Capstone Project Report

**Group 2:** u3253279, u3283959, u3246741, u3275711

*University of Canberra*

**Course Number:** 4483 Software Technology 1

**Tutor:** Pranav Gupta

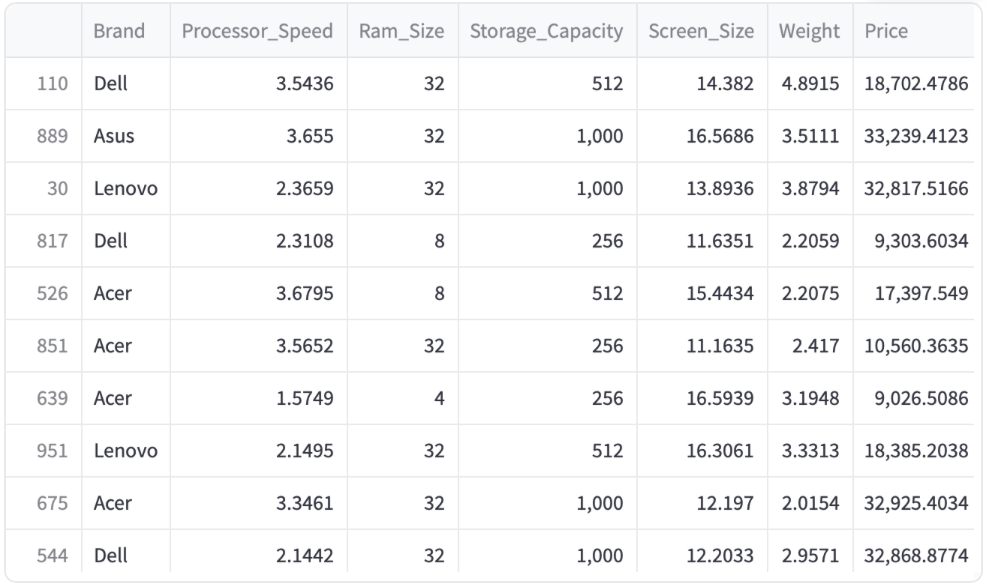
*We declare that this assignment is solely our own work, except where due acknowledgements are made. We acknowledge that the assessor of this assignment may provide a copy of this assignment to another member of the University, and/or to a plagiarism checking service whilst assessing this assignment. We have read and understood the University Policies in respect of Student Academic Honesty.*

25/10/2024

Capstone Project Report

## Part 1 – Reading the Dataset

### Sample data



Initially, the dataset was cleaned by using drop\_duplicates(), then random sample of 10 rows was pulled from the CSV file using Pandas. This sample detailed that there were 7 columns, including: Processor\_Speed, RAM\_Size, Storage\_Capacity, Screen\_Size, Weight, and Price. Note that the CSV file column names were edited for Processor\_Speed by removing an unnecessary space, and RAM\_Size by relabeling it to Ram\_Size due to issues of case sensitivity. Overall, this presented a solid initial sample snapshot of the dataset we would be working with.

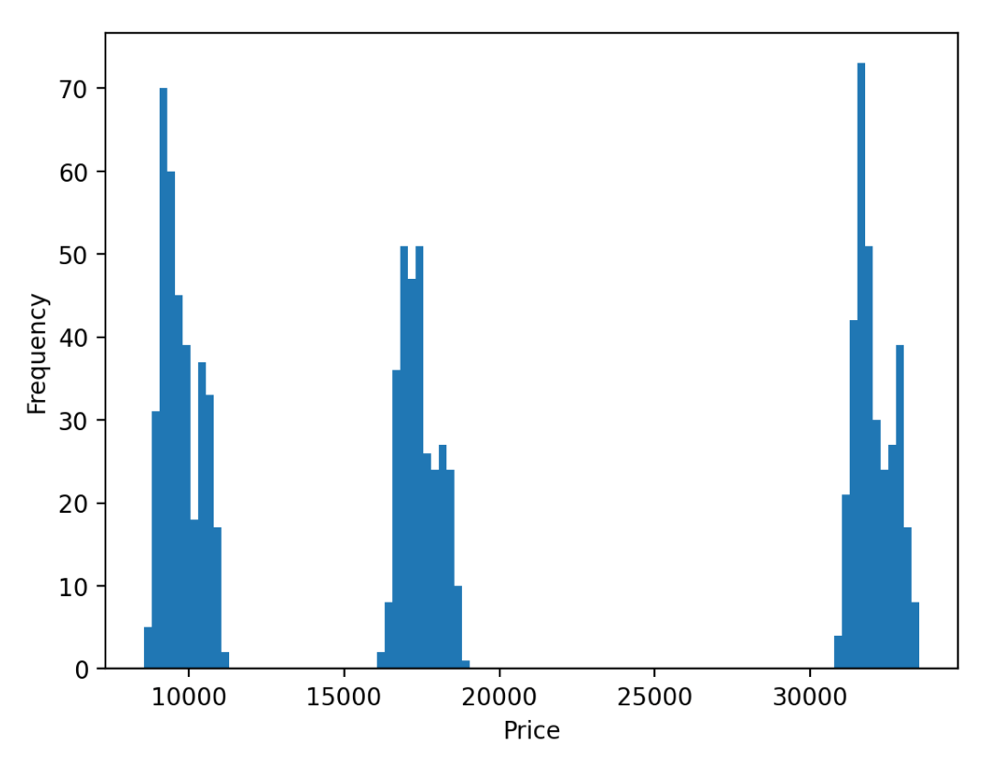
### Data Shape and Size

This dataset was further explored with .shape, outputting (1000, 7). This correlates to the rows and columns. It is evident no duplicate rows were detected as the shape remained consistent. Additionally, .size was used to confirm this, resulting in 7000 (cells) of data.

## Part 2 – Problem Statement Definition

The problem statement was derived from this data by investigating a target variable. This was deemed to be price as that is what one will predict with the data given. Therefore, a statement was formulated: “This analysis aims to investigate the relationship between laptop specifications and their price, identifying the features that have most correlation to the target variable of price in order to build a predictive model”.

