Performed the review of report and corrected the formats and did spell check. Added some more figures to the results section and updated the figure numbers.

30/8/22

Added the appendix section to the report.

This includes the code, Mahara page and ethical approval form.

Added the final draft to GitHub.

1/9/22

Submit the report draft to supervisor for review.

11/9/22

Final submission.

DUNDALK INSTITUTE OF TECHNOLOGY

School of Informatics & Creative Arts

Ethical Approval Form for Research Projects

Researcher Name: Ginu Varghese Year: 2021-2022 Course: MSc in Data Analytics

Title of project: Analysis of House Pricing in Ireland

Name of supervisor: Jack Mc Donnell Date: 22/03/2022

(if applicable)

This application is to be completed by the researcher and where appropriate, in conjunction with the project supervisor. The lead researcher/supervisor is responsible for submitting the completed form to the appropriate Research Ethics Committee (details below)

Please note: If your submission is incomplete or unclear, your application will be returned to you and your project may be delayed.

Section 1

Type of Researcher

Please tick the appropriate box below to indicate the type of researcher you are:

Undergraduate ☐

(proceed to section 2)

Completed Ethical Approval forms for undergraduate research should be submitted to the relevant Departmental Research Ethics Committee (DREC).

Drec.dcamm@dkit.ie – Department of Creative Arts Media and Music

Drec.dcsm@dkit.ie – Department of Computing Science and Mathematics

Drec.dvhcc@dkit.ie – Department of Visual and Human Centred Computing

Postgraduate ☑

(proceed to section 3)

Completed Ethical Approval forms for Postgraduate research should be submitted directly to the School Research Ethics Committee (SREC).

Srec.ica@dkit.ie

Staff

(proceed to section 3) ☐

Completed Ethical Approval forms for Staff research should be submitted directly to the School Research Ethics Committee (SREC).

Srec.ica@dkit.ie

Section 2

Please complete questions 1-4 listed below.

	Human and / or Animal Research	YES	NO
1	**Does your research involve human participants other than the following[1]?** 1. Research using exclusively secondary sources. 2. Research using materials legally accessible to the public that have legal protection, e.g., record of court judgements, data archives. 3. Research using materials that are publicly accessible and where there is no reasonable expectation for privacy, e.g., books, published third party interviews. 4. Observations of human behaviour in public where (i) those being observed have no reasonable expectation of privacy, (ii) there is no intervention on the part of the researcher nor any interaction between the researcher and those observed, and (iii) individuals are not identifiable in the results. If '**YES**', please complete **B and C** below.		
2	**Does your project involve working with animals?** – If '**YES**', please complete **B and C** below. – Please note that for ethical consideration: 'Animals' are classed as vertebrate animals including cyclostomes and cephalopods (DIRECTIVE 2010/63/EU)		
3	**Does your project involve working with participants from any of the following categories?** 1. Minors (under 18 years of age) 2. People with learning or communication difficulties		

[1] If there is any doubt, researchers should contact the Chair of the SREC.

	3. Patients 4. People in custody 5. People engaged in illegal activities If '**YES**', please complete **D** below.		
4	**Does your project have any possible ethical implications other than those outlined in questions 1, 2 and 3?** If '**YES**', please complete **E** below.		

1. **I consider that this project has no significant ethical implications[2] to be brought through the ICA School Ethics Review Process**

Please complete Section 4

2. **Is this study part of a larger project that already has ethical clearance?**

YES / NO ☐

If **YES** please answer question B II.

If **NO** please answer question C.

1. If this study is part of a larger project that already has ethical clearance, are you proposing any changes to the operational plan already ethically approved?

YES / NO

If **YES,** please complete *Sections 3 and 4.*

If **NO,** then please provide the project details below and complete *Section 4.*

[2] In determining significant implications, please consider all potential risks attached to this project. If there is any doubt, researchers should contact the Chair of the SREC.

