Sc1015 C133 Group 5 Unofficial Report
This report is a compilation of most/all the things covered in the jupyter notebook but without the code.

Part 1) Introduction (2mins)

Hello everyone we are from group 5 of C155

The rise of smartphones has revolutionized the way we communicate and interact with the world around us. With numerous mobile phones of different brands and specifications available, it can be challenging to determine which phone is the most suitable for a user's needs. Therefore, the aim of our project is to predict mobile phone user ratings based on various variables.

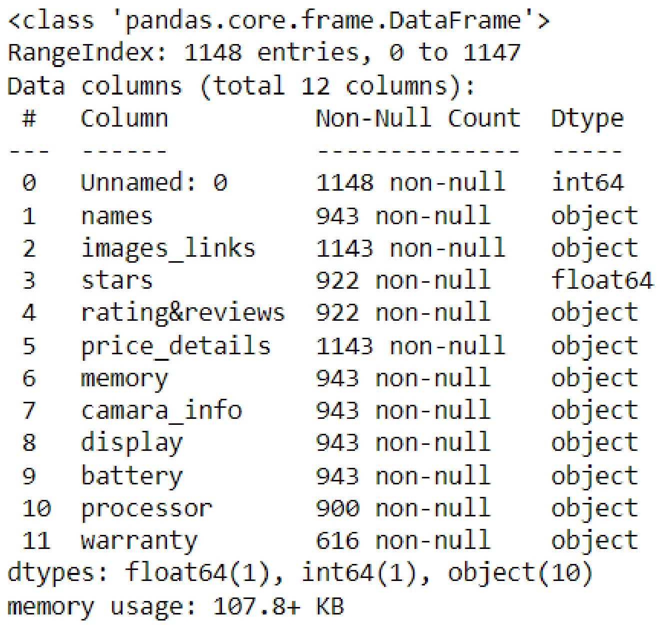
Project Objectives:

The main objective of this project is to predict how good a mobile phone is based on its variables and ratings. This prediction will assist in several ways, including:

1. Helping the brand improve based on which specifications the user feels are better: By analyzing the user ratings, we can identify which phone features the user values the most and which ones need improvement. This information will be beneficial for phone manufacturers as they can improve the phone's features according to the users' needs.
2. Help the brand predict how well their new phone will fair in the market based on its specifications.
3. Identifying the deciding factors that users look out for in a good phone: Through this project, we aim to identify the critical factors that contribute to a good phone rating. This knowledge will be useful for phone manufacturers to design phones that cater to the users' requirements.
4. Assisting users in sourcing for the phone that suits them best: By predicting the mobile phone user ratings, we can help users source for the phone that suits their needs. Users can search for a phone based on their requirements, and the model with the highest predicted rating will be the most suitable for them.

This project aims to predict mobile phone user ratings based on various variables to assist both phone manufacturers and users. It will provide valuable insights into which phone features the user values the most, the critical factors that contribute to a good phone rating, and assist users in sourcing for the phone that suits them best.

Part 2) Cleaning for each variable (1.5mins)



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1148 entries, 0 to 1147
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Unnamed: 0       1148 non-null   int64
 1   names            943 non-null    object
 2   images_links     1143 non-null   object
 3   stars            922 non-null    float64
 4   rating&reviews   922 non-null    object
 5   price_details    1143 non-null   object
 6   memory           943 non-null    object
 7   camara_info      943 non-null    object
 8   display          943 non-null    object
 9   battery          943 non-null    object
 10  processor        900 non-null    object
 11  warranty         616 non-null    object
dtypes: float64(1), int64(1), object(10)
memory usage: 107.8+ KB
```

Here is our mobile data set we got from kaggle.

Observation 1, insignificant columns

1) From the described table, we are able to see that images_links does not have any data analysis value. They are simply links for the images of the phone. therefore be dropped.
2) The warranty of the phone varies as the user is able to extend their warranty; therefore, it is not important in this analysis and will be dropped.
3) The ratings and review columns are the same for the same brand and not the rating for each different type of phone, therefore not applicable to predicting which phone is better. and will be dropped.

Observation 2: All columns except for "stars" are in string/object format.

Cleaning and formatting are needed so that the data can be analysed.

a) Names - a new column will be created that substrings the brand (categorical).
b) Price details - the original price of the phone will be extracted from the string and converted to SGD (numerical).
c) Memory - contains 3 main components: RAM, ROM, and expandable storage (SD

card) (numeric, numeric, boolean).

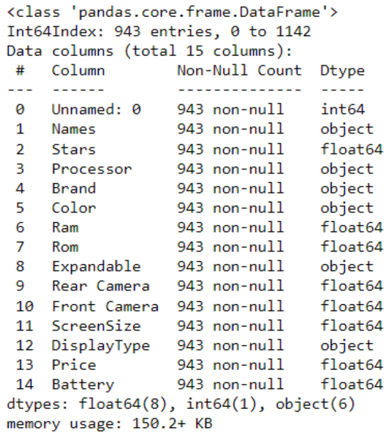
d) Camera info - the highest rear camera and front camera will be taken (numerical).

e) Display - will be split into 2 variables: size and type (numerical).

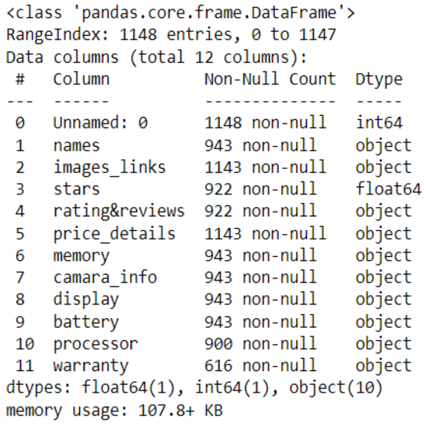
f) Battery - the number of the battery size is extracted (numerical).

g) Processor - there are a few null values which will be replaced by na (categorical).

Final columns:



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 943 entries, 0 to 1142
Data columns (total 15 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0    943 non-null    int64
 1   Names         943 non-null    object
 2   Stars         943 non-null    float64
 3   Processor     943 non-null    object
 4   Brand         943 non-null    object
 5   Color         943 non-null    object
 6   Ram           943 non-null    float64
 7   Rom           943 non-null    float64
 8   Expandable    943 non-null    object
 9   Rear Camera   943 non-null    float64
 10  Front Camera  943 non-null    float64
 11  ScreenSize    943 non-null    float64
 12  DisplayType   943 non-null    object
 13  Price         943 non-null    float64
 14  Battery       943 non-null    float64
dtypes: float64(8), int64(1), object(6)
memory usage: 150.2+ KB
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1148 entries, 0 to 1147
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Unnamed: 0      1148 non-null   int64
 1   names           943 non-null    object
 2   images_links    1143 non-null   object
 3   stars           922 non-null    float64
 4   rating&reviews  922 non-null    object
 5   price_details   1143 non-null   object
 6   memory          943 non-null    object
 7   camara_info     943 non-null    object
 8   display         943 non-null    object
 9   battery         943 non-null    object
 10  processor       900 non-null    object
 11  warranty        616 non-null    object
dtypes: float64(1), int64(1), object(10)
memory usage: 107.8+ KB
```

