**TRINITY COLLEGE DUBLIN**

**Management Science and Information Systems Studies**

**Project Report**

Creation of an Automated Valuation Model

**1. INTRODUCTION AND SUMMARY**

This chapter introduces the project background and the terms of reference. It also includes a summary of the remaining chapters in this report.

**1.1 Project Background**

The Irish property market has proven itself to be robust against major macroeconomic events, firstly recovering quickly from the market crash in 2008 going from -22% in 2011 to +22% in 2014, and yet again as house prices rose 7.7% in 2021, despite the Covid-19 pandemic (Daft.ie, 2022). The continued price growth in the Irish housing market even in the face of these major events is driven by the long-term market dynamic whereby housing demand far outweighs supply. In fact, the stock of houses for sale is at an all time low with just under 11,500 properties for sale on Daft.ie in December of 2021 as compared with close to 60,000 in 2012 (Daft.ie, 2022). Excess demand creates a highly competitive market pitting buyers against each other and spurring snap decisions that can often be misinformed. The overall aim of this project is to use data to help both sides of the market to make informed and data driven decisions when buying and selling.

The project is targeted at both the buyer and seller side of the property market and is broken into two objectives. The first objective is to create a property price prediction model, that will quickly and accurately predict property prices based on property features such as location, description, number of bedrooms and bathrooms etc. This model will help both buyers and sellers in the market. Buyers benefit from quick and accurate valuations of their prospective new homes, while sellers benefit from not having to obtain slow and costly manual valuations.

The second objective is to create another model that will predict the probable location of a property with features outlined by the user, these features will be similar to the features used in the price prediction model. This model will help buyers in the market narrow and concentrate their search for a home to areas where properties with their desired features are most likely to be or to come onto the market.

**1.2 Terms of Reference**

The terms of reference are as follows:

* Improve the predictive accuracy of the baseline price prediction model provided using regression, statistical techniques and by including text and image predictive features.
* Identify the most important predictive features of property value.
* Create an interface in which a user can input desirable property features and be returned the most probable location of properties with those features.
* Investigate how both products may be productionised and rolled out for wider use.

**1.3 Report Summary**

* Chapter 2 (Conclusions and Recommendations): Outlines the key conclusions and recommendations of the report.
* Chapter 3 (Literature Review): Consists of two literature reviews, the first details the different techniques and models used in property valuation, the second details some of the different approaches to predicting the location of properties based on independent features.
* Chapter 4 (Data Analysis and Preparation): Includes a profile of the dataset, exploratory data analysis and the data cleaning and preparation processes.
* Chapter 5 (Model Training, Testing, and Optimisation):
* Chapter 6 (Limitations and Further Work): Outlines any limitations to the work done in this project and areas of potential further research.

# **2) CONCLUSIONS AND RECOMMENDATIONS**

# **3 LITERATURE REVIEW**

This section contains a literature review on property valuation methods used in the past, the current most common methods, and new advanced methods entering the property valuation sector. Current valuation methods deviate by 10% on average from subsequent sale prices worldwide, this number is above the 10-year average of 9.3% and shows little signs of any improvement in valuation accuracy over the past decade (MSCI, 2021). This literature review will attempt to determine what causes inaccuracies in current methods, and whether the adoption of new technologies will improve the accuracy of property valuations in the future? The advantages and disadvantages of four different property valuation methods are analysed, discussed, and compared. This should inform the selection of different methods in varying use cases depending on the availability, quality, and quantity of data as well as the trade-off between method complexity and transparency.

**3.1 Property Valuation Methods and Algorithms**

Traditional Methods

Before the use of machine learning and advanced statistical techniques manual property valuations were the most prevalent method of predicting the final sale price of a property. Manual appraisers used a variety of valuation methods, the most common of which was the comparable sales approach. This method uses the sales price of similar properties recently sold in the same area, the appraiser then adjusts these prices based on differences between the sold properties and the valuation property to arrive at an estimated valuation (Pagourtzi, et al., 2003).

The comparable sales method has been dismissed however, as the sale of properties are never completely comparable. Two properties on the same street may have different levels of natural light or have had different modifications made by previous owners, and finally could be sold in different time periods under different market conditions (Kok, et al., 2017). To account for these differences appraisers may use their own judgement in adjusting the prices of 'comparables' to create an estimated property value, this introduces subjectivity into the appraisal and can result in differing valuations from appraisers of the same property.

House price indices (HPIs) are another method traditionally used before the introduction of advanced statistical methods. HPIs are time series that track the change of property values in a locality over a given period. HPIs rely on an initial previous value and regular revaluations of properties in the area with the same characteristics, changes in the values of these properties can be converted to a percentage that constitutes the HPI (European AVM Alliance, 2022).

HPIs are a quick and simple way to value a property as it is a simple exercise in multiplying the index by a previous valuation. There are however a few limitations to their usage. They rely on a previously estimation of value, if a previous estimate is unavailable, if the property was inherited rather than bought for example, there is no way to estimate the property's value by multiplying the previous estimate by the index. Additionally, HPIs are aggregations of time series data for a geographical locality and assume each property is identical, as a result their accuracy in valuing specific properties in the locality is severely limited.

