Coursework Report

Contents

[Introduction 2](#_Toc34827325)

[Exploratory data analysis 2](#_Toc34827326)

[Data Munging 2](#_Toc34827327)

[Process of cleaning 2](#_Toc34827328)

[What didn’t work 3](#_Toc34827329)

[What worked 3](#_Toc34827330)

[Modelling approach 3](#_Toc34827331)

[Why Linear regression was chosen to build the model 3](#_Toc34827332)

[Current model 3](#_Toc34827333)

[Previous models and attempts 4](#_Toc34827334)

# Introduction

This report will be explaining the steps that were taken to clean the Manhattan dataset in order to prepare it for building the predictive model. The model that has been chosen will be reviewed, given reasons why it was chosen and an explanation of the model building process.

# Exploratory data analysis

Before the data was loaded into a Jupyter notebook the csv file was inspected in excel below is what was found:

* The dataset headers start from row 5 as 1-4 have information about the dataset.
* The borough column only has values of 1
* Neighbourhood has some missing values - empty cell.
* Building class category has some missing values – empty cell.
* Tax class has some missing values – empty cell.
* The EASE-MENT column is completely empty.
* Building class has some missing values – empty cell.
* Apartment number has some missing values - empty cell. Also, the name is broken up into 3 parts.
* Residential and commercial units are directly connected to total units.
* Year build has some missing values - 0.
* Sale price has some values of $0 but they are not missing as the glossary states they are a result of ownership transfer.

The .csv file was inspected before going into Jupyter because the data was able to be analysed freely better understanding was achieved.

# Data Munging

This section will detail the data munging process and the steps taken to prepare the data for the predictive model.

## Process of cleaning

The cleaning process consisted of 6 steps:

1. Functions(id\_outlier, signs & zero) – id\_outlier is the outlier detection function and is set to locate all outliers within the sale price column that have been specified with the following values (< 40000 & >1500000) when the function is called it will move them to an outlier column ready to be removed directly after. The signs function is set to handle all non numerical values within the numerical columns and convert them into int variable types as some of them have the datatype of object. The zero function replaces all 0’s within the specified columns with (np.nan) null values which prepares them to be dropped later.
2. The apartment number and sale price column names where fixed as they had syntax errors.
3. Unnecessary columns within the code were then dropped (see drop list).
4. The total unit’s column had incorrect total values so residential and commercial units are summed up again.
5. Converted sale date to date time.
6. All missing spaces are converted into (np.nan) null values.

At the end of the cleaning process the dataset is cut down to 1,413 rows and 20 columns.

## What didn’t work

In the data cleaning process data normalisation was attempted but for the linear model that was chosen it did not have a significant impact on the R^2 of the testing or training dataset, also it made the shape of (429, 6) which is too low to build an effective model.

Computing the log for the sale price column encountered multiple problems:

1. Any 0’s within the column and the log is computed it will return an infinite value.
2. When the log sale price column is used as the Y axis for the predictive model it doesn’t give any outputs within my program and the kernel gets stuck and must be interrupted and restarted.

With outlier’s if any other column besides the sale price is treated it would dramatically reduce the shape of the data-frame and have a negative impact on the R^2.

## What worked

The biggest cleaning method to influence the R^2 of the predictive model was the removal of outliers from the sale price column as it increased the R^2 from 0.28 to 0.84.

Instead of replacing the 0 values with the average it was better to just remove them entirely from the dataset.

Implementing functions to automatically clean the data was very beneficial in reducing the lines of code and now the notebook is small but still does the data cleaning just as effective.

# Modelling approach

## Why Linear regression was chosen to build the model

* Clearly shows the relationship between the sale price and predictors.
* Reliable predictive model that uses the equation y = mx + b to predict.

## Previous models and attempts

The first model that was built was SALE PRICE ~ GROSS SQUARE FEET and achieved an R^2 of 0.10 which is incredibly low whilst the intercept was -2.3, this model was not split into a training or testing set either and I managed to increase the R^2 by adding more predictors (same as the main model) however it only increased the R^2 to 0.12 with a shape of (4297, 19).

Next removal of outliers from all predictors was defined within the function however the shape was cut down too much (387, 19) which is not a good enough size to build a reliable model. After trying to sort outliers in different combinations the once that had the most impact on the R^2 was only removing outliers from the sale price (x<40000, x>1500000) which boosted the R^2 to 0.78 on that model however because it had no training or testing set that number never changed.

## Current model

The final predictive model was build using the following select fractures on the X axis [Residential unit, Total units, Land square feet & Gross square feet], then the X axis was predicted using the sale price. My data is split into a training and testing set for the predictive model and the training set has a size of 989 whilst the testing size is 424 and contains 989 samples and 20 features.

The test size is kept at 0.3 as this was producing the best outputs. Also, the estimator is kept at 5 to accommodate all the predictors. All non-predictive columns were removed until the following 5 predictors are left ['RESIDENTIAL UNITS', 'TOTAL UNITS', 'LAND SQUARE FEET', 'GROSS SQUARE FEET'].

Best output results on linear regression model:

Y-axis intercept 793378.9387

Weight coefficients:

RESIDENTIAL UNITS: 4289.3484

TOTAL UNITS: 763.0482

LAND SQUARE FEET: 21.0187

GROSS SQUARE FEET: -7.8148

R squared for the training data is 0.817

Score against test data: 0.837

Gross square feet is seen to have a negative coefficient and the linear model was ran without it being one of the predictors and it had no direct impact on the R^2 of the training or testing set so it was allowed to remain.

