UTS: ENGINEERING & INFORMATION TECHNOLOGY

SUBJECT NUMBER & NAME	NAME OF STUDENT(s) (PRINT CLEARLY) *SURNAME* *FIRST NAME*	STUDENT ID(s)
41040 Introduction to Artificial Intelligence	Jordan Stojcevski Imogen Turner Andreas Gwyther-Gouriotis Navid Choudhury	13917707 13273267 13893627 13753037

STUDENT EMAIL

imogen.turner@student.uts.edu.au
andreas.gwythergouriotis@student.uts.edu.au
jordan.stojcevski@student.uts.edu.au
navid.a.choudhury@student.uts.edu.au

NAME OF TUTOR	TUTORIAL GROUP	DUE DATE
Hai Yan (Helen) Lu Shannan Guan Jose Luiz Meza Garcia	8	**08/11/2021 11:59pm,**

ASSESSMENT ITEM NUMBER & TITLE

Assessment Task 2: AI Project

☑ I confirm that I have read, understood and followed the guidelines for assignment submission and presentation on page 2 of this cover sheet.

☑ I confirm that I have read, understood and followed the advice in the Subject Outline about assessment requirements.

☑ I understand that if this assignment is submitted after the due date it may incur a penalty for lateness unless I have previously had an extension of time approved and have attached the written confirmation of this extension.

Declaration of originality: The work contained in this assignment, other than that specifically attributed to another source, is that of the author(s) and has not been previously submitted for assessment. I understand that, should this declaration be found to be false, disciplinary action could be taken and penalties imposed in accordance with University policy and rules. In the statement below, I have indicated the extent to which I have collaborated with others, whom I have named.

Signature of student(s) :

Andreas Gwyther-Gouriotis, Jordan Stojcevski, Navid Choudhury, Imogen Turner

Date 7/11//2021

41040 Introduction to Artificial Intelligence 2021 - University of Technology Sydney

Assignment 2: Using Artificial Intelligence to Predict Property Prices in NSW

By: Imogen Turner (13273267), Jordan Stojcevski (13917707), Andreas Gwyther-Gouriotis(13893627), Navid Choudhury (1375037)

Introduction

In the real estate market, a crucial task for real estate agents is to formulate accurate property valuations, to identify the approximate value a property is expected to sell for. This formulation is based on a variety of features, including but not limited to property size, location, condition, level of convenience and recent sales in the same area (Castle, 2017). Precise property valuations greatly benefit both the buyer and seller of a property. For a purchaser, a property valuation eradicates the risk of purchasing a property for more than its market value. Additionally, a property valuation identifies an approximate estimation of the property sale price, solidifying expectations and ensuring a property is not sold for less than what it is worth. Research shows that methods to predict property prices have had historical issues with accuracy (Chaphalkar & Sandbhor, 2013). The traditional methods of a property valuation rely on a certified valuer entering the property and assessing the variety of features the property possesses, as well as comparing the property to recent sales in the area. However, this method has proven to fail to identify important latent relations that machine learning and advanced property data analysis has to offer ("How Artificial Intelligence is Shaping the Real Estate Market", 2021). Additionally, the use of human-based appraisals is susceptible to human bias, discrimination and human errors (Neal et al., 2020).

The use of automated valuation models (AVM) on property data has been used widely in financial institutions and for property valuations. This method relies on mathematical and statistical databases containing comparable properties to calculate a property's value at a specific time point. Current issues lie within the AVM model as their performance metrics are not universally specified or consistently estimated. It has been acknowledged through published literature that the AVMs presently used do not deliver accuracy scores that meet accepted scientific standards. (Ecker et al., 2020). The poor performance of these implemented AVMs is highlighted in findings of the American online real estate marketplace, Zillow. They announced that Zillow is utilising Artificial Intelligence to gain over 15% more accuracy on their property valuations, when compared to traditional AVM methods. This acts as a testament to prove the objectively poor performance of currently implemented valuation methods (Hinchliffe, 2018). Based on these findings, we concluded that artificial intelligence adoption is no longer a choice in today's environment of rapid technological progress. It is more of a competitive need. Valuation models that do not use any amount of AI will become obsolete in the near future. Thus, Artificial Intelligence is an integral solution for property valuations as it minimises human error, bias and betters the accuracy when compared to other current widely used methods.