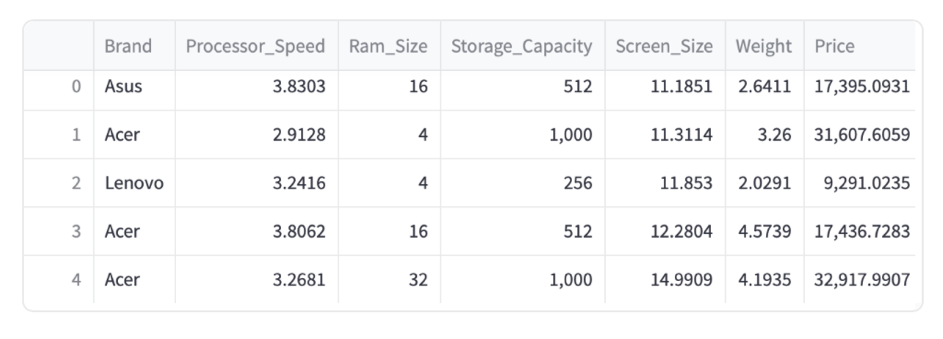
## Part 3 – Visualising the distribution of the target variable



Investigating the distribution of Price through .hist provided an interesting result. The distribution was heavily grouped around three prices: ~10000, ~17000, and ~34000, with most prices at the latter mark.

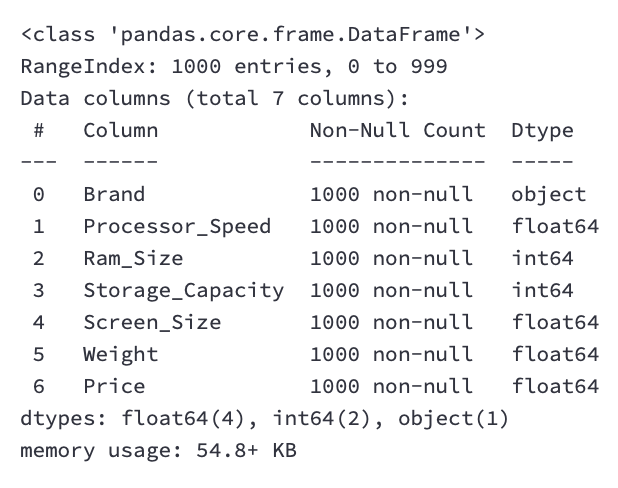
## Part 4 – Data exploration at a basic level

### Head



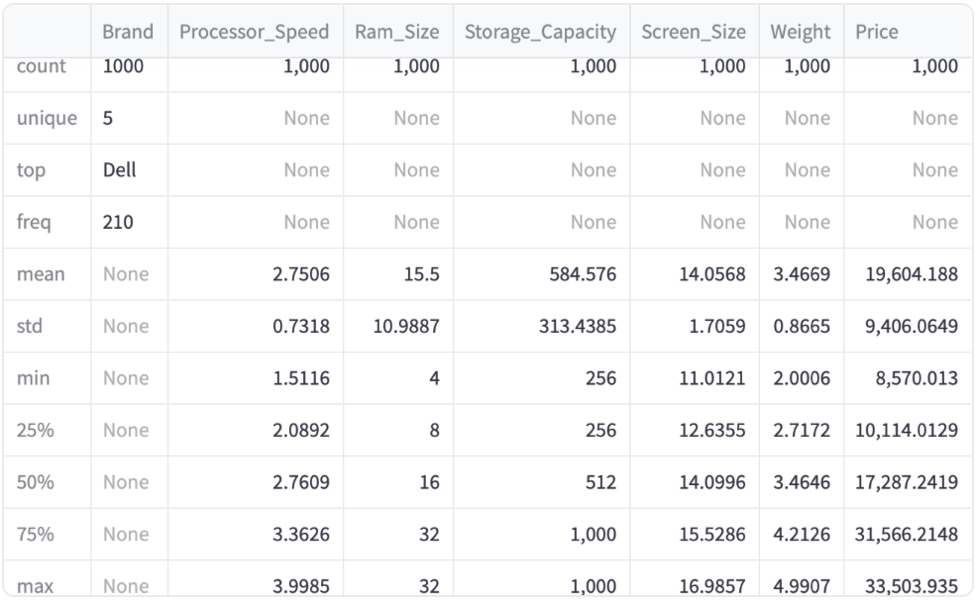
Data was initially explored with .head(), providing the first 5 rows. This did not provide any information that was not already discovered with .sample().

### Info



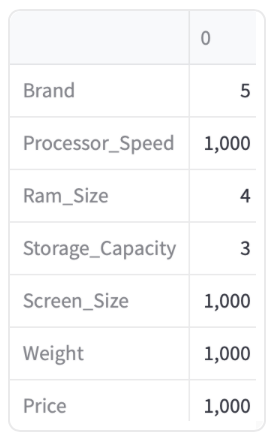
.info() was used to identify the columns, their non-null count, and type. Each column had 1000 non-nulls, meaning every cell had a value. The datatypes displayed that there was one object type (Brand), two integer columns (Ram and Storage), and the rest float values. This aligns with what is inherently expected of these categories.

### Describe



.describe() displayed valuable information about each category. Notably, the five unique brands, the minimum values, and the maximum values of the numerical columns.