Hedonic Models

Hedonic models are currently the most popular method in valuing properties. They are linear regression techniques that generate a coefficient value for property and locational characteristics to create a mathematical equation that produces a property valuation. Below is an example of a simple hedonic model equation.

The coefficient values illustrate each characteristic's contribution to the overall value of the property. The coefficient values and general regression equation are derived by training supervised machine learning algorithms with training data. The training data consists of property and locational characteristics as independent input variables and property sales prices as the dependent output variable. The accuracy of the hedonic model is determined by testing the equations performance against a testing dataset similar to the training set (European AVM Alliance, 2022).

Hedonic models share similarities with the traditional methods discussed above. Similar to the comparable sales method, value is given to characteristics of properties and differences between houses can be calculated using these value coefficients. Hedonic models are objective in determining a characteristic's coefficient value, while the comparable sales method may rely on an appraiser’s subjective opinion. This means a hedonic model will produce consistent results when valuing the same property, but this may not always be the case using the comparable sales method. Like HPIs hedonics are aggregate valuations and are not property specific, but hedonics are not limited by the necessity of a previous value like HPIs and build upon HPIs by offering more information about the overall value added by each unit of a property characteristic.

While hedonics may build upon traditional methods, that is not to say they are without limitations. Hedonics suffer from the same pitfall as HPIs, they can't value specific homes, their mathematical equations are standard for the location of the model limiting their accuracy in specific property valuation. Hedonics also assume that property characteristics such as surface area have a linear relationship with property value, which is not always the case, again limiting their accuracy (Nguyen & Cripps, 2001).

Artificial Neural Networks (ANNs)

Artificial neural networks consist of three components, an input layer, one or more hidden layers, and an output layer. Each layer is comprised of nodes that communicate with other nodes in the network mimicking the function of neurons in the human brain (IBM Cloud Education, 2020). The input layer receives external stimuli in the form of input variables and feeds them to nodes in the hidden layers. Nodes in the hidden layers act as non-linear processing units that assign each input a random weight, if the sum of the weighted inputs exceeds the node threshold it is passed on to the next layer in the network (Pagourtzi, et al., 2003; Wang & Jing-Li, 2019).

Unlike hedonic models ANNs are capable of handling non-linear relationships between variables. This allows them to better adapt to real world applications where linear relationships aren't guaranteed. In house price prediction it can't be assumed that a variable such as surface area will have a strictly linear relationship with the output variable price. ANNs are also more capable of handling messy data without as much need for data pre-processing as hedonic models (McCluskey, et al., 2012).

Though training the neural network seems a simple exercise, testing and interpreting their results seems to be a more strenuous task. Numerous studies have produced varied results, some with ANNs outperforming hedonic models, others with similar results or hedonics performing better. Amri & Tuluram (2012) find that ANNs outperform hedonic models and fuzzy logic techniques in a study of 7,849 unique transactions in Bathurst, Australia. Their ANN had an adjusted R-Squared of 0.45 while hedonics were sat at 0.37, only a slight edge. They also found that hedonic models could be improved by selective sampling of the data. Conversely McCluskey et al. (2012) and Lenk et al. (1997) found that ANNs performed no better, or worse than hedonic models, in studies of 2,694 and 288 properties respectively.

Sample size could be a factor in the varied results between different studies, as indicated by Nguyen & Cripps (2001) who found that ANNs outperformed hedonic models on medium to large datasets, but hedonics were better suited to small datasets. In small samples ANNs may learn the patterns inherent in the data too thoroughly and be adversely affected by outliers, while with larger datasets ANNs are better able to generalise than hedonics allowing for more accurate results.

Another limitation of ANNs is their interpretability, in the research conducted for this literature review ANNs were described as 'black boxes' in as many as 5 of the cited sources. The hidden layer of ANNs in which all computation and prediction is conducted is difficult to understand and explain for those without expert knowledge of their inner workings. As opposed to hedonic models that provide value coefficients for each input value, ANNs only output the final value offering little transparency as to how the valuation was derived (McCluskey, et al., 2012). In a commercial setting this could hamper ANNs practical use as appraisers may often have to explain their valuation criteria and process.

Some efforts have been made to offer greater transparency to ANNs through methods such as 'local interpretable model-agnostic explanations' (LIME) or Shapley Additive Explanations. LIME is a method in which the original input data is altered slightly and AI generated target values are used to train a hedonic model that can explain each property characteristic's contribution to the final valuation. There are limitations to the LIME method however, discrete variables in the input data can cause issues and deciding on how much to alter the original data to explain each individual valuation can prove difficult and time consuming (Gloudemans & Sanderson, 2021).

Another possible solution is Shapley values. To generate a Shapley value for a particular characteristic of a property, a random property is generated with identical values for each characteristic except our variable of interest, this value is randomised. The order of the variables is then randomised to account for variable interaction and fed into the model for both the subject and randomly generated property, then the difference between their valuations is calculated. This process is done repeatedly for a specified number of iterations. The mean of the differences in valuations is the Shapley value for that variable. Shapley values are helpful in explaining the contribution of a variable to the overall valuation however deriving them for each input variable can be time consuming and computationally expensive and thus limits their use in practice (Gloudemans & Sanderson, 2021).

Though ANNs can outperform traditional and hedonic methods, their commercial use is likely to remain limited as a result of their reputation as 'black boxes' and their need for large datasets.