Final columns: a) Names

b) Price

c) RAM

d) ROM

e) Expandable (Y/N)

f) Front camera

g) Rear camera

h) ScreenSize

i) DisplayType

j) Battery

k) Processor

l) Color

m) Brand

Observation 3: The rows with Apple phones seem to have an error in their battery section.

Extra details are inserted into the rows of Apple from other sources. Data are taken from gsmarena

Observation 4: There are a lot of rows where their data is in another column.

If the rows have one data missing, the whole row will be shifted to the left. Causing the data to be in the wrong column. The data are missing from the different columns, thus we cannot simply just shift the row at once.

Observation 5: There are a lot of null values.

The total number of rows is 1148, and the rows with names and model are 943. 1148 - 943 = 205 null phones. All will be removed.

After Observation 2 is done, then we will do the exploratory analysis:

Show all the different types for categorical variables and the number of data in that category. For numerical variables, show the descriptive statistics (mean, mode), box plot, and scatter plot. For both data types, show the count of outliers. Does any of the variables have any relationship with the stars?

From the above observations, this is what we have done for the cleaning

- Null rows for name columns are removed, as the entire row is null

Original column	What you did	Outcome of your cleaning
image_link and ratings_review and	Dropped as does not have any analysis value	NIL

warranty		
processor	- For Apple brand the processor data is in the battery column. Therefore the data is shifted. - There are brands that has their warranty stored in the processor column. They are replaced with NA - Other error data are replaced with NA	Processor
names	- Substring the names column to produce the color column - Replace all the non color eg. NAI, KUNA, NANA+ with NA	Color
Memory The type of structure in this column: 1) 4 GB RAM | 64 GB ROM | Expandable Upto 1 TB	- For expandable memory, check for the word "expand" within the string. If found, set it as yes, else no. - For ram and rom	-Ram -Rom -Expandable

2) 8 GB RAM | 128 GB ROM 3) 128 GB ROM 4) NA ROM 5) 6.6 cm (2.6 inch) QVGA Display 6)8 MB RAM | 8 MB ROM | Expandable Upto 32 GB 7) 8 MB ROM | Expandable Upto 32 GB	the process is more complicated. - different row have very different way of formatting their rom and ram - There are 7 types of formatting in this column, some have the ram in the rom column. Some use MB instead of GB. Some only have ram or only have rom. - They are split according and swapped carefully	
Camera_info Contains Rear Camera & Front Camera 13Mp + AI Lens | 8MP Front Camera 13MP Rear Camera | 5MP Front Camera '50MP + 5MP + 2MP | 8MP Front Camera'	Camera is split into two column, Rear camera and Front camera by identifying “|”. The battery info is corrected by printing out the index and corrected accordingly.	

1200 mAh Lithium Ion Battery		
Display Contains ScreenSize and Screen Type. 16.76 cm (6.6 inch) Full HD+ Display 4.57 cm (1.8 inch) Quarter QVGA Display	Split the string into two column ScreenSize & ScreenType at occurrence of ‘)’. It is noted that there is “2MP rear camera” in ScreenSize column. -> Printed out the index of its occurrence. Thankfully its only one index. After double checking with excel it is replaced with the right value. Subsequently it is noted that now the data have both “cm and inch” such as “['16.76 cm (6.6 inch” and “4.57 cm (1.8 inch” We only extracting the inch. At the same time those with no data is filled with 0.	