Title of project with ethical clearance: _____

3. **Could this project have ethical implications that should be brought before the appropriate ICA Departmental Ethics Review Committee as it will be carried out with human participants?**

   YES / NO

   If **YES,** please complete *Sections 3 and 4.*

   If **NO,** please complete *Section 4.*

4. **I consider that this project may have ethical implications that should be brought before the School Research Ethics Committee as it will be carried out with human participants in a "vulnerable" category.** ☐

   Please complete Sections 3 and 4

   All research carried out with human participants in a vulnerable category must be referred by the Departmental Research Ethics Committee to the School Research Ethics Committee for approval.

5. **Could this project have ethical implications, other than those previously outlined, that should be brought before the appropriate ICA Departmental Ethics Review Committee?**

   YES / NO

   If **YES,** please complete *Sections 3 and 4.*

   If **NO,** please complete *Section 4.*

Section 3

3.1 Application Form Checklist

Please complete Section 3 and provide additional information as attachments.

My application includes the following documentation:	**INCLUDED (mark as YES)**	**NOT APPLICABLE (mark as N/A)**
Recruitment advertisement		N/A
Participant Information Leaflet		N/A
Participant Informed Consent form		N/A
Questionnaire/Survey		N/A
Interview/Focus Group Questions		N/A
Debriefing material		N/A
Evidence of approval to gain access to off-site location		N/A
Ethical approval from external organizations. If ethical approval from external organizations is pending give details below		N/A
Details N/A		

3.2 Project Details

1. Lay description (Maximum 200 words)

Please outline, in terms that any non-expert would understand, what your research project is about, including what participants will be required to do. Please explain any technical terms or discipline-specific phrases.

The property prices in Ireland is increasing day by day. In this situation, I have chosen my project as to analyze the property prices in Ireland and find out what are the factors that are contributing towards the increase in the house prices. There are no external participants required in the project. The analysis will be done based on the data available from the property prices website and no personal information is needed here. I am planning to incorporate the different statistical as well as machine learning techniques to analyze the property prices.

Analysis will be done for the data from 2010 to 2022. There is information about the county in which the property is situated, the price of house, description of property address and date of sale. I would like to add the province and latitude-longitude information to the dataset for better analysis.

2. Research objectives (Maximum 150 words)

Please summarise briefly the objectives of the research.

- Find out the property prices in each year.
- Find out the counties in which most of the houses are sold.
- Find out the counties in which the least number of houses sold.
- Find out the house prices in each county/province.
- Find out the year in which most houses were sold.
- Find out the trend in the property pricing over the years.
- Find out either the New or secondhand dwelling has the highest prices and is it inclusive of VAT.
- To analyze how the house prices depends on the type of house and its size.

3. Research location and duration

Location(s)/Population*	DkIT Campus
Research start date	01/06/2022
Research end date	16/09/2022
Approximate duration	3 Months

* If location/Population other than DkIT campus/population, provide details of the approval to gain access to that location/population as an appendix.

3.3 Participants

		YES	**NO**	**N/A**
Do participants fall into any of the following special groups?	**Minors (under 18 years of age)**			**N/A**
	People with learning or communication difficulties			**N/A**
	Patients			**N/A**

	People in custody			**N/A**
	People engaged in illegal activities (e.g. drug-taking)			**N/A**
Have you given due consideration to the need for satisfactory Garda clearance?				**N/A**

3.4 Sample Details

Approximate number	N/A
Where will participants be recruited from?	N/A
Inclusion Criteria	N/A
Exclusion Criteria	N/A
Will participants be remunerated, and if so in what form? N/A	

Justification for proposed sample size and for selecting a specific gender, age, or any other group if this is done in your research. N/A

3.5 Risk to Participants

a) Please describe any risks to participants that may arise due to the research. Such risks could include physical stress, emotional distress, perceived coercion e.g. lecturer interviewing own students. Detail the measures and considerations you have put in place to minimize these risks

N/A

b) What will you communicate to participants about any identified risks? Will any information be withheld from them about the research purpose or procedure? If so, please justify this decision.