Below are the outputs of coefficient, std error

coef std err t P>|t| [0.025 0.975]

-------------------------------------------------------------------------------------

Intercept 8.214e+05 1.19e+04 69.065 0.000 7.98e+05 8.45e+05

GROSS\_SQUARE\_FEET -8.1034 0.204 -39.746 0.000 -8.503 -7.703

LAND\_SQUARE\_FEET 21.3236 3.387 6.296 0.000 14.680 27.967

TOTAL\_UNITS 1814.9509 1684.285 1.078 0.281 -1489.028 5118.929

RESIDENTIAL\_UNITS 2931.8528 1706.522 1.718 0.086 -415.747 6279.453

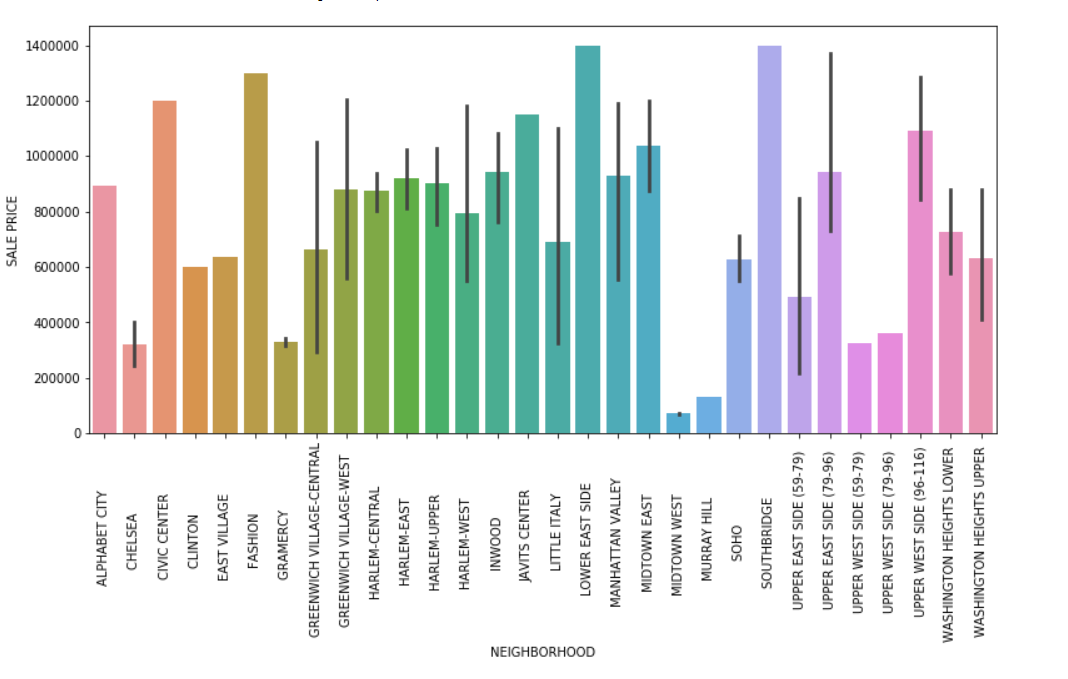
To work out the confidence interval 0.975 – 0.025 = 95

The model produced a confidence interval of 95%.

# Visualisation

After successfully building the predictive model 2 bar chart visualisations were computed.

The first was sale price across neighbourhood

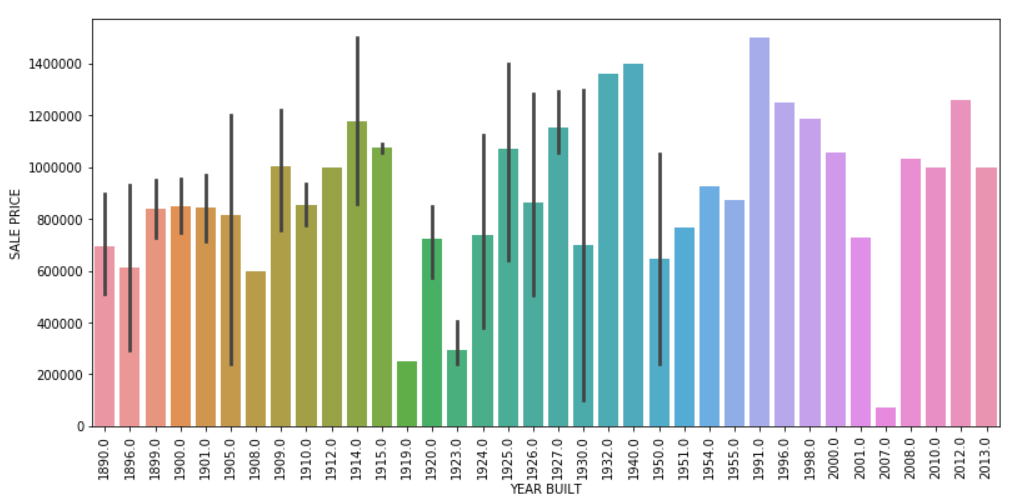


The top 3 neighbourhoods to have the highest sale price are:

1. LOWER EAST SIDE
2. SOUTHBRIDGE
3. FASHION

Although midtown west and Soho had significantly low sale prices they could be removed from the dataset as possible outliers however doing this had no effect on the predictive models R^2.

Secondly was sale price against the year the house was built



Houses build in 1991 have the highest sale price whilst for some reason in 2007 the sale price of those houses were drastically low, and this could be due to some type of mistake from the creators of the dataset.