	Subsequently the string “inch” is dropped and data type is converted to float64. Now moving on to DisplayType. Noted that there is only 942 count instead of 943, replaced the missing data with None.	
Names	To clean up the brand column of our dataset, we specifically isolated the brand name, removing all sorts of specifics such as the model names and the screen sizes from the brand column to purely focus on the brand string of the smartphone.	Brand
Price	We removed any currency symbols to isolate the pure float64 value to ensure it can be used in our model. Furthermore, since we	Price

	are focussing on the context of Singapore, we updated the price of every single price of all the smartphones to be in terms of Singapore dollar by using the exchange rate at the time the dataset was cleaned which was in March, 2023. (insert code)	
Battery	The battery wasn't perfectly in the battery column for some smartphones where the data was in neighbouring columns such as to the left of the actual battery column, the data had to be shifted to the correct column. Furthermore, since we want to work with the numerical value of the battery data in terms of float64, we removed any units that are in text and converted each battery data into a numerical data type of float64. Some of the brands like Apple had no	Battery

	battery at all thus manual research had to be done to manually enter the battery data into our dataset.	

Part 3) Exploratory analysis - each variable what did u learn from the variable alone. What did u learn plotting against stars (1.5mins)

- Present your Exploratory Data Analysis and some initial data-driven Insights from the dataset.
- You MAY also mention how you are planning to set up the Analysis / ML problem for this case.
- You MUST mention how you collected / curated / cleaned / prepared the data for this problem.
- Did you only use tools and techniques learned in this course? What ELSE did you learn / try?

After we clean and finish collating each of our parts we started analysing each variable and their relation with stars.

Part 3.1 - individual variables/predictor

3.1.1 Stars

- number of outliers for Stars: 88
- number of NULL for Stars: 21
- We see that there are a few null and outliers in Stars which may affect the prediction model later

3.1.2 Ram

- Data are generally skewed to the right, with no outliers
- 0 have the highest count for RAM and 12 have the lowest count
- from the histogram generally as ram increases the count decrease

3.1.3 Rom

- There are outliers whose Rom are above 500GB
- generally the Rom are around 0 to ~250GB

- 0 have the highest count for ROM
- unlikely to have a phone without rom storage, might need to be replaced with median or removed for a more accurate analysis

3.1.4 Expandable

- Most phone supports expandable memory (1) <600
- <400 phone does not support expandable memory

3.1.5 Processor

- The majority of the phone listed does not have their processor details, thus processor might not be the best predictor unless cleaning is done. there are very scarce phones that use the same processor

3.1.6 Color

- A large number of phones do not have a declared color
- other than NA the next highest is black
- there are too many color that might be too similar eg. blue&sea, copper&gold but we can just decide that they are the same color

3.1.7 Rear Camera

- Most of the data are 0
- Many empty data from 25-50 and 75-100

3.1.8 Front Camera

- Most of the data are 0.
- Data mainly range from 0 - 20.
- Outliers are very far from IQR.

3.1.9 ScreenSize

- Most of data are around 6-7 inch.
- From 3-6 inch onwards mostly empty data.

3.1.10 ScreenType

- Seems to have trend with different type of screen.
- Display have the highest count.

3.1.11 Price

- Too many outliers past the 1.5 of IQR while the right tail of the box plot being much longer than that of the left tail.
- This means that the majority of the price data is concentrated towards the left of the box, and there are relatively few data points with high values that are pulling the right whisker out to a longer length. This shows that prices of the smartphones range extremely from a very low price to a very high one.

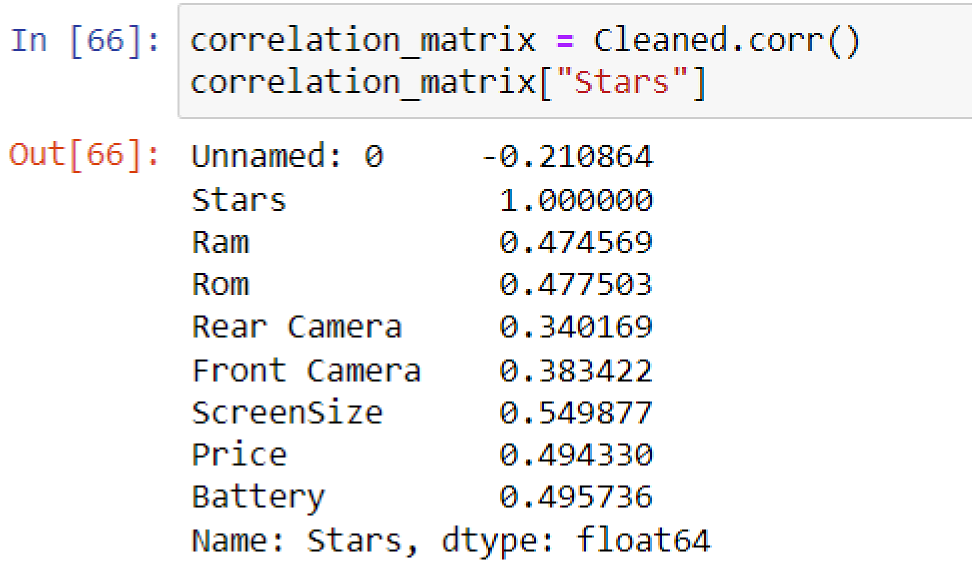
3.1.12 Battery

- The median value for battery as shown by the centre line is very much to the right. This indicates that the battery data is heavily skewed to the right with the median being about 4800 mah.
- This suggests that smartphones in our data sample happens to have much higher battery capacity than normal.

