Applied Machine Learning Report

I. Coursework 1 - House Price Prediction

II. Introduction

The purpose of this project was to determine the best performing supervised machine learning algorithm for making house price predictions, through designing an appropriate pipeline including an evaluation strategy, experimentations of different models and evidence of iterations.

Predicting house prices can prove useful for the efficiency of real estate companies to present to their clients, as well as individuals and investors who wish to understand the market to make informed decisions on their potential investments. The 'Real estate valuation' dataset for this regression task was collected from the Shindian District, New Taipei City, Taiwan.

Regression tasks predicts continuous variables (numbers) as regression analysis allows you to see the relationship between variables and indicates which variables are statically significant to the the target variable the model is trying to predict. Linear Regression, Random Forest Regression and K-Nearest Neighbors Regressor were the regression models trained, and tuned for optimum performance of this task, with Random Forest Regressor achieving 73% accuracy after cross validation.

III. Data And Preliminary Analysis

A. Data

The 'Real estate valuation' dataset was donated to the UC Irvine Machine Learning Repository by Professor I-Yeh Cheng of TamKang University, Taiwan. The UC Irvine Machine Learning Repository (2018) states that "The dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. This allows for the sharing and adaptation of the datasets for any purpose, provided that the appropriate credit is given." The tabular dataset was retrieved as an Excel (.xlsx) file format, with multivariate data containing 414 observations, 7 attributes of integer and real data types related to real estate including samples, time-series and geographical data.

B. Preliminary Analysis

Exploring the data to gain insight of each attributes, its characteristics and the correlation between them are important for the development of a more accurate prediction model. This process leads to identifying attributes that make stronger predictor variables and ones to eliminate, also known as 'feature selection', as in contrary to popular belief, more features does not always lead to better results and processing redundant and irrelevant data is an expensive operation, using a lot of computational power and memory.

During preliminary analysis, outliers were identified and later removed as for this particular task it is considered as "noise" which would lead to a skewed and less accurate price prediction. Data visualization was produced to show data exploration between each variable.

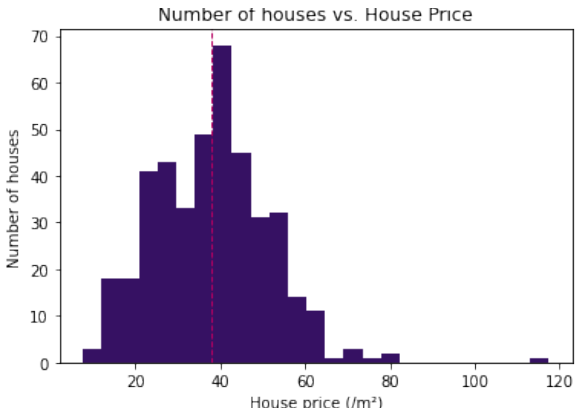
IV. Methods

The machine learning pipeline is a workflow with iterative steps for practitioners to build, train and evaluate the performance of appropriate models for their task at hand. The pipeline is great for monitoring the progress and the steps taken as well as providing a good structure for both individuals and teams to refer back to while working on their machine learning projects.

1. Load Data
2. Clean Data
3. Preliminary Analysis
4. Feature Selection
5. Feature Scaling
6. Building the model
7. Fit and train model
8. Performance evaluation
9. Improving the model
10. Deployment

A. Pre-processing

As outlined above there are five steps that make up the first stage of the pipeline, 'pre-processing'.
Loading and cleaning the data allows you to gain a basic understanding of the domain of the task, in this case, real estate. Upon loading the dataset there was an extra column which represented the index of the dataset, this was removed as it was unnecessary and could cause inaccurate results for the data analysis and predictions. Furthermore, column names were changed for consistency. This dataset did not contain any missing values, therefore imputation (dealing with missing values, by removing or filling with 0 or the mean value of the attribute) was not required. Column names were also cleaned for consistency.

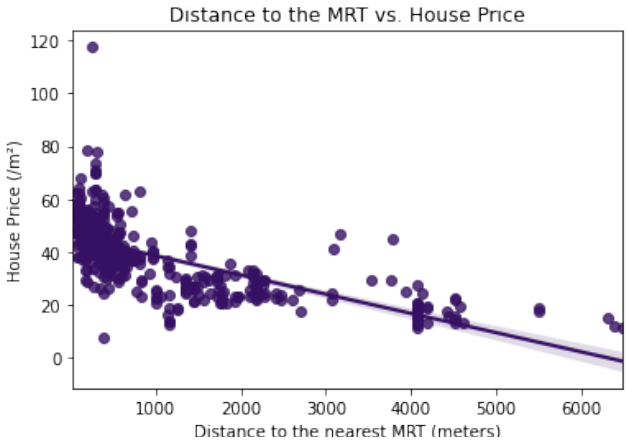


B. Preliminary Analysis

This was performed by asking questions and visualising the data to understand relationships between attributes and the target variable (house_price). The initial analysis was identifying the target variable and analysing the lowest, highest and average values of this variable. Checking the distribution of the target variable the above shows the house price and the number of houses sold in the price range. It is