N/A

3.6 Informed Consent

	YES	NO	N/A
Will you obtain active consent for participation?			N/A
Will you describe the main experimental procedures to participants in advance?			N/A
Will you inform the participants that their participation is voluntary and may be withdrawn at any point?			N/A
If the research is observational, will you ask for their consent to being observed?			N/A
With questionnaires, will you give participants the option of omitting questions they do not want to answer?			N/A
Will you tell participants that their data will be treated with full confidentiality and that, if published, it will not be identifiable as theirs?			N/A

Will the data be anonymous?			N/A
Will you debrief participants at the end of their participation?			N/A
Will your project involve deliberately misleading participants in any way, or will information be withheld? **If you answer yes, give details and justification for doing this below.**			N/A
N/A			

1. Please outline your approach to ensuring the confidentiality of data (that is, that the data will only be accessible to agreed upon parties and the safeguarding mechanisms you will put in place to achieve this.) You should include details on how and where the data will be stored, and who will have access to it. There is no personal data involved in the project. All the data being used is downloaded from https://propertypriceregister.ie/ website. It is open to the public for download and use. So, there is no confidentiality of data involved.

2. Please outline how long the data will be retained for, if it will be destroyed and how it will be destroyed.

Standard DkIT procedures and guidelines will be followed. No confidential data is being used for this project.

1.Storage – N/A

2. Access – N/A

3. Communication – N/A

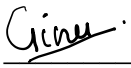
4. Digital Platform Usage – N/A

5. Data maintenance – N/A

Section 4

Researcher I have read and I understand the DkIT Ethics Policy available from:

https://www.dkit.ie/assets/uploads/documents/Research/Policies/DkIT%20Research%20Ethics%20Policy.pdf

Signed:  Print Name: Ginu Varghese Date: 22/03/2022

(Researcher)

Supervisor: Applications for Ethical Approval of Undergraduate projects are forwarded to the Departmental Research Ethics Committee for approval or referral to the School Research Ethics Committee. Applications for Ethical Approval of Postgraduate and Staff projects are sent to the School Research Ethics Committee for Approval.

I have read and approved this form & information:

Signed: _____ Print Name: Dr Martin Mc Hugh Date: 12 May 2022

(Supervisor/Head of Department/ Research Centre Director/ Head of School)

There is an obligation on the researcher and/or supervisor to bring to the attention of the Departmental/School Research Ethics Committee(s): (a) Any issues with ethical implications not clearly covered by this form (b) Any ethical issues which may arise during the carrying out of the research; (c) Any ethically significant change made to the project after approval.

Section 5 (For office use only)

STATEMENT OF ETHICAL APPROVAL (FOR UNDERGRADUATE PROJECTS ONLY)

This project has been considered using agreed department procedures and is now:

☐ Approved:

☐ Referred to the School Ethics Committee:

Signed: _____ Print Name: _____ Date: _____

(Chair of Departmental Research Ethics Committee/Head of Department)

STATEMENT OF ETHICAL APPROVAL

This project has been considered using agreed School procedures and is now:

☐ Approved:

☐ Rejected (further information sought):

Chair of School Research Ethics Committee

This project has been considered by the Ethics Committee and ethical approval is granted.

Signed: _____ Print Name: _____ Date: _____

Chair of School Research Ethics Committee

REFERENCES

Abdulai, R. and Owusu-Ansah, A. (2011). *(PDF) Hedonic regression analysis of house price determinants in Liverpool, England* [online]. , pp.6–31. Available from: https://www.researchgate.net/publication/286675668_Hedonic_regression_analysis_of_house_price_determinants_in_Liverpool_England [accessed 1 June 2022].

Advantages of Support Vector Machines (SVM). Available from: https://iq.opengenus.org/advantages-of-svm/ [accessed 2 June 2022].

Ajay Ohri. (2022). *10 Popular Regression Algorithms In Machine Learning Of 2022* [online]. Available from: https://www.jigsawacademy.com/popular-regression-algorithms-ml/ [accessed 22 August 2022].