Therefore, the aim of this project is to use **Artificial Intelligence to predict property prices in New South Wales.** The data utilised for this project is sourced from the publicly available NSW Valuer General's office ("Bulk Property Sales Information", 2021), which details bulk NSW Property Sales Information from 1990 to present day. We will be focusing on data from 2001 to 2014. The relevant attributes contained in these datasets will be used, consisting of:

1. District Code
2. Property ID
3. Sale Counter
4. Property Unit/House Number
5. Street Name (string and numerical ID)
6. Locality (string and numerical ID)
7. Postcode
8. Area
9. Area Type
10. Contract date
11. Purchase Price
12. Zoning (string and numerical ID)
13. Primary Purpose (string and numerical ID)
14. Component Code (string and numerical ID)
15. Sale Code

The system used was composed of four sections, all developed using Python and Google Colaboratory. Firstly, the data was collected using a web scraper. It was then pre-processed to extract relevant features and create additional statistical attributes. The preprocessed data was then split into training and testing datasets and the data was trained accordingly. Lastly, the testing set was implemented on a variety of machine learning algorithms. To ensure the predictive value is a continuous output that reflects a property price, we used regression models. The models explored in this project were, Random Forest Regressor, ADABoost Regressor, Support Vector Regressor, Decision Tree Regressor, Neural Network and Gradient Boosted Regressor. The best solution was identified through evaluating each algorithm's result with respect to absolute mean error, mean error range and R-squared percentage.

Project Solution Design

The project problem was to design an AI model that could predict house prices based on different data attributes from the NSW government property valuation data. In order to solve this problem the first step was to design a solution that could be implemented in code in order to produce a working AI model. The project solution design had the following sections, the data collection section which involved gathering information from the NSW government site which has valuations on properties dating back to 1990. To gather information a web scraper was developed that clicked on each of the years and downloaded the files of all the property valuations for each year totaling well over a hundred thousand files and over a few million rows of data. Due to the huge amounts of data the solution was designed in order to break up the data into different ranges of years such as 1990-2000 and then generating a dataframe for each of those years so that they can be processed using packages such as pandas and so it can train different AI models. The next step in the design was to clean the data which was followed by training and testing different AI regression models using the cleaned data set. How the data was cleaned and implemented into the solution and how it affected the AI models will be discussed in the solution implementation and evaluation. However, the most important aspect of the project solution design was deciding on what model should be used as the predictor of the housing prices.

In order to predict the housing prices for property in NSW, it was decided that a random forest regressor would be used. Due to this type of regressor performing the best out of the different models tested and providing a mean squared error of around 239355.05 for a data set with a mean housing price of 745030.62. After testing other models such as a decision tree regressor, ADA boosted regressor, support vector regressor, multilayer perceptron neural network and gradient boosted regressor the solution of using a random forest regressor was the most well suited for the problem. To understand the solution design, an understanding of what a random forest regressor is and how it works must be understood.

A random forest combines multiple decision trees and applies weights to each of the trees depending on which tree produces the best results. A single decision tree works by building a tree with branching paths based on the different variables in the data set. The idea is that the tree keeps branching off down different paths and combinations of attributes until it gets to a leaf node which would produce either a classification or in the case of the housing price problem, a number that predicts how much a house is worth. A random forest regressor works in the same way a classifier would. However, with a regressor the more branches the classifier goes down the more accurate the number produced by the regressor will be. This is different to a tree purely narrowing down which classification is more accurate using multiple branches. The decision tree alone is a very strong classifier with different parameters that can be changed to optimize the tree such as whether the tree has pruning or not which removes any branches that may be redundant

to make classifications. The quality measure can be changed which measures the quality of the attribute for classification and can be used to determine how well an attribute separates the training data. If an attribute is good at separating the data it may be chosen as a root node or higher up in the order of nodes. An example of a quality measure is information gain which determines which attributes split the data the best by calculating the entropy of the data, meaning how similar the data points are. For example an entropy of 0 means the data points are all the same, while an entropy of 1 means there are the same number of different values, i.e. 50/50 split if a classification is yes or no. Information gain can also be used in a regression tree. However, in the final solution for the housing price problem, the model implemented did not use a quality measure. Instead, it used a random split of attributes for each of the decision trees within the ensemble method. The model that was tested used 4000 n_estimators which means there were 4000 decision trees used within the model.

Overall, the design of the project solution was to gather data on housing prices, which would then be cleaned and tested on a variety of regression models. After testing the different models a random forest regressor was used to test a larger sized data set on which will be discussed in the evaluation section of the report.