### Nunique

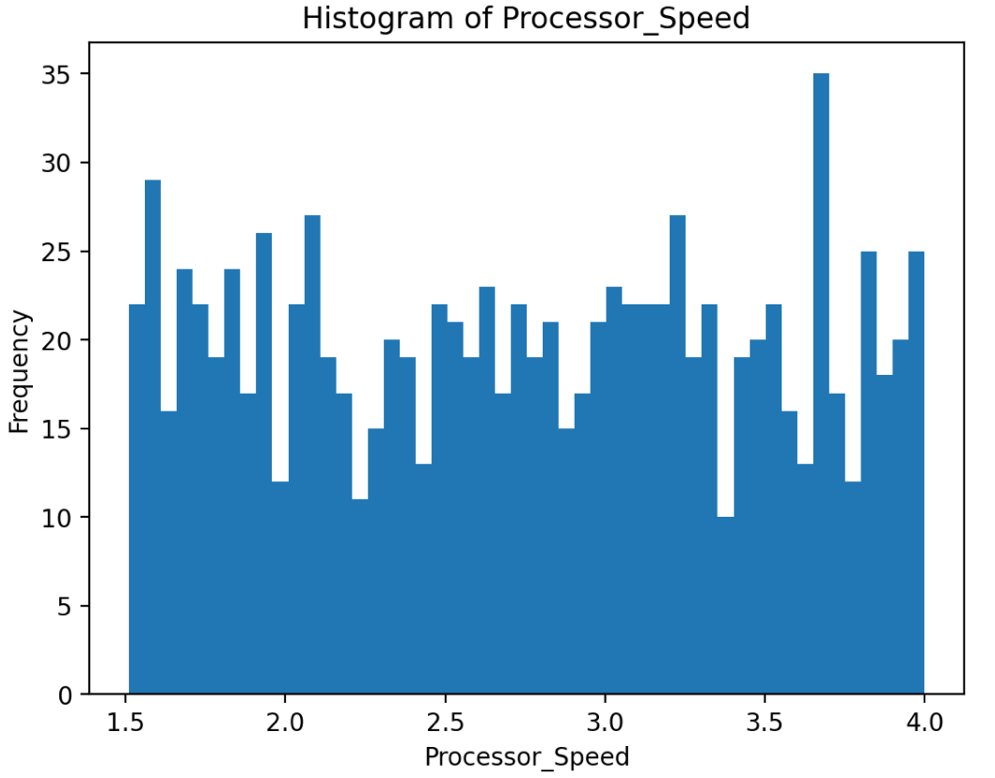


.nunique() provided the number of unique values in each column. All float values had unique values due to their precision factor. The integer value of RAM size had four discrete values, whilst storage capacity had three. Additionally, the aforementioned Brand category had five unique values.

## Part 5 – Visual Exploratory Data Analysis (EDA) of data

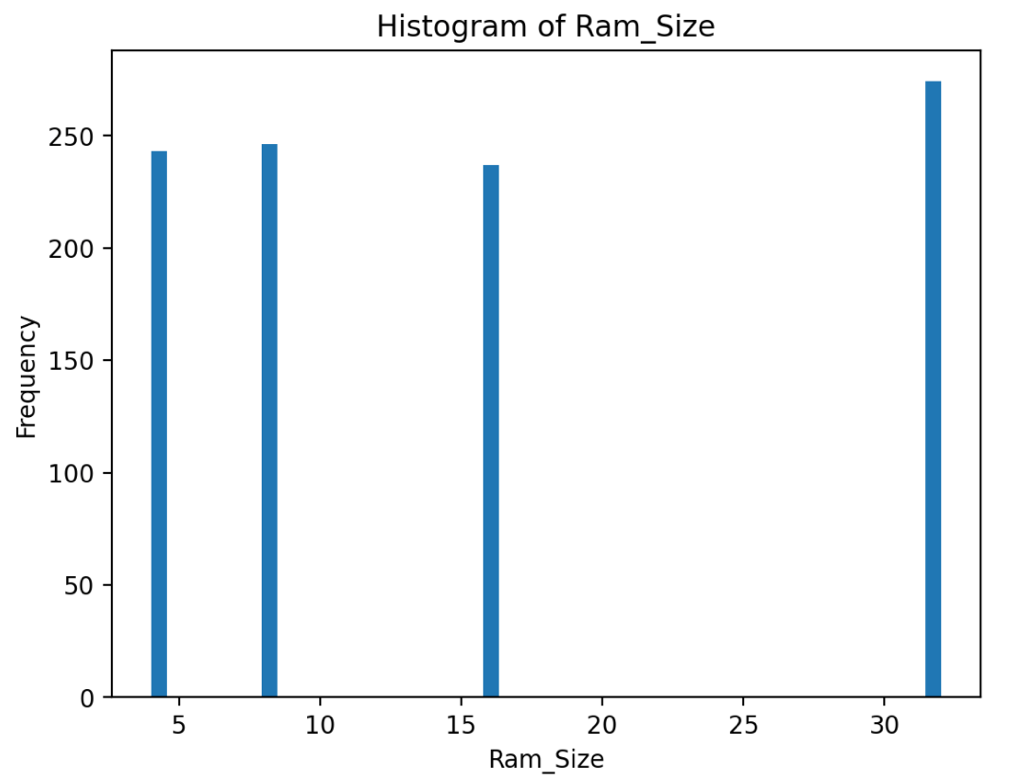
The visual exploratory data analysis (EDA) gave insight into the distribution of each variable by looping through each and producing a histogram.

### Processor Speed



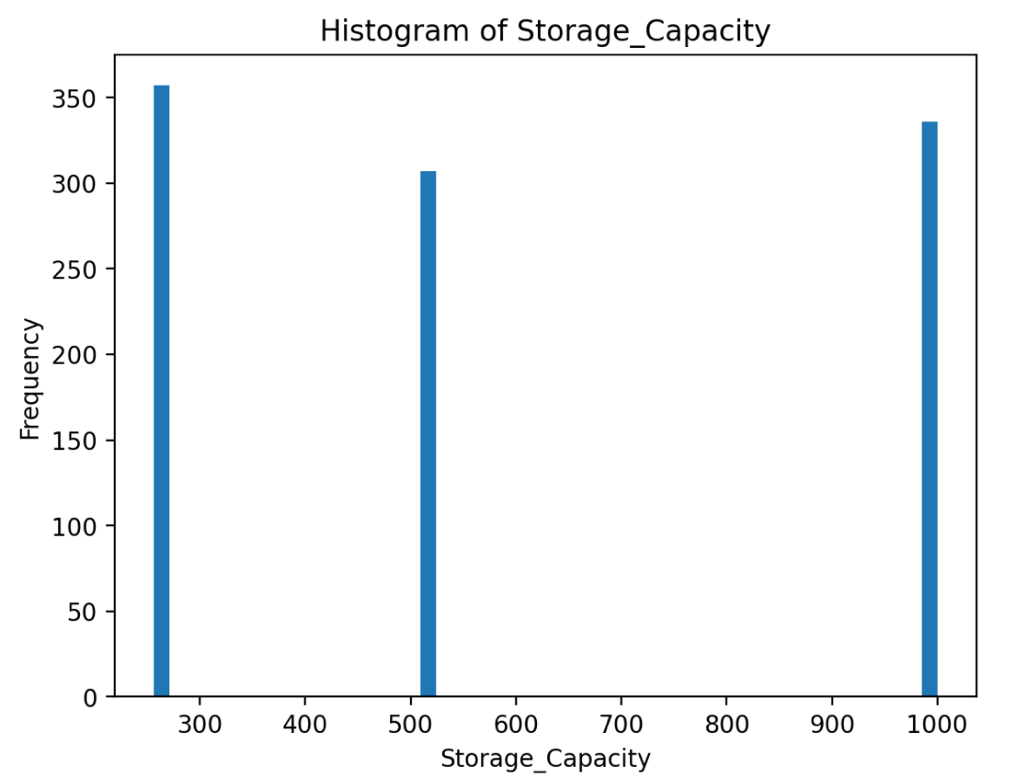
Processor speed appeared to have no correlation of the speed to how often they occur, therefore it is unlikely this would be a good predictor of price as there is no trend.