3.1.13 Brand

- Apple was the brand with the highest brand rating and the brand with the worst star rating was DIZO.

Part 3.2 - Variable relationship with Stars



```
In [66]: correlation_matrix = Cleaned.corr()
         correlation_matrix["Stars"]
```

```
Out[66]: Unnamed: 0      -0.210864
         Stars            1.000000
         Ram              0.474569
         Rom              0.477503
         Rear Camera      0.340169
         Front Camera     0.383422
         ScreenSize       0.549877
         Price            0.494330
         Battery          0.495736
         Name: Stars, dtype: float64
```

Rear and Front camera has a weak positive correlation with stars Ram, Rom, price, battery, and screen size have a moderate positive correlation

3.2.1 Ram with Stars

- from the scatter plot there is no obvious relationship
- the correlation is 0.474 which have a weak positive correlation
- Ram might be a potential predictor for stars

3.2.2 Rom with Stars

- from the scatter plot there is no obvious relationship
- the correlation is 0.477 which have a weak positive correlation
- Rom might be a potential predictor for stars

3.2.3 Expandable with Stars

- from the scatter plot there is no obvious relationship
- how every phone with expandable memory have ratings that are 3.4 & below
- Phones without expandable memory seems to have more datapoint for stars above 4.6
- Expandable memory might be a good predictor

3.2.4 Processor with Stars

- There is no obvious relationship
- NA seems to have the Stars of all values
- SC9832 have the lowest stars
- Too little data that uses the same processor. This causes the data to be very spread out

3.2.5 Processor with Color

- NA has data for all values of stars
- copper has quite a high stars
- midnight has high Stars
- The lowest stars is from champagne color
- although black is the highest color aside from NA, the ratings are generally above 3.7

3.2.6 Rear Camera with Stars

- From scatterplot seems to have no obvious relationship with stars
- Corelationship value is 0.340
- May not be a good predictor of Stars

3.2.7 Front Camera with Stars

- From scatterplot seems to have no obvious relationship with stars
- Corelationship value is 0.383
- May not be a good predictor of Stars

3.2.8 ScreenSize with Stars

- Corelationship is 0.550(>0.5)
- Possible predictor of stars

3.2.9 ScreenType with Stars

- Seems to have general trend for Stars.

- None has the highest median for Stars. - However not a good indicator as this no data.

3.2.10 Battery with Stars

- Correlation coefficient of 0.496 suggests that there is no clear strong relationship between battery and how users of the smartphones rate them.
- The circular nature of the data points is indicating that the relationship between battery and star rating is extremely weak.

3.2.11 Price with Stars

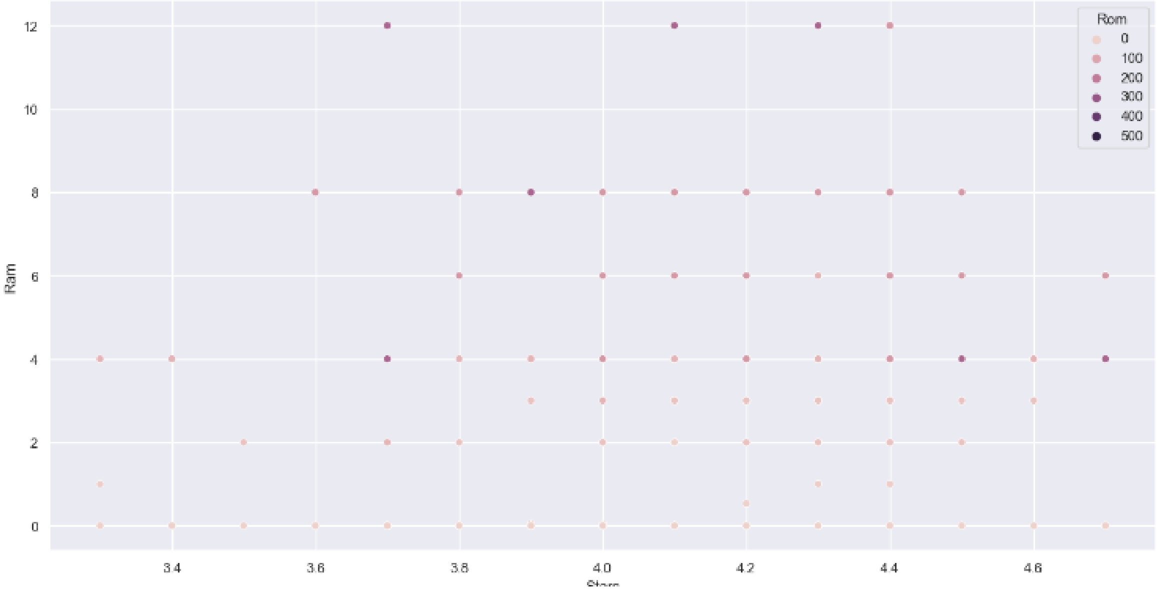
- Generally observed that there are much cheaper phones with a star rating above 4.0. It is also observed that the higher the price of a smartphone, it is much likely for it to be rated very highly.