Alfiyatin, A.N., Febrita, R.E., Taufiq, H. and Mahmudy, W.F. (2017). Modeling House Price Prediction using Regression Analysis and Particle Swarm Optimization Case Study : Malang, East Java, Indonesia. *International Journal of Advanced Computer Science and Applications* [online], 8(10). Available from: www.ijacsa.thesai.org [accessed 1 June 2022].

Amrutha. (2022). *Decision Tree Machine Learning Algorithm - Analytics Vidhya* [online]. *https://www.analyticsvidhya.com/* [online]. Available from: https://www.analyticsvidhya.com/blog/2022/01/decision-tree-machine-learning-algorithm/ [accessed 5 June 2022].

analyticsvidhya. (2018). *XGBoost Algorithm | XGBoost In Machine Learning* [online]. *analyticsvidhya.com* [online]. Available from: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/ [accessed 5 June 2022].

Andrade, F. (2021). *A Simple Guide to Scikit-Learn — Building a Machine Learning Model in Python | by Frank Andrade | Towards Data Science* [online]. *https://towardsdatascience.com/* [online]. Available from: https://towardsdatascience.com/a-beginners-guide-to-text-classification-with-scikit-learn-632357e16f3a [accessed 5 June 2022].

Antonina Mavrodiy. (2005). *Factors analysis of real estate prices* [online]. Available from: https://kse.ua/wp-content/uploads/2019/02/mavrodiy.pdf [accessed 23 August 2022].

Bajaj Aayush. (2022). *Performance Metrics in Machine Learning [Complete Guide] - neptune.ai* [online]. Available from: https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide [accessed 22 August 2022].

Brian Beers. (2022). *P-Value: What It Is, How to Calculate It, and Why It Matters* [online]. Available from: https://www.investopedia.com/terms/p/p-value.asp [accessed 5 September 2022].

Brownlee, J. (2014). *A Gentle Introduction to Scikit-Learn* [online]. *https://machinelearningmastery.com/* [online]. Available from: https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/ [accessed 5 June 2022].

CFI. (2022). *Hedonic Regression Method - Overview, Application, Function* [online]. *https://corporatefinanceinstitute.com/* [online]. Available from: https://corporatefinanceinstitute.com/resources/knowledge/other/hedonic-regression-method/ [accessed 5 June 2022].

CFITeam. (2022). *Hedonic Regression Method - Overview, Application, Function* [online]. Available from: https://corporatefinanceinstitute.com/resources/knowledge/other/hedonic-regression-method/ [accessed 19 August 2022].

Charlie Weston. (2021). *How 'working from home' has turned rural areas into property hotspots - Independent.ie* [online]. Available from: https://www.independent.ie/business/personal-finance/property-mortgages/how-working-from-home-has-turnedrural-areas-into-property-hotspots-40746871.html [accessed 5 September 2022].

Chen, J.H., Ong, C.F., Zheng, L. and Hsu, S.C. (2017). Forecasting spatial dynamics of the housing market using Support Vector Machine. *International Journal of Strategic Property Management* [online], 21(3), pp.273–283.

Clay Ford. (2018). *Interpreting Log Transformations in a Linear Model | University of Virginia Library Research Data Services + Sciences* [online]. Available from: https://data.library.virginia.edu/interpreting-log-transformations-in-a-linear-model/ [accessed 5 September 2022].

Coughlan, M. (2022). *How everything is different for today's first-time buyers* [online]. Available from: https://www.rte.ie/news/primetime/2022/0130/1276779-how-everything-is-different-for-todays-first-time-buyers/ [accessed 2 June 2022].

CSO. (2022). *Home - CSO - Central Statistics Office* [online]. Available from: https://www.cso.ie/en/index.html [accessed 5 September 2022].

Deepanshi. (2021). *Linear Regression | Introduction to Linear Regression for Data Science* [online]. *https://www.analyticsvidhya.com/* [online]. Available from: https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-your-first-machine-learning-model-linear-regression/ [accessed 5 June 2022].