### RAM Size



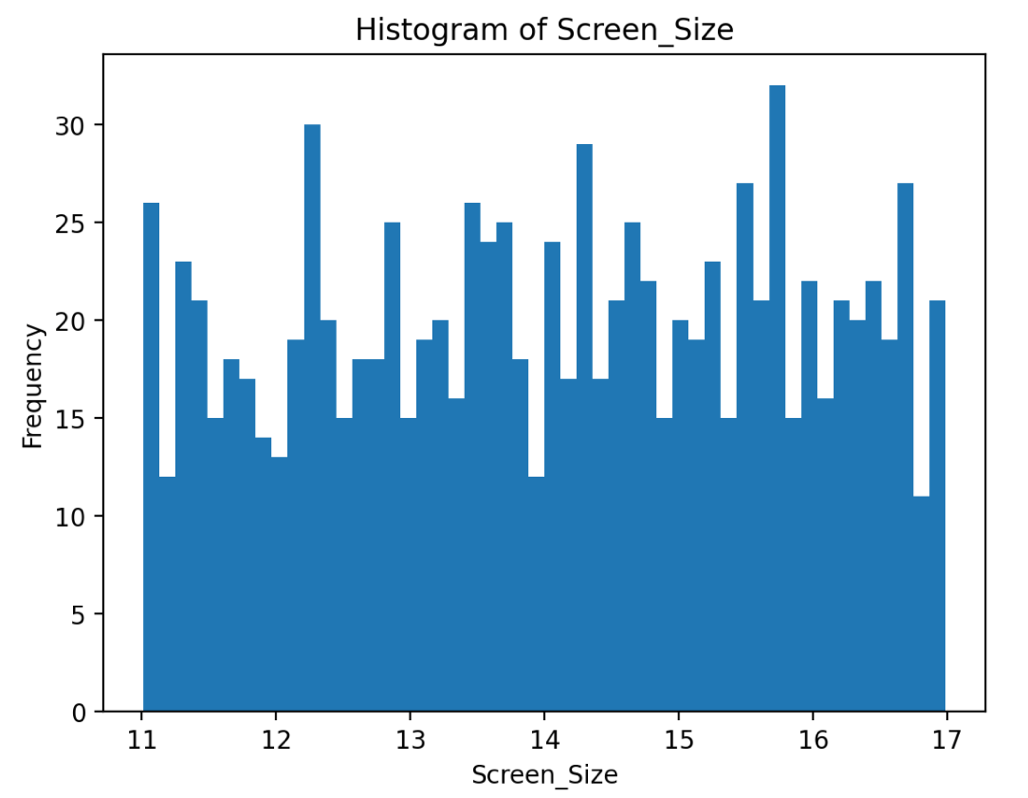
RAM Size is a discrete variable, featuring capacities of 4, 8, 16, and 32 gigabytes. This may prove to be a possible indicator of price, however, may not be accurate due to price having three primary groupings instead of four.

### Storage Capacity



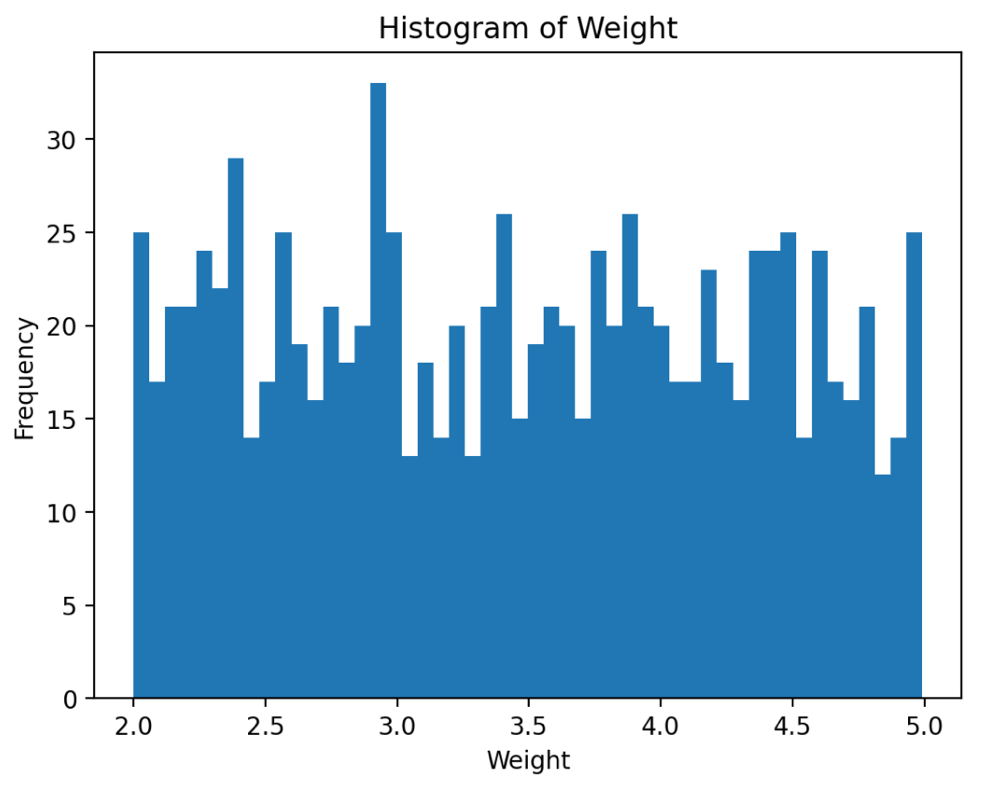
Storage capacity is another integer value that may provide a strong correlation and predictor for price. This is as it has three discrete values at 256, 512, and 1000 gigabytes which may align with the three groupings of price.