Solution Implementation

1. **Web Scraper**

The solution implementation involved implementing a web scraper to collect data from the property valuation page on the NSW government site, followed by data cleaning and then implementing the data to train the AI models. The web scraper is designed to go on the property valuation page and move through each year within the annual sales data section of the website. Once a year is selected a zip folder containing the sales data for that year is downloaded. In the code the web scraper iterates through each year on the website and downloads them in the following link: `'/content/drive/Shareddrives/PropertyTrack/files/{year}.zip'`. Once the files have all been downloaded they are then extracted from the zip and all the files are outputted into a file based on the year they were extracted from. Once this was completed and all the data was stored it could then be broken up into data frames which could be used to draw out a specific number of files from the google drive to be used for testing and training the models. Figure 1 details the full code used to extract the number of files.

```
# In range one folder get the number of files
import shutil
import os as os
dir_name_two = '/content/drive/Shareddrives/PropertyTrack/rangeOne'
files = os.listdir(dir_name_two) # This is an array of names of folder... They have .zip on the end
file_count = len(files) # Get the number of files in the folder

# Create dataframe - This number can be changed
df_one = getDataframe(10000,"B",'/content/drive/Shareddrives/PropertyTrack/rangeOne/')

print(df_one.shape)
print("done")
```

Figure 1: Python code to extract data frame from web scraped data

2. Data Cleaning

Once all the data had been set up, the next step was to clean the data so that any irrelevant data attributes could be removed and the necessary attributes could be included in the training of the models with suitable attribute types, value types and without missing values. The irrelevant data consisted of the attributes; 'Dealing Number', 'Record Type', 'Download Date/Time', 'Property name', 'Strata Lot Number', 'Settlement Date', 'Nature of Property', '% Interest of Sale'. These attributes are useful to explain when and how the property got sold, however, in terms of predicting the housing prices the data does not provide any relevant information and would most likely diverge the regression models from an accurate prediction. The code used to remove the unwanted data attributes is seen in figure 2.

```
# Data points that aren't useful
# Dealing Number, Download DateTime, Property Name, Record Type, Strata Lot Number
df_one = df_one.drop('Dealing Number', 1)
df_one = df_one.drop('Record Type', 1)
df_one = df_one.drop('Download Date/Time', 1)
df_one = df_one.drop('Property name', 1)
df_one = df_one.drop('Strata Lot Number', 1)
df_one = df_one.drop('Settlement Date', 1)
df_one = df_one.drop('Nature of Property', 1)
df_one = df_one.drop('% Interest of Sale', 1)
```

Figure 2: Python script to remove unwanted attributes

Where df_one is the name of the data frame and the drop method is used to remove or drop the column from the data set. The data set began with 25 attributes, however, after the data cleaning the data frame only contained 17 attributes. Once the irrelevant data attributes had been removed the next step was to change the typing of all the attributes. When the data set was downloaded, all of the attributes, no matter whether the values in the attributes were numerical, were set as type string meaning that the types of the remaining attributes had to be changed and any missing values had to be removed. This involved changing values of type string to values of a numerical type such as a float or int. The code that changed all the necessary attribute types and numerical types is seen in figure 3.

```
df_one["District Code"] = df_one["District Code"].astype(float)
df_one["Property id"] = df_one["Property id"].astype(int)
df_one["Sale Counter"] = df_one["Sale Counter"].astype(float)
df_one["Property Post Code"] = df_one["Property Post Code"].astype(int)
df_one["Purchase Price"] = df_one["Purchase Price"].astype(float)
# df_one["Area"].astype(float)

print(df_one.dtypes)
```

Figure 3: Python code to change string variables to floats or ints.

However, when changing the types of the attributes some attributes only contained string values such as ‘Property Street Name’, this meant that not only did the types of the attributes have to be changed but the values inside the attributes did as well. To implement this, a label encoding method was used to give numerical values to replace each of the string values. For example ‘Property Street Name’ had its values changed to numbers which corresponded to its street name meaning if a house was sold on the street LEWIS LANE it was given a numerical value of 1. This was used across all the attributes that contained only string values including "Property Street Name", "Property Locality", "Zoning", "Primary Purpose" and "Component code". After implementing the code to numarise all of the string values within the attribute the following table, shown in figure 4 was produced to highlight the changes:

Property Unit Number	Property House Number	Property Street Name	Property Locality	Property Post Code	Area	Area Type	Contract Date	Purchase Price	Zoning	Primary Purpose	Component code	Sale Code	Property Street NameNumeric	Property LocalityNumeric	ZoningNumeric	Primary PurposeNumeric	Component codeNumeric
	553	BALD HILLS RD	GRENFELL	2810	130	H	20131016	180000.0	RU1	VACANT LAND	RME		0	0	0	0	0
	671	LEWIS LANE	GRENFELL	2810	1.836	H	20131018	58300.0	RU1	SILO	RMW		1	0	0	1	1
	2991	THE LAKES WAY	TARBUCK BAY	2428	10.4	H	20131005	241000.0	R	VACANT LAND	PN		2	1	1	0	2
	94	BAYVIEW RD	TEA GARDENS	2324	839.7	M	20131009	655000.0	A	RESIDENCE	AV		3	2	2	2	3
	1	SETTLERS WAY	TEA GARDENS	2324	749	M	20130822	260000.0	A	VACANT LAND	AV		4	2	2	0	3

Figure 4: Table which showcases changes to string variables

Once the attributes had their string values changed to numerical values their typing was then changed to either a float or int which finalised the data cleaning implementation.