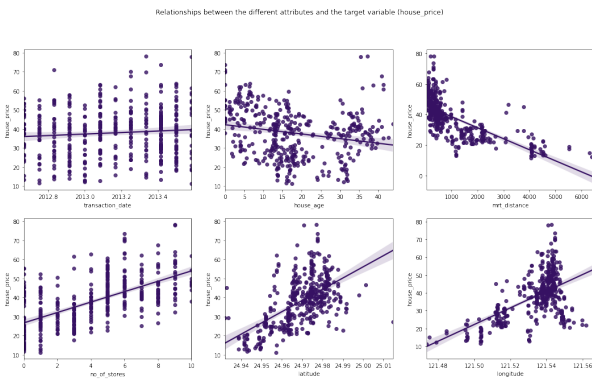
skewed right, where the right side is the highest house price sold, with only one house selling for 117.5/m². Most houses sold between 10/m² - 65/m². Around 15 houses' sale price was between 60/m² - 80/m².

Initial assumption based on the variables is that the distance to the nearest metro station would have a high correlation to house prices as it is common knowledge that nearby transportation is an important factor when considering purchasing a home. The figure below shows the impact of the distance of the nearest metro station (mrt_distance) on the house price (house_price) where we can also see some outliers, which were later removed as they would alter the slope of the regression line, this would lead to an inaccurate reading.



The conclusion from this observations are as follows:

- It seems that the houses with the shortest distance to the metro station have a higher house price.
- It is also evident that the houses with a shorter distance to the metro station, sold more than those with greater distance to the metro station.
- The outlier on the plot shows the house with highest price sold at 117.5/m², situated 252.58 meters to the nearest metro station.
- However, another outlier shows that the house sold at the lowest price was situated close to a metro station with a distance of only 393.26 meters.
- Nevertheless, the distance to the nearest metro station is likely to be a strong predictor for the house price.



The sub-plots are an overview of all the scatter plots of each attributes and the relationship to the target variable, this is also useful to identify strong predictor variables for the prediction model. Some key take aways from these plots are (from left to right):

1. The transaction of house sales (transaction_date) spans over a 1 year period from August 2012 - August 2013, despite the complicated formatting of this attribute, we can see that the house price rose over 60/m² in the beginning of 2013. However, the regression line indicates there is little to no relationship between this attribute to the house prices.
2. The houses range between 0 to approximately 45 years old, but there seems to be very little correlation to the house prices which could mean that the age of the house (house_age) may be a weak predictor variable for the house price.
3. There seems to be a big cluster where the distance to the nearest metro station (mrt_distance) is <1000 meters. There also seems to be more houses sold where there is a closer distance to a station, this indicates a strong predictor variable for house prices.
4. There is a linear trend of the number of stores within the living circle and its impact on the house price, this could also be a strong predictor variable and shows that the buying market in that district values convenience as also reflected in the previous observation (distance to the nearest metro station).
5. There is a cluster of house prices sold between 40/m² - 50/m² at a latitude between 24.97 and 24.98 and there are very little houses sold after the latitude of 24.99.
6. Longitude shows a positive trend, with a cluster of houses sold between 30/m² - 60/m² at 121.54 longitude. Latitude and longitude could also be good predictors of house prices as locations do play a big part in house sales.

The observations gave some clarity on the feature selection process, which was done through filtering by correlation. The heat-map shows strong negative correlation between mrt_distance and house_price, as previously observed the further away the property is the a metro station, the house price decreases. This heat-map also confirms that the transaction_date is not a strong predictor variable and therefore removed to prevent noise for the prediction model. The attribute 'house_age' had the lowest correlation to the house price, after transaction_date, but was not removed for the purpose of the first model iteration.

The final part of data pre-processing was 'feature scaling'. The dataset was normalized by converting the varying range of values to the same range, between [0,1]. The reason for normalization is to prevent bias, as attributes with a higher range can be mistakenly analysed as an important predictor variable for the results of our task.

C. Model Training

This part of the pipeline is an iterative process to identify the best model for the task. Firstly, a model is selected (Linear Regression, Random Forest Regressor, K-Nearest Neighbor Regressor), the data that was prepared earlier is then split into training set (X) and testing set (y). The training data are the predictor variables selected during 'feature selection' to train the model (house_age, mrt_distance, no_of_stores, latitude and longitude) and the testing data is the target variable (house_price).

The model is fitted, tested and evaluated on its performance leading to the 'improving the model' stage to gain a higher performance accuracy.