### Screen Size



Screen Size, similarly to Processor Speed, does not appear to form any distinct distribution whether bell curve or otherwise. There is little trend in this data, foreshadowing a low correlation to price.

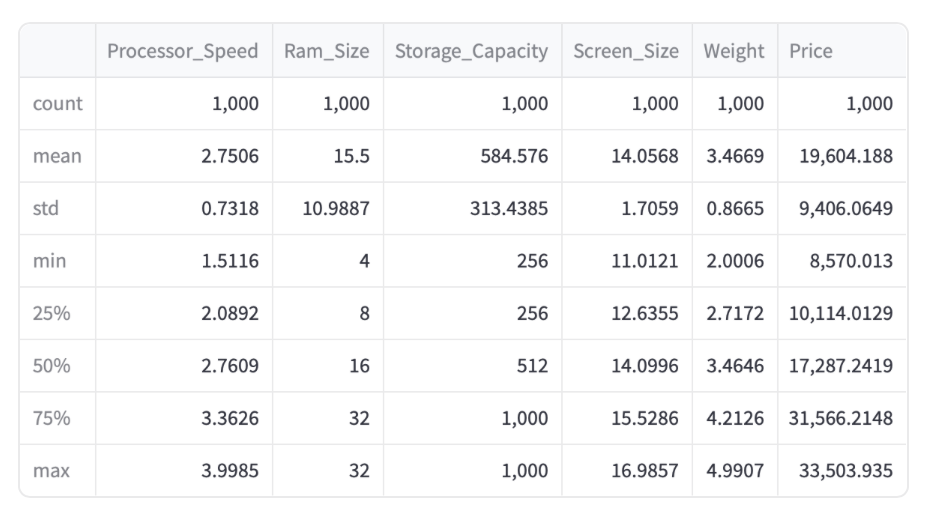
### Weight



Finally, Weight also appears to have another distribution without trend, another indicator of a poor predictor.

## Part 6 – Outlier analysis

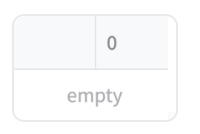
In order to remove any outliers from the data set, a dataframe of numerical-only columns was created by dropping brand. A z-score function was applied to the dataset (taking the absolute value to make all positive) and assigning it to a variable which was then used to create a dataframe where the z score was less than 3 standard deviations. Printed below is a summary of this new dataframe:



As is shown, there are still 1000 rows. This evidences that there were no outliers in the dataset, allowing the analysis to proceed.

## Part 7 – Missing values analysis

In order to determine any missing values, the dataframe was checked for .isnull() and summed each instance where there were no values. Below is the result:



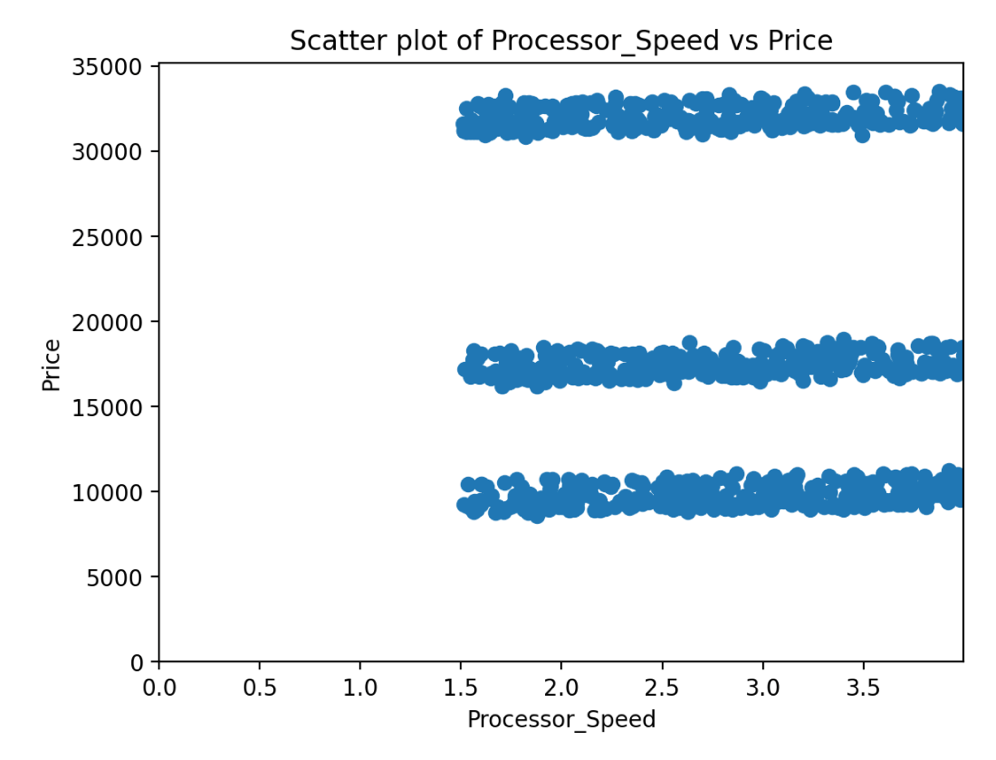
This dataframe sum is empty, once again confirming that there are no null values in the dataframe.