Ensemble Methods

Ensemble methods combine simple individual models to generate a more accurate generalised model. Ensemble models can be created through numerous different techniques, but the two most commonly used methods are bagging and boosting (Diettrich, 2017).

Bagging or bootstrap aggregating generates a series of independent samples of the same size and distribution as the base dataset. These are then used to train simple models and the output for the predictors are combined through averaging for regression or majority voting for classification. This generates an ensemble output that one would expect to be more accurate than the individual outputs within the the ensemble (Ganaiea, et al., 2022). An example of a bagging algorithm is the Random Forest Algorithm, bagging is applied decision trees to create an ensemble output or forest with less variance and improved accuracy over a single decision tree.

Boosting algorithms create ensembles by iteratively building simple models and attempting to reduce an error metric (e.g., residual mean squared error) each iteration by penalising models with larger errors. The ensemble model is a sum of the weighted models, with more weight being given to base models of higher accuracy. Boosting algorithms generally use regression and classification trees much the same as the Random Forest Algorithm (Kok, et al., 2017).

Ensemble methods have gradually entered the property valuation space with numerous studies conducted in the past few years. Kok et al. (2017) compare Random Forest and two boosting models, Gradient Boosting and eXtreme Gradient Boosting (XGBoost), with a hedonic model and see positive results with both ensemble methods significantly outperforming the hedonic model on a dataset of 5,018 sales transactions. Both boosting models outperform the Random Forest model in each testing metric. Another interesting finding in this study, the importance of locational characteristics over property-specifics, illustrates another advantage of ensemble methods. The study outlines the importance of variables in the valuation process, something that ANNs were not able to provide and hampered their understandability. Boosting methods are also much more robust to low quality data, capable of handling missing data, categorical data, and outliers without much need for data pre-processing or cleaning (McCluskey, et al., 2014).

Although ensemble methods are relatively new in the property valuation space there seem to be multiple advantages to their use. Their capability to handle low quality data without much need for preparation or specifying an algorithm to use gift ensemble methods the same adaptability as ANNs. While the easily interpretable decision trees that ensembles are comprised of and the ability to determine the relative importance of variables in the property price prediction offer the same level of transparency as hedonic models.

Conclusions

The current gap between valuations and final sales prices as stated by MSCI (2020) could be linked to any of the downfalls in currently used valuation methods; subjectivity in manual valuations, inability to value specific properties, non-linear relationships between predictors and target variables etc. but pinpointing which are most problematic is difficult and more research is necessary.

Artificial neural networks show promising results in terms of improving the accuracy of valuations as they can handle non-linearity and messier data but fall short in providing transparency in their valuations. This could limit their use in commercial settings where transparency may be a priority, although the use of LIME or Shapley values could help to provide limited assistance in improving their transparency and overall understandability.

Ensemble methods show good promise in providing accurate results as well as transparency and interpretability through variable importance and decision tree visualisation. They are relatively new in the space however and more extensive research into their application to the property valuation sector is needed before they can conclusively bridge the gap between valuations and sales prices.

# **4) DATA ANALYSIS AND PREPARATION**

This chapter contains a data profile and quality assessment, exploratory data analysis, and details the data cleaning and pre-processing procedure.

**4.1) Data Profile**

Data Description

The dataset consists of 2,982 observations and 16 variables for property sale ads placed in Dublin, Ireland. The dataset is made up of 7 numerical (3 discrete and 4 continuous), 6 categorical (1 ordinal and 5 nominal) and 3 unstructured variables. A description of each variable is given in APPENDIX A.

Data Quality Assessment and Cleaning

"High-quality datasets are essential for developing machine learning models." (Gudivada, et al., 2017)

The quality of the data that is fed to machine learning models directly impacts the efficiency of the models and the accuracy of their output. Prior to training a model it is important to assess the quality of the data and improve the dataset through data cleaning processes (Gudivada, et al., 2017).

Typical data quality assessments applied to organisations evaluate the data across six dimensions, timeliness, completeness, uniqueness, validity, consistency, and accuracy (CDC, n.d.). As the dataset is static and there is no data warehouse or pipeline, the data quality assessment in this report will consider only the following dimensions, completeness, uniqueness, and validity.

Completeness refers to the number of missing data points in the dataset. Analysing the missingness patterns of data points allows us to assess whether the absence of a value is indicative of an absence of a property feature or whether it constitutes a null value. This will help us determine how best to handle missing data. There are missing values in 8 of the 16 variables in the datasets, with 4 variables missing more than 10% of their values. The completeness of each variable is shown in APPENDIX B.1. To assess the type of missingness present in the data, the missingness pattern heatmap in APPENDIX B.2 was created, this visualises the correlation between the presence of a value for one variable and the presence of a value for another.

It is evident from the heatmap that the presence of 'no\_of\_units' negatively impacts the presence of values in other variables. The 'no\_of\_units' variable is linked to 59 sparsely populated new development ads in the data.

Uniqueness measures the number of duplicated rows in the dataset. Thankfully there were no duplicates present in the data. There were however some variables, 'county' and 'environment', that were constant across all observations.

Validity is the extent to which variables conform to a given format. Checks were made for the presence of any special characters present in the categorical data, any non-numerical values in the numeric data, and for any other aberrations from the format of the structured data.