3. Data Training and Regression Model Implementation

The data was trained using the train_test_split module imported from sklearn. The feature columns were implemented, which excluded the property price column. The purchase price column was implemented as a y dataframe. The respective x and y testing and training datasets were defined with the test size implemented as 0.3. Figure 5 shows the code used to split respective train and test datasets. The data was then trained per each regression model by importing their specific module, and using the .fit function to train the X and Y training sets.

```
from sklearn.model_selection import train_test_split

feature_cols = ["District Code", "Sale Counter", "Property Post Code", "Property Street NameNumeric", "Property LocalityNumeric", "ZoningNumeric", "Primary PurposeNumeric", "Component codeNumeric"]
y = df["Purchase Price"]
X = df.loc[:, feature_cols]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

Figure 5: Python code used to split train and test datasets

The data was then trained per each regression model by importing their specific module, and using the .fit function to train the X and Y training sets. Figure 6 shows the example for the Random Forest parameters and training code.

```
[ ]  # Random Forest regressor

     from sklearn.ensemble import RandomForestRegressor
     clf = RandomForestRegressor(n_estimators=500)
```

```
[ ]  clf.fit(X_train, y_train)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=500, n_jobs=None, oob_score=False,
                      random_state=None, verbose=0, warm_start=False)
```

Figure 6: Random forest regressor implementation and training code

The final step in the implementation process was importing different AI models to test which one achieved the best results when predicting the housing prices of property in NSW. This was done using the following code shown in figure 7 for each regression model utilised:

```
y_pred_RF = clf.predict(X_test)
```

Figure 7: Random Forest prediction code

The measure to decide which model performed the best was an absolute mean error. This means that the difference between the predicted and true values are calculated and then summed and the average difference between the predicted and true values, positive or negative, resulted in the absolute mean error range. This was the best scale for judging which model performed the best as the goal of the models is to make a prediction with as minimal error as possible as opposed to having an accurate classification model. The models were imported from sklearn packages and the following were chosen. A Random Forest Regressor, ADA Boost Regressor, Support Vector Regressor, Decision Tree Regressor, Neural Network and a Gradient Boosted Regressor. These were chosen as prior research showed that random forest and gradient boosted regressors perform very well in these types of problems and other models were chosen in order to see if non-ensemble models such as a decision tree regressor and SVR would outperform the ensemble methods. In order to run all of these models they were first initialized, then the data was fit to the

model using the fit method and finally a prediction was made which was then put against the true values to calculate the absolute mean error. After the implementation of the models was complete the random forest regressor came out on top and was then tested with 10,000 files to simulate a more accurate real world setting. The results of testing the models and how the random forest regressor performed on 10,000 files will be discussed in the evaluation section of the report. Overall the solution was successfully implemented and 6 working models were developed to predict the housing prices of property in NSW.

Evaluation of the Solution

To evaluate the accuracy of the explored Artificial Intelligence algorithms, the data was restricted to 100 files which is approximately 54,400 input variables, consisting of 3,200 rows and 17 columns. This was done to reduce time and errors associated with high resource cost. Google Colabatory was used in an attempt to train all explored models using an input of 10,000 files. However, this resulted in extended processing times, shortage of RAM and failure in successful data training.

As we implemented regression algorithms, the accuracy cannot be determined in the form of an accuracy score. Therefore, we employed three valuative metrics suitable for regression algorithms. These metrics detailed how accurate the predictions generated were based on the predicted value deviations with respect to the input values. From these calculations, we were able to pick our final solution model based on the best statistical data generated.

Firstly, the absolute mean error was calculated to determine how large the expected error can be from the average forecast. Absolute mean error was used to measure the difference between two continuous variables, in this case the y_test values and the y_pred values. The calculation takes the average sum of the absolute prediction errors on all instances of the test set, by measuring the extent of errors generated by the prediction output. The absolute mean error range was calculated by subtracting and adding the absolute mean to the true mean of the prices. This was used to gain an idea of the maximum and minimum error range. Figure 8 details the calculation necessary to determine the absolute mean error as well as the python code used to calculate the absolute mean error value of each explored regression model.

$$\text{MAE} = \underbrace{\frac{1}{n}}_{\text{test set}} \sum_{i=1}^{n} \underbrace{|y_i}_{\text{predicted vaue}} - \underbrace{\hat{y}_i|}_{\text{actual value}}$$