## Part 8 – Feature selection (Visual and statistic correlation analysis for selection of best features)

### Continuous

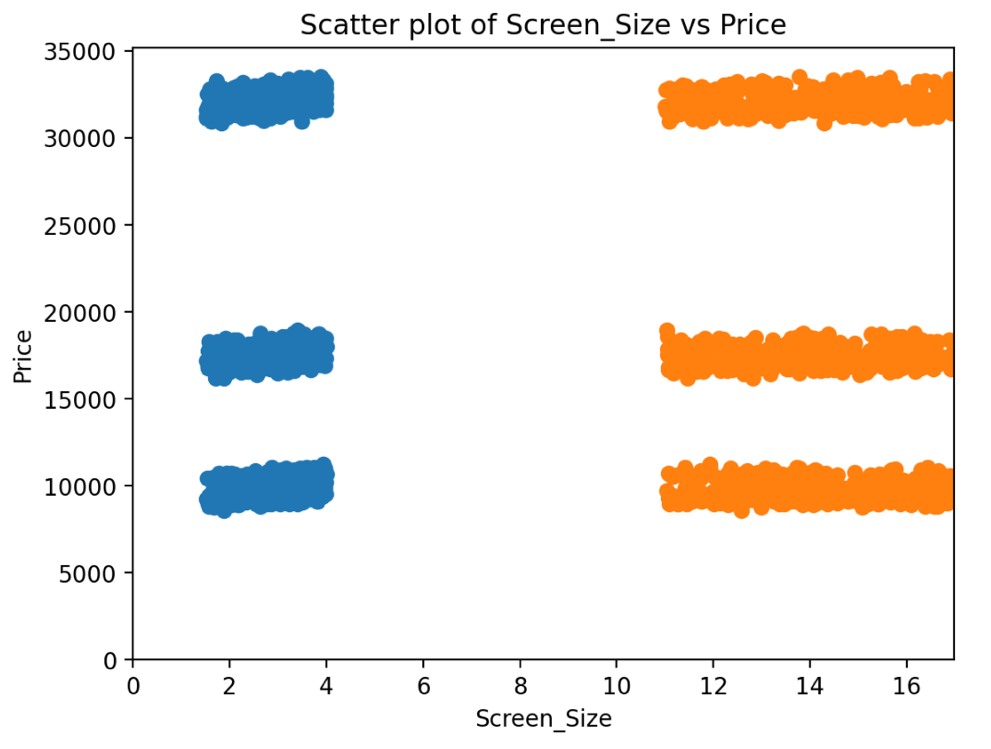
Correlations for the continuous variables against price were plotted through scatterplots, limiting the x values to the maximum of the variable.

#### Processor Speed



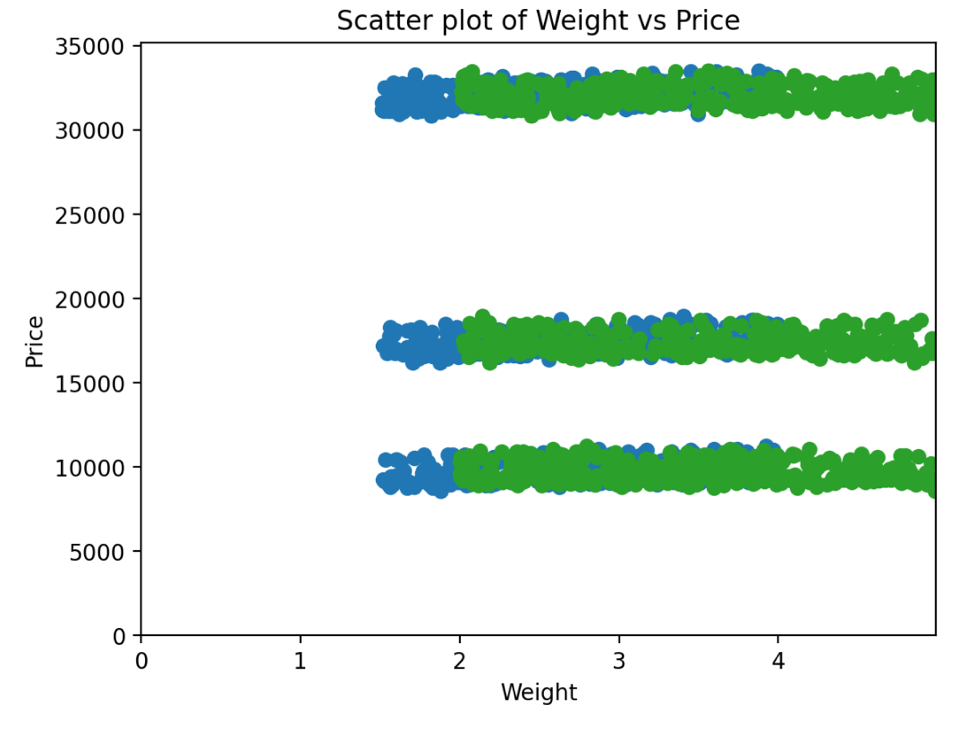
Processor speed, as predicted, had a weak correlation of -0.05 (using .corr). This is evident in the graph, as a $10000 laptop may have a 1.5GHz or 4GHz processor. This is likewise with a $32000 laptop, having processor speeds ranging from 1.5GHz to 4GHz.

#### Screen Size



Screen size had peculiar data, as screen sizes were split in a 2-4in range as well as an 11-17in range. However, just as with the Processor Speed, there is no correlation – as would be evident with a scatterplot that may follow a general line. The correlation coefficient was even smaller at -0.03.

#### Weight

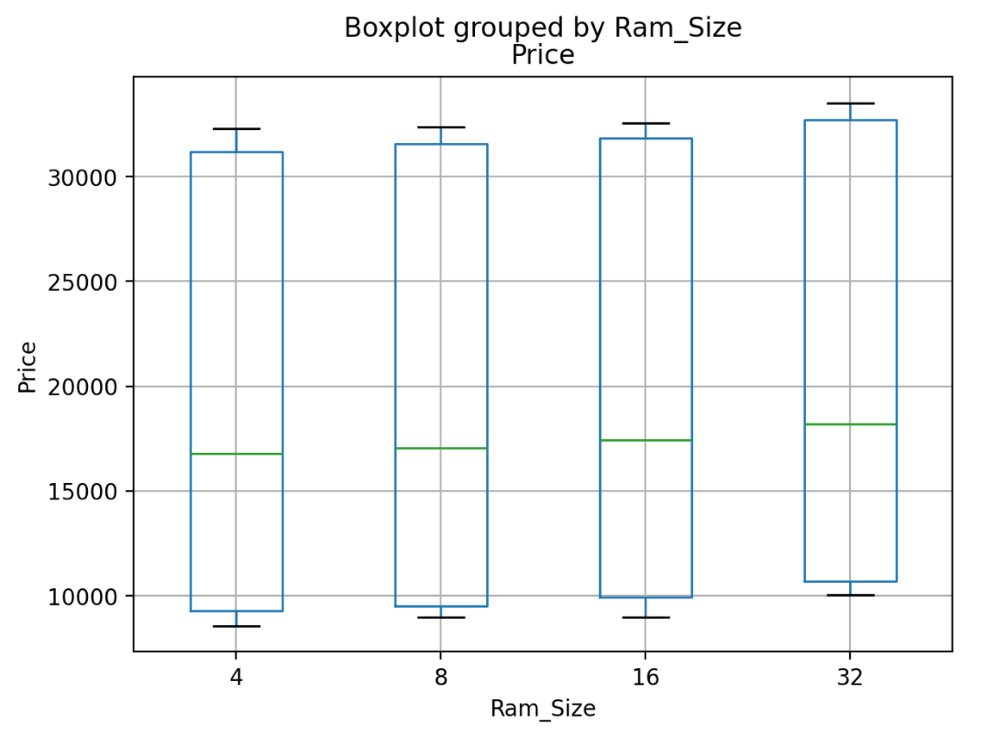


The weight correlations also provided no correlation, featuring a coefficient of 0.04.