Overall, the quality of the dataset is good, there are some missing values present in the data, but these are handled by removing sparse observations and dropping variables that fall below an acceptable completeness threshold. The absence of duplicates or incorrectly formatted data also enhance the quality of the dataset.

**4.2) Exploratory Data Analysis**

Relationships between variables

Correlation matrices and variance inflation factors (VIF) were used to identify highly correlated variables and assess whether any variable redundancy exists resulting from information overlap. The correlation matrices and variance inflation factors are shown in APPENDIX C. Analysing solely the numerical values, strong correlation exists between 'surface' and 'beds', this is expected as more bedrooms generally equates to more surface area. The VIF for both variables also indicate a moderate correlation between the two variables. It was decided that neither variable would be removed as their correlations with other variables were not strong and both provide valuable information about a property.

When all variables, categorical and numerical, are analysed we see a strong correlation between 'area' and both geographical coordinate values 'latitude' and 'longitude', again this is an expected relationship but not one that is seen as problematic to our either of our models.

Unstructured Data

Unstructured data is qualitative data that does not follow a set format such as text, images, audio etc. as such it can't be processed by conventional tools and models (IBM Cloud Education, 2021). The unstructured data in the dataset is free text entered by real estate agents describing each property and its features.

Natural language processing (NLP) tools were used to pre-process and analyse the unstructured text data. The text data is by nature noisy and contains many words that won't provide any useful information to the models, the aim of natural language processing in this project is to eliminate any noise and extract informative features that will boost model accuracy.

Points of interest are the frequency of different words and word types in property descriptions. The most frequently used words are shown in the word cloud above. Different word types may also be informative to the models, nouns for example may indicate the presence of a property feature, a garage for example. Meanwhile different adjectives may be used to describe properties of a certain price bracket or location.

|  |  |  |
| --- | --- | --- |
|  | **Mean Count** | **Mean Length** |
| **All Words** | 397.49 | 6.40 |
| **Nouns** | 157.72 | 6.37 |
| **Adjectives** | 42.14 | 6.31 |
|  |  |  |
|  | **Top 25% (Price)** | **Bottom 25% (Price)** |
| **Mean Noun Count** | 223.73 | 113.49 |
| **Mean Adjective Count** | 62.29 | 28.62 |
| **Mean Noun Length** | 6.03 | 6.21 |
| **Mean Adjective Length** | 6.29 | 6.09 |

Furthermore, the number of nouns or adjectives could be informative to the models, more nouns may mean more features, more adjectives may indicate a nicer property or location. The average length of words was also analysed, longer adjectives could potentially be used in more expensive property descriptions, for example 'pristine' condition in contrast with 'good' condition. Analysis of these features can be found in the tables below.

Outlier detection and removal

Outliers were detected firstly by visualising the distribution of each numerical variable to identify any skewing. These distributions can be found in APPENDIX D. Positive skew was identified in five of the six numerical variables indicating the presence of 'high outliers' that are greater than most other observations in the dataset. Negative skew is identified in the final variable indicating 'low outliers' that fall below the level of other observations.

Noting the skewed nature of the variables, it was decided that in the absence of normal distributions, using the inter-quartile range (IQR) would be a more suitable method than Z-scores to numerically identify outliers and remove them from the dataset. The formulae below were used to identify both types of outliers, where Q1 is the first quartile and Q3 is the third.

The number of outliers for each variable is shown in the table below. These were removed from the dataset leaving a final number of 2,458 observations.

|  |  |
| --- | --- |
| **Variable** | **Outliers** |
| bathrooms | 25 |
| beds | 17 |
| latitude | 168 |
| longitude | 5 |
| price | 221 |
| surface | 203 |
| **Total** | **465** |

**4.3) Data Pre-processing**

Before training the models with the dataset, the data needs to be pre-processed. This is done in two stages, the first stage handles the structured data, the second the unstructured data.

Structured Data

After cleaning the data by removing any outliers or sparsely populated observations or variables, the data is sent for pre-processing. Pre-processing ensures that the data is in a format that is ingestible to the models. This involves removing any sparsely populated rows and columns, imputing any remaining missing values, and transforming and encoding the data where needed.

The sparsely populated new development ads were dropped from the training dataset as were the 'no\_of\_units' and 'property\_category' variables. This improved completeness but the 'facility' variable still fell below an acceptable completeness threshold (75%) and was also dropped. The remaining missing numeric values found in the 'price' and 'surface' variables were imputed to have a value of 0. This may seem counterintuitive, as a property can't have a surface area of 0 square metres and generally don't sell for free, however modern machine learning algorithms such as XGBoost recognise these 0 values as missing and treat them accordingly.

The missing categorical values ('ber\_classification') are handled in the labelling process. Many regression algorithms don't accept categorical values in their original form as such they must be encoded. Two common methods exist, one-hot encoding and label encoding. One-hot encoding decomposes categorical variables into binary indicator variables or dummy variables. Using one-hot encoding would increase the dimensionality of the dataset by 171 sparsely populated variables and as such it was decided that labelling would be a more appropriate solution. Label encoding assigns the unique values of each variable an integer value. This does not increase the dimensionality of the dataset. To further reduce dimensionality any constant variables with no variance ('county' & 'environment') were dropped from the dataset.