```
mean_absolute_error(y_test, y_pred)
```

Figure 8: Absolute mean error equation and python script

Additionally, to further evaluate the metrics of the implemented models, R^2 was used. R^2 is the squared correlation between the observed result values and the model's projected values in regression models. The score generated can be multiplied by 100 to produce a percentage, which quantifies the disparity in the output variable being explained by the input variable. An R-squared value of 0 suggests that there is no variance correlation between the independent variables (X) and the dependent variable (Y). Whereas an R-squared value of 1 indicates that the variance correlation between the independent and dependent variables are fully accounted for in the model. Therefore, the higher the R-squared value, the better the model. Figure 9 showcases the calculation used to determine each algorithm's R-Squared value and the python code used to calculate the R-squared values for each regression model.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

```
coefficient_of_dermination = r2_score(y_test, y_pred)
print(coefficient_of_dermination)
```

Figure 9: R-squared equation and python script

According to table 1, the models that performed poorly were: neural network, support vector regressor and ADA Boosted Regressor. This is evident when evaluating their high absolute mean error values, which are indicative of potential highly inaccurate results. As well as minute R-squared percentages – with Support Vector Regressor scoring -2%, indicating there was no variance correlation between the input and output values. It is interesting to note that with ADA Boosted Regressor, the R-squared percentage is the third highest value at 55%. However, due to the mean absolute error value being a substantial value over 600,000, the R-squared percentage in relation to the ADA Boot Regressor's accuracy was ignored.

Table 1: Metric evaluation results for each regression model implemented

Regression Model	Absolute Mean Error	Absolute Mean Error Range	R-Squared Percentage
Random Forest	239355.05	505675.57 - 984385.68	61%
ADA Boost	600454.93	30374.82 - 1459686.43	55%
Support Vector	340582.62	404447.99 - 1085613.25	-2%
Decision Tree	296239.29	448791.33 - 1041269.92	47%
Neural Network	524668.03	341494.88 - 1148566.36	0.053%

Gradient Boosted	281876.14	463154.48 - 1026906.76	56%

The metric evaluations for each model revealed that the Random Forest Regressor and Boosted Gradient Regressor models achieved the most appealing results, indicative of the best performances with respect to our project aim. This is shown by the Random Forest Regressor's mean absolute error calculated to be 239,355, which is the lowest mean absolute error value produced by all other models. Additionally, the R-squared percentage was calculated to be 61%. This means that 61% of the variability in the predicted data is explained by the model. Conversely, this also showcases that 49% of the variability in the predicted data cannot be supported by the Random Forest Regressor model. Comparatively, the Gradient Boosted Regressor achieved a mean absolute error of 281,876, and a R-squared value percentage of 56%. These statistical values were lower than the Random Forest Regressor. However, they were considerably close when compared to other regression models. An honourable mention is given to the Decision Tree Regressor, as it placed in third place for its performance evaluation. However, Random Forest has proven to be more precise than a Decision Tree model, as it utilises a randomised selection of data that is better generalised that produces more precise predictions.

To evaluate and compare the effectiveness of using Random Forest and Gradient Boosted Regressors to predict property prices, we implemented the two models again. However, a data frame consisting of 10,000 files was used. The implementation of 10,000 files in the data frame resulted in 4,575,363 input values appropriately trained and tested by each competing regressor. Table 2 showcases the results of 10,000 files used on the Random Forest and Gradient Boosted Regressor.

Table 2: Metric evaluation results for the two best performing models using 100 files

Regression Model	**Absolute Mean Error**	**Mean Error Range**	**R-squared percentage**
Random Forest	293661.67	527663.37 - 1114986.71	55%
Gradient Boosted	362343.07	458981.96 - 1183668.10	47%

These results reveal that with the addition of 9,900 files, both models performed worse. However, Random Forest Regressor outperformed Gradient Boosted Regressor with an absolute mean error of 29,3661, which is 68,682 less than the absolute mean error calculated from the Gradient Boosted Regressor model. This outperformance is reiterated by the R-squared percentage values, with random forest regressor obtaining the higher percentage at 55%. Therefore, through this evaluation, we concluded that the implementation of Random Forest Regressor to predict property prices in NSW was the best option.