FitzGerald, G. (2007). *What caused the Celtic Tiger phenomenon? – The Irish Times* [online]. *The Irish Times* [online]. Available from: https://www.irishtimes.com/opinion/what-caused-the-celtic-tiger-phenomenon-1.950806 [accessed 8 June 2022].

Frew, J. and Jud, G.D. (2020). Estimating the Value of Apartment Buildings. *https://doi.org/10.1080/10835547.2003.12091101* [online], 25(1), pp.77–86. Available from: https://www.tandfonline.com/doi/abs/10.1080/10835547.2003.12091101 [accessed 2 June 2022].

Gandhi, R. (2018). *Support Vector Machine — Introduction to Machine Learning Algorithms | by Rohith Gandhi | Towards Data Science* [online]. *https://towardsdatascience.com/* [online]. Available from: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 [accessed 5 June 2022].

Gao, J., Ren, H. and Du, Y. (2018). Analysis of factors influencing the price of real estate based on interpretative structural model. , 1 February 2018, pp.43–49. Available from: https://www.atlantis-press.com/proceedings/ifmeita-17/25891072 [accessed 23 August 2022].

Gazette Desk. (2021). *Supply chain constraints will push up house prices, say builders* [online]. *Law Society of Ireland* [online]. Available from: https://www.lawsociety.ie/gazette/top-stories/2021/05-may/supply-chain-constraints-will-push-up-house-prices-say-builders [accessed 2 June 2022].

Gu, J., Zhu, M. and Jiang, L. (2011). Housing price forecasting based on genetic algorithm and support vector machine. *Expert Systems with Applications* [online], 38(4), pp.3383–3386.

Ho, W.K.O., Tang, B.S. and Wong, S.W. (2020). Predicting property prices with machine learning algorithms. *https://doi.org/10.1080/09599916.2020.1832558* [online], 38(1), pp.48–70. Available from: https://www.tandfonline.com/doi/abs/10.1080/09599916.2020.1832558 [accessed 1 June 2022].

Hurley, A.K., Sweeney, James, Hurley AoifeHurley, A.K., Hurley, A. and Sweeney, J. (2022). Irish Property Price Estimation Using A Flexible Geo-spatial Smoothing Approach: What is the Impact of an Address?. *The Journal of Real Estate Finance and Economics 2022* [online], 5 May 2022, pp.1–39. Available from: https://link.springer.com/article/10.1007/s11146-022-09888-y [accessed 30 May 2022].

Ihre, A. (2019). Predicting house prices with machine learning methods. , 2019.

investopedia. (2021). *Hedonic Regression Definition* [online]. *investopedia.com* [online]. Available from: https://www.investopedia.com/terms/h/hedonic-regression.asp [accessed 5 June 2022].

Ja'afar, N.S., Mohamad, J. and Ismail, S. (2021). Machine learning for property price prediction and price valuation: A systematic literature review. *Planning Malaysia* [online], 19(3), pp.411–422.

Javatpoint. (2010). *Keras Tutorial | Deep Learning with Python - Javatpoint* [online]. Available from: https://www.javatpoint.com/keras [accessed 25 August 2022].

javatpoint. *Machine Learning Decision Tree Classification Algorithm - Javatpoint* [online]. Available from: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm [accessed 19 August 2022].

Jose Doval Tedin, M., Faubert, V. and Economy, E. (2020). Housing Affordability in Ireland. , December 2020. Available from: https://ec.europa.eu/info/publications/economic-and-financial-affairs-publications_en. [accessed 2 June 2022].

K, D. (2019). *Top 5 advantages and disadvantages of Decision Tree Algorithm | by Dhiraj K | Medium* [online]. *Medium* [online]. Available from:

https://dhirajkumarblog.medium.com/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a [accessed 2 June 2022].

Kenton, W. (2022). *Analysis of Variance (ANOVA) Definition & Formula* [online]. *investopedia.com* [online]. Available from: https://www.investopedia.com/terms/a/anova.asp [accessed 7 June 2022].