Finally, to improve predictive accuracy for the price predictive model a log transform is made to the independent variable 'price'. This is done as the distribution of 'price' is positively skewed, by using a log-transform this distribution (see APPENDIX E) more closely resembles a normal distribution, additionally the variance of the independent variable is decreased this allows for improved predictive accuracy.

Unstructured Data

The 'description\_block' and 'features' columns are combined to create a consolidated unstructured column, named 'desc\_feat', containing all the property's descriptive information. This column is then cleaned. All words are converted to lower case and any special characters, punctuation, and numbers are removed. Any 'stop words' are then removed from the text, stop words are commonly used words that don't add any information to the text, for example 'the', 'a', 'I' etc. Any remaining words with less than 3 characters are removed, these were found to be typos in most cases. Finally, any additional whitespace is trimmed from the text.

The cleaned data is then tokenised. Tokenisation is the process of splitting a sentence into individual words and storing them in an array.

**"this is a new property"** becomes **["this", 'is", "a", "new", "property"]**.

The tokenised text then goes through a process called "Part of Speech (POS) Tagging". This assigns a tag to each tokenised word, the tag indicates the "part of speech" (noun, verb, adjective, adverb, etc.) of the tokenised word. Using the POS tags, new columns containing only words of certain POS can be created. For the purposes of this project new columns are created only for nouns and adjectives. Additional numerical columns for the count and average length of nouns and adjectives are created.

Each of the unstructured columns ('desc\_feat', 'nouns', and 'adjectives') is then lemmatised. Lemmatisation is the process of "removing inflectional endings only and returning the base or dictionary form of a word" (Manning, et al., 2008). By removing inflections, "bats" becomes "bat" and "boy's" or "boys' " become "boy" etc. Lemmatisation will also convert more complex cases, for example past tense "bought" will become "buy".

Lemmatisation is helpful in reducing the dimensionality of 'bag of words' matrices, these are binary dummy matrices that indicate the presence of a word or phrase in a text. By removing inflections, lemmatisation ensures there won't be separate matrix columns for 'doors' and 'door' for example. Bag of words matrices were made for each lemmatised unstructured column and are used to create informative features to train the models and improve their accuracy.

To extract the most informative features analysis on word frequency is conducted. As seen in the word cloud earlier the most frequently used words ('kitchen, 'bedroom', 'bathroom' etc.) are not very informative, most if not every property will have these features. The least used words are also uninformative as they are often typos or property specific. As such words with moderate frequencies are seen as potentially the most informative.

The dataset was subset into 3 groups, the top 25% most expensive properties, the middle 50%, and the bottom 25%. Words that featured in 30% to 60% of these properties’ descriptions were extracted as keywords and their corresponding columns in the bag of words matrices were added to the dataset. Any duplicated columns were removed.

A limitation to the bag of words approach is that it disregards the context or order of the words, for nouns and adjectives this is not a major issue, however when taking the whole descriptions this can cause some very noisy data to enter the bag of words. One solution to this is to use bigrams, bigrams are consecutively written words in the description, this helps add to the context and preserve the order of words in the dataset. Bigrams are used in the bag of words matrix for the lemmatised description.

Final Dataset Description

The final dataset consists of 2,458 observations and 157 variables. 3 categorical variables, 13 numerical variables, and 141 binary dummy variables that indicate the presence of a word in the property's description or features.

**5) MODEL SELECTION, TRAINING AND TESTING**

This chapter details the model selection, training, and testing processes. It also contains empirical results and analysis.

**5.1) Price Prediction Model**

Model Selection

The baseline model provided in the project brief used a hedonic OLS regression, use of this model is continued to provide a baseline for more advanced modelling methods. The selection of advanced modelling methods was informed by the project requirements and findings from the literature review. Interpretability was seen as an important factor in the model selection process given the requirement to output the predictive importance of variables. Given ANNs 'black-box' reputation and inability to produce variable importance, it was decided that it would be more appropriate to use ensemble methods in property price predictions.

Two ensemble methods were discussed in the literature review, Random Forest with its use of bagging, and XGBoost and its use of boosting. Both models are trained on the dataset, and their results compared to one another as well as the baseline hedonic model. The same training dataset is used by all models.

Training, Testing, and Optimisation

Prior to model training the pre-processed unstructured data was split into two subsets. These subsets contained the unstructured variables for the most expensive and cheapest 25% of ads. The extracted nouns, adjectives and lemmatised bigrams for these subsets were then merged with the structured data to create the final dataset for the models, the count and average length of each word-type were also included. The final dataset contained 2,458 observations and 122 variables. 3 categorical variables, 13 numerical variables, and 106 binary dummy variables that indicate the presence of a word in the property's description or features. The dummy variables consist of 69 nouns, 15 adjectives and 22 lemmatised bigrams.

The dataset was split into training and a testing sets in order to evaluate the accuracy of the models. 80% of the data was used to train the models with the other 20% being used to test the accuracy of the models. Three metrics were used to assess model accuracy Median Absolute Percentage Error (MdAPE), Mean Absolute Error (MAE), and R-Squared. MdAPE and MAE will give us an indication as to how precise the model is, giving us a percentage and real number value. R-Squared indicates the variance in price that is explained by the independent variables and will give us an insight into the importance of different variables and the goodness of fit of the models.