### Integer

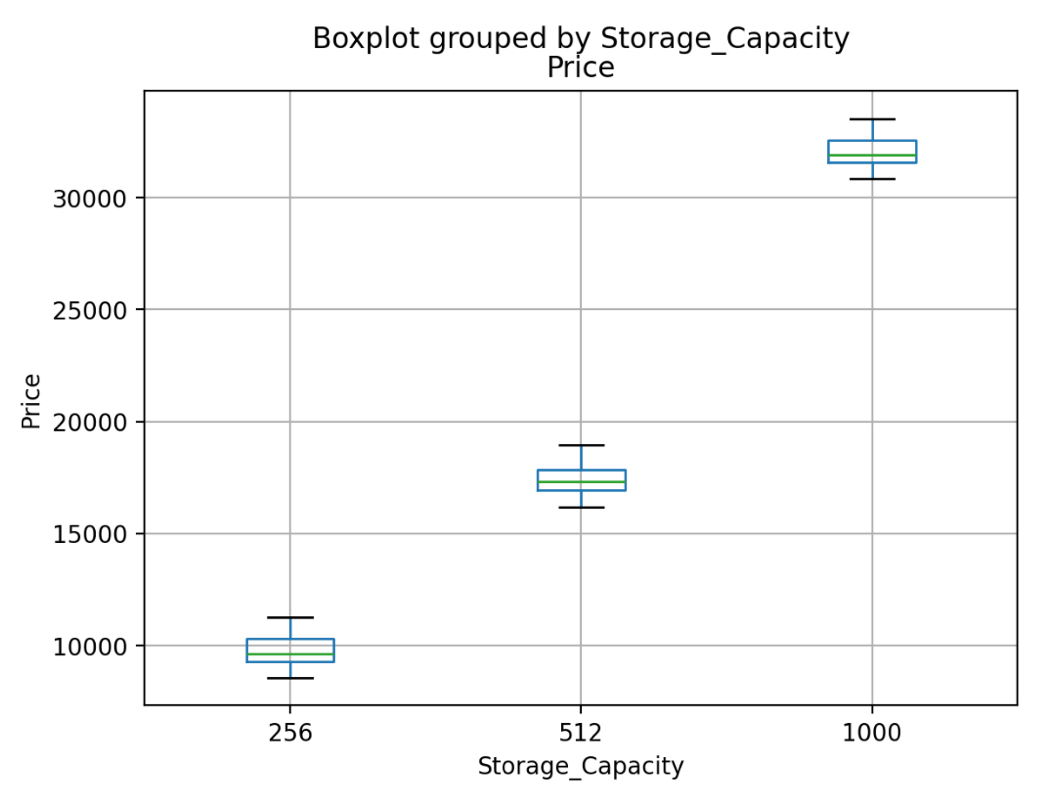
Integer value correlations to price were displayed using box plots. These are useful for the discrete values possessed by RAM and Storage capacities.

#### RAM Size



Unlike what was anticipated, RAM Size had a very large range of values for each capacity. For the most part, each capacity correlated to almost the entire range of price points, where a laptop with 4GB or 32GB of RAM could both be priced at the ~$10000 mark. The correlation value was 0.06, whilst being the most correlated so far, unfortunately is still a very weak correlation.

#### Storage Capacity



Finally, the box plot of Storage Capacity produced values that were very reliable. With low variations of maximums, minimums, and small interquartile ranges, the three capacities appear to accurately predict the grouped price points. This was also confirmed by the correlation coefficient of 1.00. Note: This was rounded to 2 decimal places, so whilst not a flawlessly accurate predictor, Storage Capacity was a very high correlation.

## Part 9 – Statistical feature selection (categorical vs continuous) using ANOVA test

An Analysis of Variance test was run, looping over each unique brand name and creating a subset of price for each. This produced a list of series of prices for each brand. f\_oneway was used to conduct the ANOVA test, storing the output in p and stat, where p is the probability of observing the stat, and stat is the ratio between the averages of the groups to the variance thereof. If these are approximately equal, then a null hypothesis is assumed. These were the results:

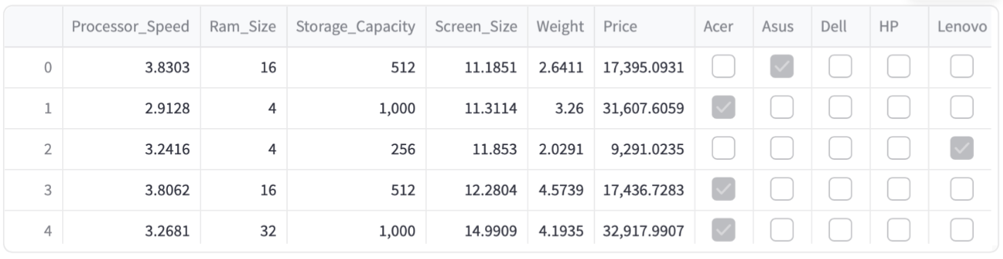
Stat: 0.55, p=0.70

Therefore, a null hypothesis was assumed as the p value was more than 0.05, suggesting the brand does not affect the price.

## Part 10 – Selecting final predictors/features for building machine learning/AI model

Despite their lower correlations, all integer and float values would be used despite the lower correlations. This is due to the lack of high correlating data in the dataset, and guidance given to the investigator from their tutor. These inclusions will be adequate to show the method of prediction, however the reliability of the outcome will be impacted. Also note, object types (Brand) will not be included due to the null hypothesis.