D. Performance Evaluation

The evaluation for the regression task is measured by R^2 score, Mean Squared Error (MSE) and Mean Absolute Error (MAE). R^2 score measures how well the chosen regression model fits to the testing data, the best score is 1 and can be interpreted as a percentage (100%). MSE is a loss function which returns the average of the squared difference between predicted values and the expected predicted values in a dataset, the best value is 0. MAE calculates the mean difference between the predicted value and the actual value, the best value is 0. Evaluating is an important step in machine learning to compare models and identify areas for improvement where under-fitting or overfitting may arise.

E. Improving the model

Often when training with a small dataset such as the 'Real estate valuation' dataset, can lead to either underfitting or overfitting. Underfitting is when the model cannot train properly on the given data this could be due to the lack of data or inadequate data, and produces a high error rate as well as not being able to generalise to new data. Overfitting however is more prevalent, especially in smaller datasets where the perfomance accuracy is high but the model cannot generalize on new data, due to the model training on the noise that exists in the particular dataset. Overfitting can be prevented in the data cleaning stage, however with small datasets sometimes more data is needed, one of the ways to do this is to perform cross validation. K-fold cross validation was chosen for this task to resample the dataset by shuffling and splitting into 5 groups where one of the groups is held out for testing while the rest is being trained. This returns a new evaluation for the dataset as it is able to learn from multiple shuffles and groups as a result, the model returns a better performance accuracy.

F. Deployment

This stage concludes the best performing machine learning model for the task and is ready to make predictions with new data and valuable insights can be taken to make business decisions, note that the deployment stage is not the end but the final part of an iterative process as models should be monitored and maintained regularly for consistent accuracy.

EXPERIMENTS

Linear Regression - This model was chosen for its simplicity to implement and results can be easily interpreted. However, it is sensitive to outliers as it can impact the slope giving an inaccurate reading. Linear regression could cause overfitting however, as previously mentioned this can be avoided during data processing or with cross validation which was performed as for this task. It was evident the performance of the linear regression dropped from ~67% to to ~61% after cross validation, which poses the question of overfitting at the first training of the model.

Random Forest Regression - This ensemble learning algorithm was chosen for working with non-linear data to make better prediction through building multiple decision trees and randomly selecting features which are later averaged out to provide the model, it is very efficient and in this case it yielded a higher R^2 score than the linear regression model at ~71%. After cross validation, the performance increased by ~2%, to ~73%. It is also the best performing model of all three models trained for this ask after the cross validation process. Random Forest however is susceptible to overfitting so the hyper-parameters need to be tuned.

K-Nearest Neighbors Regressor - This model makes a calculated prediction based on the k number of neighbours in the proximity. The initial performance gave a high R^2 score of ~73% and was later decreased to ~71% after cross validation. This model is susceptible to under-fitting or overfitting if the optimal K value is not chosen correctly.

REFLECTION

Random Forest Regressor was the best performing model for this task after cross validation, where kNN Regressor initially performed better but dropped the R^2 score dropped after K-fold validation. As the dataset was quite small, it can prove challenging to gain very high R^2 score accuracy without the model being prone to overfitting. Key take aways from this coursework is to experiment more with hyper-parameter tuning and try sequential feature selection to optimize the performance of the prediction model. The overall concept of each models was difficult to grasp at first instance and would require more time and practice of experimentations to full understand the best model for each task.

Performance Metric	**Model Comparison**		
	Linear Regresison	***Random Forest Regressor***	***kNN Regressor***
R^2 score	0.67	0.71	0.73
Mean Squared Error (MSE)	0.01	0.01	0.01
Mean Absolute Error (MAE)	0.08	0.07	0.07
Cross Validation (R^2 score)	0.61	0.73	0.71

REFERENCES

1. UCI Irvine Machine Learning Repository, (2018), Real estate valuation dataset, https://archive-beta.ics.uci.edu/ml/datasets/real+estate+valuation+data+set [Accessed on 1 March 2022]

COURSEWORK 2 - IMAGE CLASSIFICATION WITH CONVOLUTIONAL NEURAL NETWORK

INTRODUCTION

The aim for this project is to design a Convolutional Neural Network (CNN) algorithm to classify images into predefined classes of the given training datasets.

A Convolutional Neural Network (CNN) is one of the most popular types of Neural Network (NN) architectures which is used to perform image processing, as well as natural language processing tasks. The timeline of CNN dates back all the way to the 1950's where Frank Rosenblatt invented *Perceptrons*, linear classifier algorithm used for supervised machine learning to classify input objects into classes.

Performance Metrics		CNN model	Optimized CNN model
Accuracy (Max)	Training	0.718	0.883
	Validation	0.658	0.675
Loss (Min)	Training	0.861	0.342
	Validation	1.071	1.300

Code For Courseworks

Coursework 1
https://drive.google.com/file/d/146kZymUmFc1DL-20tTIDVvLhiKEx1HkS/view?usp=sharing

Coursework 2
https://colab.research.google.com/drive/1kfR5JcQeH6gbMstSS6BslWedRb_sqx_l?usp=sharing