To further evaluate the accuracy of the Random Forest Regressor, scatter plots were created to visualise any trends when comparing the observed prices to the predicted prices. Figure 10 details the scatter plot with y_test on the X-axis, and y_pred_RF on the Y-axis. The numbers shown on the plot are in scientific notation and multiplied by 10^6.

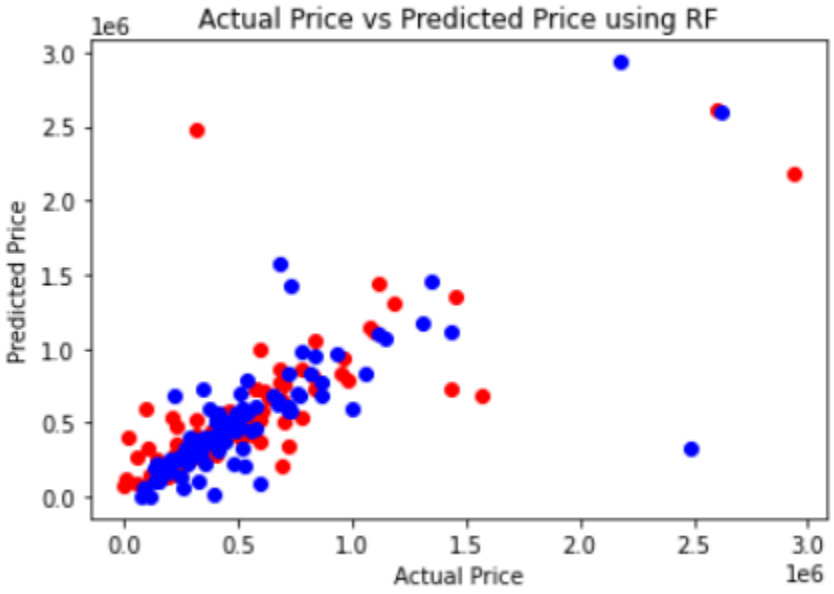

Figure 10: Scatter plot of Random Forest Regression model actual price vs predicted price

This graph shows that there is some moderate correlation between the smaller values of actual price and predicted price. The correlation is shown to become weak as the price predictions and actual prices progress higher in value. This insinuates that the most accurate predictions were made on smaller prices, whereas more errors were made when trying to predict the value of more expensive properties. To further visualise the results from the Random Forest Regressor model, the residuals were calculated by using the simple equation. Residual = Observed – Predicted. The values generated were then used as the y-axis, denoted as standardised residuals. Y_pred was used for the x axis. This graph allowed us to visualise how many predictions were correct if they resided around y = 0. Values dwelling around negative y values implies that the prediction was too high compared to the observed value. Scatter points above 0 indicate that the predictions

generated were too low. Figure 11 details the scatter plot of residuals and predicted price using Random Forest Regressor.

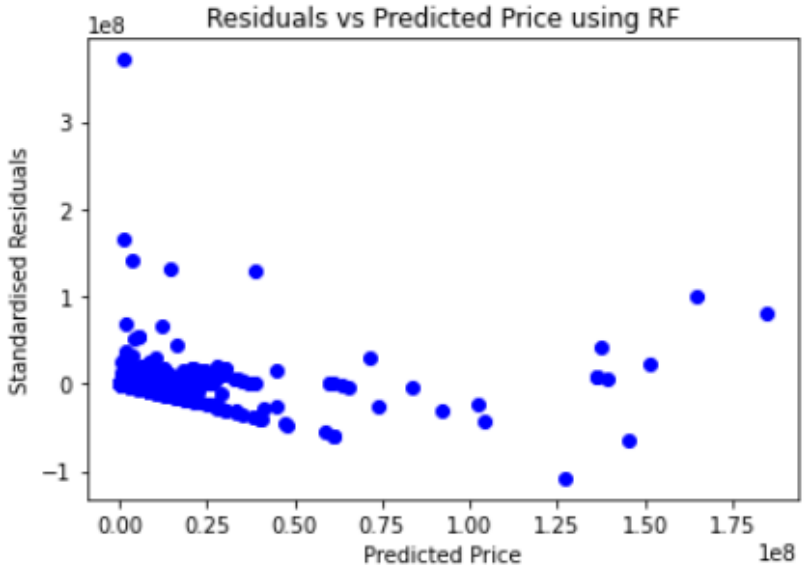

Figure 11: Scatter plot of Random Forest Regression model residuals vs predicted price

This scatter plot shows that there is an unequal scatter of points which indicates possible heteroskedasticity. This is evident by the variance of the residuals increase in conjunction with the increase of the predicted values. This means that the variance of the residuals is unequal over the predicted price range. An explanation as to why this graph could be considered heteroscedastic in nature can be due to the large input prices implemented in this model, which indicates wide ranges of largest and smallest observed values, thus affecting the variance of the residuals. However, there are a multitude of predictions residing around y = 0 which indicates the estimates were close to the observed value. Conversely, there were also values residing above and below y = 0 which implies there were some predictions that were too high or too low.

In conclusion, the regression model chosen was the Random Forest Regressor, this is because it provided the best results when compared to the other five models explored. However, there is still room for improvement as the model possesses a substantial mean absolute error value of over $200,000, which is a high dollar value error when trying to predict the cost of a property. Consequently, the R-squared percentage was not considered high enough to be scientifically valid, as a percentage of upwards of 80% is widely appraised as being accurate. Additional ways to improve this model in the future will be discussed in the remarks and discussion section of this report.

Remarks and Discussion

Because of advancements in computing technology, it is now possible to evaluate social data that could not previously be gathered, processed, or analysed. In this report, we have looked at implementing different artificial intelligence techniques to aid in the prediction of property valuation in NSW. This is a small venture in a field with high potential for further research.

Six different artificial intelligence models were applied, and absolute mean error and R-squared values were calculated to evaluate their suitability. Random forest regressor produced the best results in this scenario. Winky K.O. Ho, Bo-Sin Tang & Siu Wai Wong found similar results in their 2020 study, where they compared three machine learning algorithms, random forest, SVM and gradient boosted machine for determining the value of a home in Hong Kong (Winky K.O. Ho, Bo-Sin Tang & Siu Wai Wong, 2020). They used three performance metrics, one of which was mean squared error, and found that random forest had the best predictive power. The correlation of these results validates our findings and gives us confidence in our implementation of the AI techniques.

Among all regression methods, random forest generally proves to be the most accurate. This approach can handle large amounts of data with thousands of variables. The averaging of the ensemble of decision trees reduces overfitting which also contributes to a higher accuracy. It is relatively quicker to train as only a subset of features is at work. Because we can save created forests for future applications, prediction speed is substantially faster than training speed. The variation of each decision tree is high, but the bias is modest.