Kumar, N. (2019). *The Professionals Point: Advantages of XGBoost Algorithm in Machine Learning* [online]. *theprofessionalspoint* [online]. Available from: http://theprofessionalspoint.blogspot.com/2019/03/advantages-of-xgboost-algorithm-in.html [accessed 2 June 2022].

Kunovac, D. and Zagreb, K.K. (2019). Residential Property Prices in Croatia. , 2019.

Li, D.Y., Xu, W., Zhao, H. and Chen, R.Q. (2009). A SVR based forecasting approach for real estate price prediction. *Proceedings of the 2009 International Conference on Machine Learning and Cybernetics* [online], 2, pp.970–974.

Limsombunc, V., Gan, C. and Lee, M. (2004). House Price Prediction: Hedonic Price Model vs. Artificial Neural Network. *American Journal of Applied Sciences* [online], 1(3), pp.193–201.

MacFarlane, I. (2022). *How the war in Ukraine could affect UK property – Show House* [online]. *https://www.showhouse.co.uk/* [online]. Available from: https://www.showhouse.co.uk/news/how-the-war-in-ukraine-could-affect-uk-property/ [accessed 2 June 2022].

Mbaabu, O. (2020). *Introduction to Random Forest in Machine Learning | Engineering Education (EngEd) Program | Section* [online]. *Section* [online]. Available from: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/ [accessed 2 June 2022].

Miao, D., Tang, H. and Wang, B. (2021). Support Vector Regression with Gaussian kernel for Housing Prices Prediction. *Journal of Physics: Conference Series* [online], 2021, pp.1–9. Available from: https://iopscience.iop.org/article/10.1088/1742-6596/1994/1/012023/pdf [accessed 1 June 2022].

Montero, J.-M. and Fernández-Avilés, G. (2014). Hedonic Price Model. *Encyclopedia of Quality of Life and Well-Being Research* [online], 2014, pp.2834–2837.

Available from: https://link.springer.com/referenceworkentry/10.1007/978-94-007-0753-5_1279 [accessed 2 June 2022].

Mu, J., Wu, F. and Zhang, A. (2014). Housing Value Forecasting Based on Machine Learning Methods. , 2014. Available from: http://dx.doi.org/10.1155/2014/648047 [accessed 31 May 2022].

Petrov, M. (2009). *The vulture has landed | EurobuildCEE* [online]. Available from: https://eurobuildcee.com/en/magazine/866-the-vulture-has-landed [accessed 2 June 2022].

Phan, D. (2018). Housing Price Prediction using Machine Learning Algorithms: The Case of Melbourne City, Australia. *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)* [online], 2018, pp.1–8. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8614000 [accessed 1 June 2022].

Phan, T.D. (2019). Housing price prediction using machine learning algorithms: the case of Melbourne city, Australia. *Proceedings - International Conference on Machine Learning and Data Engineering, iCMLDE 2018* [online], 15 January 2019, pp.35–42. Available from: https://researchers.mq.edu.au/en/publications/housing-price-prediction-using-machine-learning-algorithms-the-ca [accessed 25 May 2022].

Pow, N. and Janulewicz, E. (1995). Applied Machine Learning Project 4Prediction of real estate property prices in. *Machine Learning* [online], 20(3), pp.273–297.

Qualitrics. (2022a). *How to Determine Sample Size in Research | Qualtrics* [online]. *Qualitrics* [online]. Available from: https://www.qualtrics.com/uk/experience-management/research/determine-sample-size/ [accessed 11 June 2022].

Qualitrics. (2022b). *What is ANOVA (Analysis Of Variance) | Qualtrics* [online]. Available from: https://www.qualtrics.com/uk/experience-management/research/anova/ [accessed 19 August 2022].

Quang, T., Minh, N., Hy, D. and Bo, M. (2020). Housing Price Prediction via Improved Machine Learning Techniques. *Procedia Computer Science* [online], 174, pp.433–442.

Rahman, M.M., Khanam, R. and Xu, S. (2012). The Factors Affecting Housing Price in Hangzhou: An Empirical Analysis., 2012.