3.2.12 Brand with Stars

- More well known brands like Apple, Google and Oppo were found to be more likely to be favourably rated.

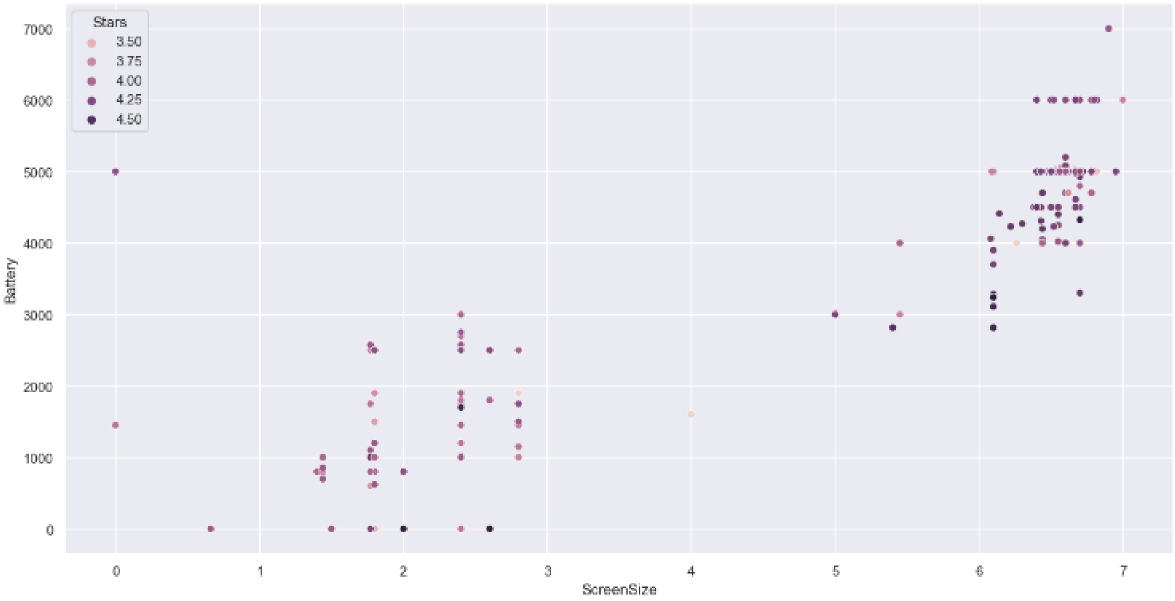
Part 3.3 - complex relationship with Stars

3.3.1 Ram & Rom Vs Stars



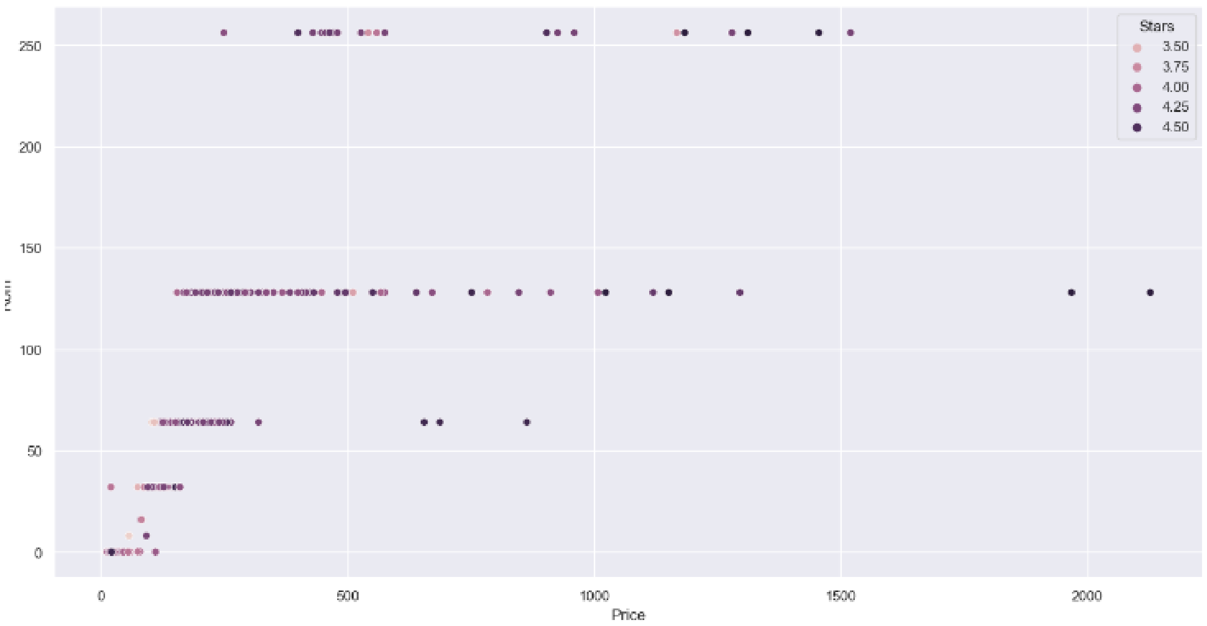
- Generally as rom increase, ram increase. however there doesnt seem to have a relation between stars
- Still no obvious relation of the stars and ram/rom

3.3.2 ScreenSize&Battery Vs Stars



- generally as battery and screenSize increase, The shade of the stars also darken depicting higher stars
- They have a linear relation
- they are good predictors for stars

3.3.3 Price&Rom Vs Stars



- generally the higher price for each rom size have higher stars
- they might be good predictors for stars

Part 4) Model (3mins)

- If you used ML (regression, classification, or something else); mention mainly WHICH one(s).
- You may now briefly CLARIFY why and how the ML problem(s) aim(s) to solve your objective.
- How did you apply ML technique(s) to SOLVE your problem? Which model(s), how and why?
- Did you only use tools and techniques learned in this course? What ELSE did you learn / try?