## Part 11 – Data conversion to numeric values for machine learning/predictive analysis

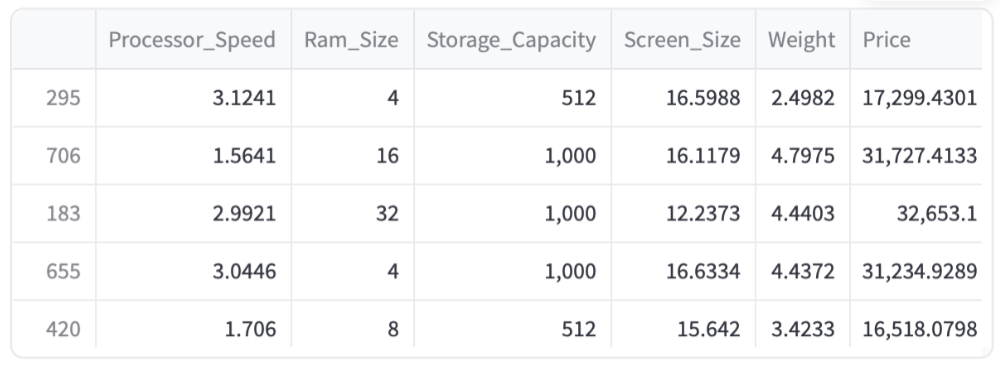


Whilst not using the categorical data, the Brands were converted to numerical values with get\_dummies() and concatenating the columns to the data frame to produce the above result.

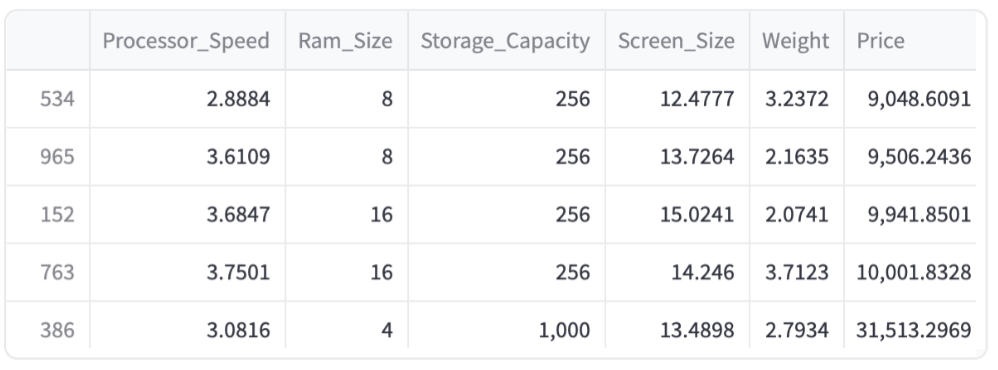
## Part 12 – Train/test data split and standardisation/normalisation of data

The data was split into a training and testing set with a test\_size of 0.5 using train\_test\_split(). This data is shown below and ready to be analysed and predicted by the models in the next step.

### Test Data



### Training Data



## Part 13 – Investigating multiple regression algorithms

Five algorithms were explored including Linear Regression, Decision Tree Regressor, Random Forest Regressor, K-Nearest Neighbour Regressor, and the SVM Regressor. These models were trained using the x and y split of data, then made predictions for the x test set, and finally determined Pearson’s correlation coefficient of the y test set and the prediction made. Below are the results for each:

### Linear Regression

Linear Regression's r2 value: 0.99954

Linear Regression's predictions per column (Speed, RAM, Storage, Screen, Weight): [$16725.51 $9060.75 $32666.60 $9357.07 $9274.16]

### Decision Tree Regressor

Decision Tree Regressor's r2 value: 0.99904

Decision Tree Regressor's predictions per column (Speed, RAM, Storage, Screen, Weight): [$16372.18 $9087.07 $32533.07 $9432.75 $9471.95]

### Random Forest Regressor

Random Forest Regressor's r2 value: 0.99938

Random Forest Regressor's predictions per column (Speed, RAM, Storage, Screen, Weight): [$16647.24 $9025.63 $32619.06 $9202.99 $9379.63]

### K-Nearest Neighbour Regressor

K-Nearest Neighbour Regressor's r2 value: 0.99933

K-Nearest Neighbour Regressor's predictions per column (Speed, RAM, Storage, Screen, Weight): [$16757.13 $9095.19 $32779.96 $9253.56 $9227.26]

### SVM Regressor

SVM Regressor's r2 value: -0.05857

SVM Regressor's predictions per column (Speed, RAM, Storage, Screen, Weight): [$17161.52 $17077.75 $17343.76 $17077.74 $17077.75]

## Part 14 – Selection of best model

The linear regression model appeared to produce the best r2 value, indicating a very strong correlation (at 0.9995) between its prediction and the actual data. This was selected to be the model that would be deployed.

## Part 15 – Deployment of the best model in production

The model was deployed in a serialized file that may be run by any user. The CSV and model.py files are available from the GitHub repository <https://github.com/tane-simons/assessment3>. Dependencies for the model include pandas, streamlit, scikit-learn, and scipy.stats. The deployment of the model is run through streamlit in order to provide a usable and streamlined interface with native support for input validation.

# References

GeeksforGeeks. (2020, May 11). *StringIO Module in Python*. GeeksforGeeks. <https://www.geeksforgeeks.org/stringio-module-in-python/>

Pandas. (2020). *pandas.DataFrame — pandas 0.25.3 documentation*. Pydata.org. <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>

pandas. (n.d.). *pandas.DataFrame.nunique — pandas 1.2.4 documentation*. Pandas.pydata.org. <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.nunique.html>

scikit-learn. (2018). *sklearn.model\_selection.train\_test\_split — scikit-learn 0.20.3 documentation*. Scikit-Learn.org. <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html>

Snowflake Inc. (2024). *API Reference - Streamlit Docs*. Docs.streamlit.io. <https://docs.streamlit.io/develop/api-reference>

Zach. (2020, July 13). *How to Perform a One-Way ANOVA in Python*. Statology. <https://www.statology.org/one-way-anova-python/>

Zach. (2021, May 31). *How to Use Pandas Get Dummies - pd.get\_dummies*. Statology. <https://www.statology.org/pandas-get-dummies/>

Software Mill. *Pros and Cons of using Streamlit for simple demo apps. https://softwaremill.com/pros-and-cons-of-using-streamlit-for-simple-demo-apps/*