Ray, S. (2017). *SVM | Support Vector Machine Algorithm in Machine Learning* [online]. *https://www.analyticsvidhya.com/* [online]. Available from: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/ [accessed 5 June 2022].

Reddan, F. (2018). How high can they go? Six factors affecting Irish house prices – The Irish Times. *The Irish Times* [online], 18 September 2018. Available from: https://www.irishtimes.com/business/personal-finance/how-high-can-they-go-six-factors-affecting-irish-house-prices-1.3628349 [accessed 2 June 2022].

Saini, A. (2021). *Decision Tree Algorithm - A Complete Guide - Analytics Vidhya* [online]. *https://www.analyticsvidhya.com/* [online]. Available from: https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/ [accessed 5 June 2022].

Selim, S. (2008). DETERMINANTS OF HOUSE PRICES IN TURKEY: A HEDONIC REGRESSION MODEL Sibel SELİM., 9(1), pp.65–76.

Shi, H. and Li, W. (2009). Fusing Neural Networks, Genetic Algorithms and Fuzzy Logic for Analysis of Real Estate Price., 2009, pp.1–4. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5362675 [accessed 2 June 2022].

Shinde, N. and Gawande, K. (2018). VALUATION OF HOUSE PRICES USING PREDICTIVE TECHNIQUES. *International Journal of Advances in Electronics and Computer Science* [online], 2018, pp.2393–2835. Available from: http://iraj.in [accessed 3 June 2022].

Sullivan, A. (2021). *House prices: 'Wall of money' hits European real estate | Business | Economy and finance news from a German perspective | DW | 03.06.2021* [online]. Available from: https://www.dw.com/en/house-prices-wall-of-money-hits-european-real-estate/a-57765308 [accessed 2 June 2022].

SVM Algorithm - Javatpoint 2010. Available from: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm [accessed 18 August 2022].

Thamarai, M. and Malarvizhi, S.P. (2020). Information Engineering and Electronic Business. *Information Engineering and Electronic Business* [online], 2, pp.15–20. Available from: http://www.mecs-press.org/ [accessed 30 May 2022].

Vallor, S., William, J. and Rewak, S.J. *An Introduction to Data Ethics MODULE AUTHOR: 1* [online]. Available from: https://techethics.ieee.org [accessed 5 June 2022].

Vineet Jaiswal. (2018). *Interpret R Linear/Multiple Regression output | by Vineet Jaiswal | Analytics Vidhya | Medium* [online]. Available from: https://medium.com/analytics-vidhya/interpret-r-linear-multiple-regression-output-lm-output-point-by-point-also-with-python-8e53b2ee2a40 [accessed 22 August 2022].

Wang, X., Wen, J., Zhang, Y. and Wang, Y. (2014). Real estate price forecasting based on SVM optimized by PSO. *Optik* [online], 125(3), pp.1439–1443.

Waseem, M. (2022). *Linear Regression for Machine Learning | Intro to ML Algorithms | Edureka* [online]. *edureka* [online]. Available from: https://www.edureka.co/blog/linear-regression-for-machine-learning/ [accessed 2 June 2022].

Wu, J.Y. (2017). Housing Price prediction Using Support Vector Regression. , 2017.

Yee, L.W., Abu Bakar, N.A., Mohd Zainuddin, N.M. and Mohd Yusoff, R.C. (2021). Using Machine Learning to Forecast Residential Property Prices in Overcoming the Property Overhang Issue., 2021, pp.1–6. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9573830&casa_token=_yLGMqWPaLAAAAAA:bQg91cipVoaA5gHUGRqe1MsvooGwR_WjCbHx-xgvWqsLXaEbPpiN6WdyOnt45_GP7BmPgNvL0KkBJA&tag=1 [accessed 2 June 2022].

Yiu, T. (2019). *Understanding Random Forest. How the Algorithm Works and Why it Is… | by Tony Yiu | Towards Data Science* [online]. *Towards Data Science* [online]. Available from: https://towardsdatascience.com/understanding-random-forest-58381e0602d2 [accessed 11 June 2022].