The two ensemble methods, XGBoost and Random Forest, ingest several 'hyperparameters' in addition to the training data to tune their performance and correct overfitting. There are a number of ways to tune these hyperparameters, these include brute force methods such as 'GridSearch' or Bayesian optimisation methods using 'HyperOpt'. Bayesian methods have been shown to be faster and more accurate tuning models in practice and as such 'HyperOpt' was used to tune both ensemble methods.

Results

Two objectives were kept in mind when creating the price prediction models. The first was to provide the most accurate price prediction. The second objective was to assess the predictive importance of variables, specifically variables derived from the unstructured data. To accomplish both these objectives, the final dataset was incrementally fed to each of the three models. The incremental training process is outlined with the help of the XGBoost predictive results table below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MdAPE** | **MAE** | **R-Squared** |
| **Structured (Base)** | 9.75 | 53415.80 | 0.823 |
| **Base + Nouns** | 11.52 | 61009.86 | 0.772 |
| **Base + Adjectives** | 11.11 | 64287.03 | 0.741 |
| **Base + Lemmas** | 11.07 | 56941.40 | 0.805 |
| **Base + Nouns + Adjectives** | 11.88 | 63700.74 | 0.771 |
| **Base + Nouns + Lemmas** | **8.69** | 47365.43 | **0.867** |
| **Base + Adjectives + Lemmas** | 9.26 | 51362.08 | 0.745 |
| **Base + All Word-Types** | 8.85 | **47202.34** | 0.864 |
| **Base + Counts + Lengths** | 9.45 | 51336.54 | 0.834 |
| **Base + All + Counts** | 9.76 | 51958.86 | 0.841 |

Firstly, structured data was used to train the model, unstructured variables are then incrementally added. This allows us to assess the predictive importance of nouns as compared to adjectives and so on.

Chart, scatter chart

Description automatically generatedXGBoost performs the best overall providing the most accurate predictions and outperforming the OLS and Random Forest models in every training iteration. The accuracy results for all three models can be compared in APPENDIX X. The best XGBoost model is highlighted above and consists of structured data, nouns, and lemmas. In the plot below we can see the predicted price as compared to the actual price for this model.

We can see that in the region from €200,000 to €400,000 that there is reasonably good fit with many datapoints centred around the 45-degree line. As the price rises however there is more varied results, and this can be seen when assessing the predictive accuracy by price range as seen in APPENDIX X. We can see that as the price rises the number of training rows starts to decrease meaning the model can't predict as accurately. Overall, this model provides reasonable accuracy and would be expected to improve if given a larger dataset to train on.

Importance of Variables

Table

Description automatically generatedIn completing the second objective, assessing the variable importance, we look firstly at the importance of different word-types. We can see from the comparable results in APPENDIX X, that the word-types add and sometimes detract from the model in varying degrees. Of the three word-types adjectives add the least to the model in isolation, improving the MAE and R-Squared of the OLS model only slightly and negatively impacting the XGBoost and Random Forest models across all metrics. Lemmas improve the OLS and Random Forest models across all metrics, but nouns are slightly better at improving model performance. Interestingly adding any word-type in isolation to XGBoost negatively impacts performance as compared to the base dataset with only structured variables. Nouns and Lemmas are the best combination and produce the best overall model. Including all word-types can also improve performance and produced the best Random Forest model and resulted in the lowest MAE in all three modelling algorithms.

After assessing the impact of different word-types a more granular approach is taken to assess the most important variables overall. The variable importance for the best model (XGBoost with Nouns + Lemmas) is shown in APPENDIX X. We can see that the structured data is more impactful than any of the variables derived from the unstructured descriptions. Unsurprisingly the geographical variables 'latitude' and 'longitude' are the most important predictors of property value in this model, echoing the "Location x 3" trope in real estate. Among the unstructured variables nouns have greater representation than lemmatised bigrams in the most important features supporting the table results in displaying their greater predictive power. \*Prefixes are added to unstructured variables in graphing to indicate word-type

By plotting the decision tree behind the XGBoost model in APPENDIX X, we can see the decision path of the model and how it derives a final predicted price for a property and the impact variables play in the prediction process. An example is highlighted by the black box, here we can see that where 'N\_wardrobes' < 0.5 (i.e., not present in the ad description) the final leaf node values are lower than those where 'N\_wardrobes' is present. This gives us extra insight into 'N\_wardrobes' relationship with price, its inclusion in the description can be important in prediction.

**5.2) Location Prediction Model (Classification)**

The price prediction model is an exercise in regression, predicting a number on a continuous scale. The location prediction model is an exercise in classification, predicting a nominal category or class associated with an observation, in this case the class is the property area. There are different requirements and constraints to both prediction exercises. In this classification task, the biggest constraint is the size of the dataset as compared with the number of classes. There are 2,458 observations in the final dataset, and 144 distinct areas. This large number of areas poses a problem, there are many areas with few observations. As we saw in the regression exercise the accuracy decreased when price rose as there were less training observations available. 45% of the areas have less than 10 observations, meaning the models will have too few training observations to produce an accurate output for many areas. To further illustrate this, three models were trained with the current dataset, a multinomial logistic regression model, an XGBoost Classifier, and a Random Forest Classifier, their respective accuracy scores were: 4.2%, 6.43%, and 12.6%.