4.1 Model 1 K Nearest Neighbours

K Nearest Neighbours is chosen as the first model because it is considered a good model with accurate prediction. It makes predictions based on the distance between the nearest data points. It is accurate as it does not make any assumptions and adapts to changes. Therefore i believe that it is a good model to predict the Stars.

Since ScreenSize, Price, Battery, Ram & Rom got the highest correlation, they are chosen as predictors¶

4.1.1 Model 1: K Nearest Neighbours Attempt 1 (2 neighbours, 3 Predictor-ScreenSize, Ram, Rom)¶

Attempt 1: Summary¶

- used 3 predictors: ScreenSize, Ram, Rom
- ScreenSize is used as it has the highest correlation with stars
- Ram and rom is used as they are closely related to each other, therefore the graphs will have a better depict f the relationship
- NA rows is replaced with median
- the MSE is very low but we can see how we might improve by adding more predictor and more neighbours

4.1.2 Model 1: K Nearest Neighbours Attempt 2 (3 neighbours, 3 Predictor-ScreenSize, Price, Battery)

Attempt 2: Summary

- used 3 predictor: ScreenSize, Price, battery and increase the number of neighbours
- ScreenSize is used as it have the highest correlation with stars
- Ram and rom is used as they are closely related to each other, therefore the graphs will have a better depict f the relationship
- na rows is replaced with median
- the mse for test decrease from 0.05 to 0.031 which means the accuracy improved
- we will try to improve the accuracy by putting in more predictor

4.1.3 Model 1: K Nearest Neighbours Attempt 3 (3 neighbours, 3 Predictor-ScreenSize, Price, Battery)

- used 3 predictor: ScreenSize, Price, battery and increase the number of neighbours
- ScreenSize is used as it have the highest correlation with stars
- Ram and rom is used as they are closely related to each other, therefore the graphs will have a better depict of the relationship
- na rows is removed instead of replacing it with median
- the mse for test decrease from 0.036 to 0.032 which means the accuracy improved
- reason for doing so is because we are predicting the Stars, and replacing it with median doesnt seem resonable in this case as it is not like a time series where the data is in sequence.
- next i will try improving the accuracy by fitting in more predictors

4.1.4 Model 1: K Nearest Neighbours Attempt 4 (3 neighbours, 5 Predictors-ScreenSize, Price, Battery, Ram, Rom)

Attempt 4: Summary¶

- used 5 predictors: ScreenSize, Price, battery, Ram, Rom
- ScreenSize is used as it has the highest correlation with stars
- Ram and rom is used as they are closely related to each other, therefore the graphs will have a better depiction of the relationship
- na rows is removed

  - the mse for test increased from 0.032 to 0.039 which means the accuracy got worse

  - Reason for this might be because of overfitting
- next I will try improving the accuracy by reverting back to using attempt 3 and try GridSearchCV to find the best k

4.1.5 Model 1: K Nearest Neighbours Attempt 5 - GridSearchCV model for k Neighbour & using attempt 3

Attempt 5: Summary

From GridSearchCV we can see that the best number of neighbours is 15 for kNN model

- from the graph we observe that the points are more culstered
- even though the gridsearch CV says the best is 15 neighbours but the accuracy mse for the model increase
- next i will try Adding Weighted Average of Neighbors Based on Distance to see if the accuracy increase.

4.1.6 Model 1: K Nearest Neighbours Attempt 6 - Adding Weighted Average of Neighbors Based on Distance with GridSearchCV model for k Neighbour & using attempt 3

Attempt 6: Summary

From GridSearchCV with Weighted Average of Neighbors based on distancee

- from the graph we observe that the plot is very similar to attempt 5
- however the mse increased slightly from 0.028 to 0.029
- next i will try to further Improving on kNN in scikit-learn With Bagging

4.1.7 Model 1: K Nearest Neighbours Attempt 7 - Improving on kNN in scikit-learn With Bagging model

Testing with bagging model to see if there will be any improvements. Bagging is choosen before of the following reasons:

Bagging Model is used as it is a relatively simple machine learning model is fitted to a large number of different models using an ensemble method. It modifies each fit slightly. Decision trees are frequently used in bagging, but kNN works just as well.

Performance-wise, ensemble methods frequently outperform individual models. One model might occasionally be off, but on average, a hundred models should be off less frequently. The predictions will be less erratic as the errors of various individual models are likely to average out.