Another important thing to consider if we want to implement this project in a real-world setting, was the amount of time the computation of these machine learning algorithms took compared to other property pricing models. The algorithm to use is determined by a variety of parameters, including the size of the data set, the computational capacity of the equipment, and the amount of time available to wait for the results. Google Colab's shortage of RAM and CPU limitations also proved to be a major obstacle in terms of training the models. The session would run out of RAM every time the massive dataset of over 10,000 files was inserted. As the primary concern for comparison reasons here was the predictive accuracy and not the waiting time, random forest eventually trumped support vector and gradient boost regressors.

The reliability of the findings in this project could have been further amplified through the use of property feature data, such as the number of bedrooms/bathrooms in each house. This sort of personal data is not widely available on the internet. Real estate companies spend a lot of time and money in obtaining this data to build complex prediction models. Zillow, for example, until recently had a deal with the list hub that provided them with the majority of their listing data across the United States. Almost all of their data comes from Multiple Listing Services (MLS),

county records, user-entered data and pictures, and manually entered data from agents. Zillow receives data from a variety of third-party sources, many of which are undisclosed (Adams, 2020).

To conclude, the use of machine learning in property research is still in its infancy. More time and money researching this topic will yield incredible results in terms of real world application. A price prediction model for property buyers is a game-changer and could be a lucrative business. Incorporating additional property transaction data from a bigger geographical location with more attributes, or analysing other property categories other than housing development, could be a future study direction.

Individual Reflection

Andreas Gwyther-Gouriotis:
The project ended up being quite successful as all the models were able to be trained and produce some decent results. After looking at the predictions for some houses and versus other houses the ones that were the furthest off ended up being outliers in terms of how much the house was actually worth meaning it significantly affected the size of the error for some of the models such as the random forest. However, the implementation of the code was completed smoothly and each of the components to the assignment were able to be completed within the time discussed within our group meaning the project was quite smooth. Aspects that could be improved was how the AI was being run for training and testing. Since the AI was being run on a google collab it was not possible to train the AI on all the files in the data frame from 2001-2014. Multiple team members tried running the files to try and train the AI, however, no team member was able to train any of the models as each time the collab session would run out of RAM which was unfortunate. Even so, the random forest was still able to be run on over 10,000 files and over 200 000 rows of data which still gave the model appropriate training for real world problems and showed it would be able to perform with a relatively low error margin when used to predict much larger sets of data. One aspect of the project that was not expected was how long the data cleaning and gathering process would take. A large majority of the time set to complete the project was given to cleaning the data and ensuring that the models could not only be trained using the data but that it would produce successful results. Overall the project was completed to a high satisfaction and I am happy with the results of the group especially seeing as this is our first AI project.

Jordan Stojcevski:
I am really happy with the results from the AI implemented by myself and my other group members. Having some minor experience in the real estate sector, I know it can be very tricky to give a valuation on a property and with current trends in the market it is becoming increasingly difficult to do that. Testing multiple different AI models showed me that the data used and

features given to the model are very important to the overall accuracy of the predictions. I can definitely see the applications of this being used in the real world and think it was a great problem to work on. It would have been good to try implement some other AI methods and integrate those predictions as features into another model, such as analysing the condition of a property from a google images street view and giving it a classification, analysing sentiment of blog posts and tweets from reputable sources in the real estate sector, as well as integrating demographic data into the model.Overall I am extremely happy with the results from this assignment and think I gathered some great experience by doing it.

Navid Choudhury:
Not only was this assignment successful in teaching me how different artificial intelligence techniques work, but also their importance in a real world setting. Learning about the technicals in the lecture was a great start before tackling this project and building predictive models. I was a little nervous in the beginning as I have only done subjects that teach Java so far, but my group mates were incredibly helpful when we worked on the Google Colab file together. A web scraper was developed to collect the data from a reputable source, and it was cleaned to remove irrelevant attributes. Implementing six AI techniques meant I could push the boundaries of my knowledge in this field to learn about how each of them work. The comparison between them also added a new dimension as it enabled more research into the drawbacks they present in the real world. Looking into previous studies and seeing how they also came to similar conclusions instilled a lot of confidence in how we approached this assignment. Pre-processing the data and building the models was very time-consuming but we got the results we were looking for in the end. Overall, I am very happy with how the assignment went. The topic we picked was not only interesting, but also very relevant in today's climate. If researched with more time and effort, it could develop into a business venture. Working alongside my group mates was a tremendous experience as well. It has certainly developed a keen interest in me for the application and benefits of artificial intelligence in the real world.

Imogen Turner:
When approaching the group project, I was initially confronted by a lack of confidence in my ability to understand and apply Artificial Intelligence to a real-world scenario. The examples for assignment 3 were easy to understand when implemented on small datasets that held no true worldly relevance. However, as I explored this assignment, I began to acknowledge the importance that artificial intelligence possesses in relation to property price predictions. It eradicates human processing time, bias, errors and provides an idealistic solution when analysing and exploring large amounts of data. I thoroughly enjoyed researching and understanding the benefits of AI, and translating that textbook knowledge into a practical setting, with the Python script development of our project. Although I have completed other subjects detailing programming with Python, I found that this assignment, and other assignments in this subject

allowed my python skills to excel. Rather than just copying and pasting lines of code, this project allowed me to understand exactly what the code was doing and its relevance to the AI problem we were trying to solve. Additionally, this project allowed me to grasp the pros and cons of various machine learning algorithms when comparing their error rates and other statistical metrics, especially when trying to quantify the accuracy of a regression model which can only be done using error metrics. I was able to identify and understand why our final solution worked best and what factors inhibited this project to excel, such as failure to identify high correlative data from the NSW Valuer General data and issues with Google Colabatory when trying to implement models using a data frame of over 10,000 files. Moreover, I was also able to acknowledge the arduous process of data cleaning and collection using a web scraper, as that section of our project took the most amount of time. Over all, I am extremely happy with this project and think it showcased a real world problem that can be solved by Artificial Intelligence.

References

Castle, J. (2017). *Property valuations and price estimates | CHOICE*. CHOICE. Retrieved 2 November 2021, from https://www.choice.com.au/money/property/buying/articles/property-valuations-and-price-estimates.

Chaphalkar, D., & Sandbhor, S. (2013). Use of Artificial Intelligence in Real Property Valuation. *International Journal Of Engineering And Technology (IJET)*, *5*(3). https://doi.org/0975-4024

How Artificial Intelligence is Shaping the Real Estate Market- Co-libry. Co-libry. (2021). Retrieved 2 November 2021, from https://co-libry.com/blogs/property-ai-real-estate/.

Neal, M., Strochak, S., & Young, C. (2020). How Automated Valuation Models Can Disproportionately Affect Majority Black Neighborhoods. *HOUSING FINANCE POLICY CENTRE - URBAN Institute*. Retrieved 2 November 2021, from https://www.urban.org/sites/default/files/publication/103429/how-automated-valuation-models-can-disproportionately-affect-majority-black-neighborhoods_1.pdf.

Ecker, M., Isakson, H., & Kennedy, L. (2020). An Exposition of AVM Performance Metrics. *Journal Of Real Estate Practice And Education*, *22*(1), 22-39. https://doi.org/10.1080/15214842.2020.1757352

Ecker, M., Isakson, H., & Kennedy, L. (2020). An Exposition of AVM Performance Metrics. *Journal Of Real Estate Practice And Education*, *22*(1), 22-39. https://doi.org/10.1080/15214842.2020.1757352

Bulk Property Sales Information. NSW Government Valuer General. (2021). Retrieved 2 November 2021, from https://valuation.property.nsw.gov.au/embed/propertySalesInformation.

Adams, D., 2020. *Where Does Zillow Get Its Data?*. [online]. Available at: <https://therealestatedecision.com/where-does-zillow-get-its-data/> [Accessed 5 November 2021].

Winky K.O. Ho, Bo-Sin Tang & Siu Wai Wong (2021) Predicting property prices with machine learning algorithms, Journal of Property Research, 38:1, 48-70, DOI: 10.1080/09599916.2020.1832558