To circumvent this problem two alternative approaches were taken. The first approach clusters the observations using latitude and longitude to create aggregated areas and reduce the number of classes. K-means clustering will be used to cluster the data and multinomial logistic regression, XGBoost classifier, and Random Forest classifier models will be used to predict the area cluster. The second approach changes this problem from a classification task to a multi-output regression task and attempts to predict both the latitude and longitude of properties. A Random Forest regressor will be used to perform the regression approach as it inherently supports multiple outputs.

Training, Testing, and Optimisation

In the price prediction model, the dataset was derived by extracting the nouns, adjectives, and lemmatised words from the top and bottom 25% of properties in terms of price. The location prediction dataset consists of the structured variables, nouns, adjectives, and lemmatised words/bigrams that occur in 30-60% of *all* observations. This is done as it would be extremely time consuming to create different bags of words for each area in the classes and would be difficult to apply in practice to a multi-output regression model for latitude and longitude.

The regression model will make use of the same metrics as used for the price prediction model, MdAPE, MAE, and R-Squared. The clustered approach will use the classification accuracy score as its measure of accuracy. The data will be split into training (80%) and testing (20%) data in the same manner as before.

All ensemble methods, XGBoost and Random Forest classifiers and regressors will be tuned using 'HyperOpt' as in the price prediction model.

Predictive Results (Cluster Approach)

Clustered aggregates of areas were created with number of clusters ranging from two to twelve. The optimum number of clusters using KneeLocator was 4, however more granularity would be required of a commercially viable product, and 12 was decided to be the maximum number of clusters allowed. Plots of the clusters can be seen in APPENDIX X. Different areas of Dublin can be made out in the plots with higher numbers of clusters, Dun Laoghaire and the coastal south-east, inner-city Dublin, Howth in the north-east, etc.

The accuracy scores of all three clustered models are shown in the table below. Across all three models the predictive accuracy falls as the number of clusters rises. This is a result of training observations being spread more thinly across a larger number of classes in 12\_clusters as compared with 2\_clusters for example.

|  |  |  |  |
| --- | --- | --- | --- |
| n\_clusters | Multinomial Logistic Regression | XGBoost | Random Forest |
| 2\_clusters | 72.07% | 79.06% | 75.15% |
| 3\_clusters | 58.26% | 69.40% | 58.93% |
| 4\_clusters | 57.70% | 66.74% | 54.62% |
| 5\_clusters | 49.50% | 63.04% | 50.72% |
| 6\_clusters | 48.51% | 62.22% | 51.33% |
| 7\_clusters | 46.20% | 59.55% | 46.41% |
| 8\_clusters | 42.94% | 57.29% | 46.00% |
| 9\_clusters | 37.04% | 49.49% | 40.86% |
| 10\_clusters | 42.10% | 51.13% | 41.27% |
| 11\_clusters | 39.00% | 50.51% | 38.81% |
| 12\_clusters | 38.12% | 54.00% | 37.58% |

Again, we see that XGBoost is the best performing model across all iterations, with Multinomial Logistic Regression and Random Forest performing similarly. When deciding the optimal model to use users of this approach face a trade-off decision between granularity and accuracy. The model that will be used to interpret the importance of variables is XGBoost with 6 clusters, this model offers reasonable granularity and is reasonably accurate.

Importance of Variables (Cluster Approach)

Table

Description automatically generated

Like the price prediction model, we can see that the structured variables provide more predictive power than the variables derived from descriptions. This feature importance is almost the inverse of that of the pricing model, where locational variables (latitude and longitude) were the most important features in predicting price, here price is the most important variable in predicting location. Some other interesting variables in the most important features are 'L\_city\_centre' rather intuitively it would indicate a property close to the city centre, 'L\_luas' would indicate proximity to the Luas, Dublin's tramline system. 'N\_view' also features, this could indicate a coastal property with 'sea views' perhaps. As compared with the pricing model, the unstructured variables are not as overshadowed by the importance of structured data in predicting location. This could suggest that descriptions are more powerful for predicting the location of a property than the property price.

Predictive Results (Regression Approach)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MdAPE** | **MAE** | **R-Squared** |
| **Latitude** | 0.04 | 0.0311 | 0.35872938 |
| **Longitude** | 0.61 | 0.0517 | 0.43027817 |

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedThe results for the multioutput Random Forest model in the table above look promising on the surface, with low MdAPE and MAE suggesting small error. However, when we put this in the context of a low variance target variable like latitude latitude these are large errors that could place the property kilometres away from its actual location. Visualising the output with scatterplots offers a true reflection on the accuracy of the model.

From the above plots we can see that the model is a poor fit for both outputs as datapoints deviate from the line of best fit. This is supported by the R-squared values for both latitude and longitude.

Graphical user interface, table

Description automatically generatedGraphical user interface, application, table, Excel

Description automatically generatedImportance of Variables (Regression Approach)

We can see that both outputs have differing variables of importance. Price features prominently in both as we might expect given its relationship to locational variables shown in the previous models. Again, we see that L\_luas and L\_city centre are important predictors of both latitude and longitude. The variables derived from the unstructured descriptions are of greater importance in this model similar to the cluster approach. This may further support the claim that descriptions are more useful in location prediction than price prediction.