Therefore Bagging is used together with Knn. (Reference from https://realpython.com/knn-python/)

Attempt 7: Summary¶

From GridSearchCV with Weighted Average of Neighbors based on distancee

- from the graph we observe that the plot is very similar to attempt 5
- however the mse did not change from attempt 6
- thus accuracy did not improve

4.1.8 Conclusion for model 1

Model 1: Conclusion¶

MSE For Test dataset For Each Attempt

- attempt 1: 0.047 (2 kNearestNeighbours, 3 Predictor-ScreenSize, Ram, Rom)
- attempt 2: 0.042 (3 kNearestNeighbours, 3 Predictor-ScreenSize, Price, Battery. Replace NaN with median)
- attempt 3: 0.025 (3 kNearestNeighbours, 3 Predictor-ScreenSize, Price, Battery. Remove NaN instead into)
- attempt 4: 0.033 (3 kNearestNeighbours, 5 Predictors-ScreenSize, Price, Battery, Ram, Rom)
- attempt 5: 0.027 (GridSearchCV model for k Neighbour & using attempt 3)
- attempt 6: 0.028 (Adding Weighted Average of Neighbors Based on Distance)
- attempt 7: 0.029 (Improving on kNN Model With Bagging model)

Test Dataset	Attempt 1	Attenpt 2	Attempt 3	Attempt 4	Attempt 5	Attempt 6	Attempt 7
MSE	0.0554	0.0314	0.0377	0.0474	0.0286	0.0283	0.0326
Score	0.1177	0.3222	0.3637	0.0971	0.4105	0.4285	0.4887

Best attempt is number 6 with the lowest MSE of 0.0283.

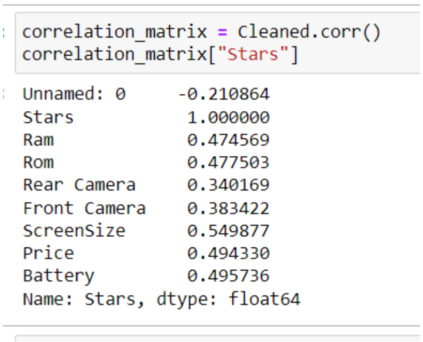
- it uses 3 nearest neighbours with 3 predictor - ScreenSize, Price, Battery

- using GridSearchCV with KNN and Adding Weighted Average of Neighbors Based on Distance¶

- this is the best outcome for the k Nearest Neighbour Model

4.2 Model 2: Linear & Polynomial Regression

Regression is suitable model as all the variable have positive correlation with stars. This means that generally, as the variable increase, stars increases. Regression is the one of the most simplest and straightforward method.



```
correlation_matrix = Cleaned.corr()
correlation_matrix["Stars"]
```

```
Unnamed: 0      -0.210864
Stars            1.000000
Ram              0.474569
Rom              0.477503
Rear Camera      0.340169
Front Camera     0.383422
ScreenSize       0.549877
Price            0.494330
Battery          0.495736
Name: Stars, dtype: float64
```

I will firstly do regression for individual variables(univariate regression) followed by multiple variables (multivariate regression).

4.2.1. Univariate Regression

Take the 5 highest correlation, to do univariate regression for screensize, rom , ram , battery & price. 4 Attempts are tested for each variable.

- Attempt 1: Linear Regression for each variable
- Attempt 2: Polynomial Regression for each variable
- Attempt 3 : Linear Regression with outliers removed for each variable
- Attempt 4: Polynomial Regression with outliers removed for each variable

Analysis:

Analysis of each variable:

- Screensize: MSE did not decrease after changing to polynomial. After removing outliers, linear mse dropped from 0.0489 to 0.0315. For polynomial it dropped from 0.04904 to 0.0315. Suggest it is heavily affected by outliers.
- Rom: MSE did not decrease after changing to polynomial or removing outliers.
- Ram: MSE decrease after changing to polynomial. However removing outlires increased MSE instead.

- Battery: MSE decreases slightly after changing to polynomial from 0.0378 to 0.0376. MSE did not decrease after removing outliers. Suggests polynomial is a better choice for battery.
- Price: MSE decreased after changing to polynomial from 0.0507 to 0.0458. MSE decreased after removing outliers with linear dropping from 0.0507 to 0.3112 and polynomial dropping from 0.0458 to 0.0298. Significant drop after removing outliers suggests it is heavily affected by outliers.

Lowest MSE for test for each variable:

- Screensize : 0.0315 linear without outlier
- Rom: 0.0315 linear with outlier
- Ram: 0.0369 **polynomial** with outlier
- Battery: 0.0376 **polynomial** with outlier
- Price: 0.0298 **polynomial** without outlier.

Highest R^2 for test for each :

- Screensize: 0.4131736150000693 **polynomial** without outlier
- Rom: 0.4131736150000693 linear with outlier
- Ram: 0.3533109179776871 **polynomial** with outlier
- Battery: 0.2843203553364382 **polynomial** without outlier
- Price: 0.3744187353074736 **polynomial** without outlier

Conclusion:

In conclusion polynomial seems to fair the best for univariate as the many of the lowest mse and highest R^2 is reached with polynomial.

4.2.2 Multivariate Regression

Since lowest MSE of 0.0298 is reached with price with polynomial regression, price is chosen as primary predictor and subsequent variables compatitible with polynomial regression is added on as predictor. The ranking of predictor is as follows: Price , Ram , Battery and ScreenSize. Polynomial and Linear regression is conducted with 6 attempts.