# **6) LIMITATIONS AND FURTHER WORK**

# **APPENDICES**

Appendix A: Description of variables in the dataset.

Table

Description automatically generated

Appendix B: Completeness of each variable.

Table

Description automatically generated

Chart, histogram

Description automatically generatedAppendix B.2: Missingness Pattern Heatmap

Appendix C:

C.1.: Correlation Matrix for Numeric Variables.

Graphical user interface, chart, treemap chart

Description automatically generated

Table

Description automatically generatedC.2.: Variance Inflation Factors for Numeric Variables.

C.3.: Strength of Association Matrix for All Variables.

Chart, treemap chart

Description automatically generated

Graphical user interface

Description automatically generatedAppendix D: Distributions of numeric variables

Appendix E: Price Distribution vs. Log Price Distribution

Chart, histogram

Description automatically generated

**OLS Price Prediction Accuracy**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MdAPE** | **MAE** | **R-Squared** |
| **Structured (Base)** | 17.07 | 86666.57 | 0.578 |
| **Base + Nouns** | 14.82 | 77208.55 | 0.674 |
| **Base + Adjectives** | 17.66 | 86526.70 | 0.586 |
| **Base + Lemmas** | 14.68 | 78407.18 | 0.643 |
| **Base + Nouns + Adjectives** | 14.76 | 77150.28 | 0.671 |
| **Base + Nouns + Lemmas** | 14.47 | 72638.56 | **0.713** |
| **Base + Adjectives + Lemmas** | 15.22 | 78854.16 | 0.646 |
| **Base + All Word-Types** | 14.42 | **72415.30** | 0.708 |
| **Base + Counts + Lengths** | 16.61 | 83522.98 | 0.604 |
| **Base + All Variables** | **14.22** | 72541.53 | 0.710 |

**XGBoost Price Prediction Accuracy**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MdAPE** | **MAE** | **R-Squared** |
| **Structured (Base)** | 9.75 | 53415.80 | 0.823 |
| **Base + Nouns** | 11.52 | 61009.86 | 0.772 |
| **Base + Adjectives** | 11.11 | 64287.03 | 0.741 |
| **Base + Lemmas** | 11.07 | 56941.40 | 0.805 |
| **Base + Nouns + Adjectives** | 11.88 | 63700.74 | 0.771 |
| **Base + Nouns + Lemmas** | **8.69** | 47365.43 | **0.867** |
| **Base + Adjectives + Lemmas** | 9.26 | 51362.08 | 0.745 |
| **Base + All Word-Types** | 8.85 | **47202.34** | 0.864 |
| **Base + Counts + Lengths** | 9.45 | 51336.54 | 0.834 |
| **Base + All Variables** | 9.76 | 51958.86 | 0.841 |

**Random Forest Price Prediction Results**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MdAPE** | **MAE** | **R-Squared** |
| **Structured (Base)** | 14.18 | 69,051.50 | 0.738 |
| **Base + Nouns** | 13.96 | 68901.31 | 0.741 |
| **Base + Adjectives** | 14.32 | 69166.95 | 0.736 |
| **Base + Lemmas** | 14.02 | 68880.99 | 0.739 |
| **Base + Nouns + Adjectives** | 13.89 | 68778.86 | 0.743 |
| **Base + Nouns + Lemmas** | 14.05 | 68718.59 | 0.743 |
| **Base + Adjectives + Lemmas** | 14.12 | 68985.94 | 0.739 |
| **Base + All Word-Types** | 14.31 | **68630.96** | **0.745** |
| **Base + Counts + Lengths\*** | 13.89 | 69361.15 | 0.735 |
| **Base + All Variables** | **13.84** | 69226.73 | 0.739 |

\* Counts + Lengths: Counts are the number of each word-type in a description (e.g., count of nouns), Lengths are the average length of word-types in description (e.g., average length of adjectives).

**Accuracy vs. Price Range**

|  |  |  |  |
| --- | --- | --- | --- |
| **Price Range** | **MdAPE** | **# Testing Rows** | **# Training Rows** |
| **(50000, 100000]** | 215.58 | 1 | 0 |
| **(100000, 150000]** | 37.29 | 1 | 11 |
| **(150000, 200000]** | 11.03 | 30 | 104 |
| **(200000, 250000]** | 11.93 | 70 | 236 |
| **(250000, 300000]** | 10.12 | 74 | 306 |
| **(300000, 350000]** | 8.96 | 57 | 273 |
| **(350000, 400000]** | 8.46 | 61 | 234 |
| **(400000, 450000]** | 9.52 | 34 | 152 |
| **(450000, 500000]** | 8.5 | 38 | 159 |
| **(500000, 600000]** | 10.9 | 51 | 195 |
| **(600000, 700000]** | 7.81 | 25 | 87 |
| **(700000, 800000]** | 10.88 | 22 | 102 |
| **(800000, 1000000]** | 15.39 | 23 | 88 |

**Diagram

Description automatically generatedXGBoost Price Prediction Decision Tree**

Diagram, calendar

Description automatically generated**Cluster Plots**