Summary of Multivariate

MSE for each attempt:

- Attempt 1 MSE:0.0333 (Polynomial regression, 2 predictor : Price,Ram)
- Attempt 2 MSE:0.0340 (Linear regression, 2 predictor : Price,Ram)
- Attempt 3 MSE:0.0295 (Polynomial regression, 3 predictor : Price,Ram,Battery)
- Attempt 4 MSE:0.0321 (Linear regression, 3 predictor : Price,Ram,Battery)
- Attempt 5 MSE:0.0369 (Polynomial regression, 4 predictor : Price,Ram,Battery,Scrensize)
- Attempt 6 MSE:0.0337 (Linear regression, 4 predictor : Price,Ram,Battery,Scrensize)

Attempt 3 gives the lowest MSE of 0.0295 which is polynomial with 3 predictor, Price,Ram,Battery. This is pretty close to the lowest MSE achieved of 0.0298 for univariate regression for polynomial regression of price.

4.3 Model 3: Random Forest

It is a learning algorithm that combines the results of multiple decision trees to make predictions in a more accurate manner. It is one of the best model available to make predictions of data that do not have a linear relation with one another. Price and battery data were chosen due to a relatively strong relationship with stars where the higher the price, the more likely the star rating would be as shown clearly by the scatter plot previously.

Mean squared error values and **multivariate diagrams such as heat maps and decision trees were used** to demonstrate the performance of the random forest model.

4.3.1. Random Forest Attempt 1

The heat map suggests that both predictors are equally important in predicting the target variable since the values for price and battery against star rating were found to be 0.49 and 0.5 respectively in the heat map. In other words, both predictors contribute relatively equally to the prediction of the target variable, and removing either one of them could have a negative impact on the random forest model.

4.3.2. Random Forest Attempt 2 (Replacing NA values with the median of the respective column)

The second attempt of replacing the NA values with the median of the column did not improve the performance of the model at all where the MSE value remained at 0.0339.

4.3.3. Random Forest Attempt 3 (Removal of outliers of the respective columns)

The final attempt of removing the outliers of the respective columns significantly improved the performance of the model. The heat map values for price and battery increased to 0.46 and 0.55 respectively. The MSE value reduced to 0.0328 suggesting an improvement in the prediction accuracy by the random forest data model.

4.3.4. Random Forest Attempt 4 (Using categorical data instead)

Analysis of the Random Forest Data Model:

The final attempt of removing the outliers of the respective columns significantly improved the performance of the model. The heat map values for price and battery increased to 0.46 and 0.55 respectively. The MSE value reduced to 0.0328 suggesting an improvement in the prediction accuracy by the random forest data model.

Thus in conclusion, the best MSE value obtained using the random forest model happened to be 0.32 **among the 3 attempts which was attained in attempt 3**.

Part 5) Conclusion - what we learn. Isit good? (2mins)

- What is the OUTCOME of your project? Did it solve your original problem? Anything interesting?
- What are your data-driven INSIGHTS and recommendations / views towards the target problem?

Upon comparing the three major models of k Nearest Neighbours, Linear/Polynomial Regression and random forest, our attempts indicates that K Nearest neighbours is the most accurate model with the lowest Mean Squared Error (MSE) of 0.0283 & the highest score of 0.4285. Therefore, K Nearest Neighbours is chosen as our final model for our problem statement.

We have four main interesting observations.

Firstly, we realised that, despite using a more complex model such as bagging on top of k nearest neighbours, the accuracy did not increase, thus proving that a more complex model does not neccesarily improve the accuracy of the model.

Secondly, adding more predictors does not always improve the model but, in turn, causes overfitting and reduces accuracy.

Thirdly and surprisingly, larger screen size seems to be the most prominent specification in deciding the phone's star rating.

Lastly, due to the nature of our star rating being from 3.3 to 4.7 and a single decimal place, It can be predicted as a numerical or categorical variable, which opens up more possibilities for more models.

From our analysis, we have two major insights.

Firstly, a bigger screen size and battery is positively correlated with a higher star rating. This might be because a bigger screen enables the user to have more things displayed at the same time, and the battery allow loger usage which helps to improve the user's productivity, thus placing it at a higher demand and rating.

Second, a higher sale price seems to result in a better rating, which might be due to the user perceiving a higher sale price as a margin for better quality, thus causing people to pursue a higher priced phone.

Next are our recommendations for our project and problem.

Firstly, we recommend that the phone company continue to upgrade their screen and battery qualities to have a higher rating and to stay relevant in the phone market.

Secondly, even with numerous attempts to improve accuracy, our best score is only 0.4285. which is relatively low. Thus, maybe trying a different model than those attempted might yield a better outcome.

Lastly, the lack of data in the dataset with all the NA in different predictors and NA being the majority for some predictors resulted in lower accuracy. Thus, trying a different mobile phone dataset with fewer missing data points might improve our model.

In conclusion, our model accurately predicts the Stars rating of mobile phones, and this can provide insights to companies to identify the phone specifications that impact ratings, thereby improving sales and earnings. For potential phone buyers, our model can offer useful information on phone performance based on specific features and user requirements. Therefore, we believe that our predictions using the mobile phone dataset are valuable in the world of emerging technologies.

References

1. NTUlearn
2. https://towardsdatascience.com/understanding-random-forest-58381e0602d2
3. https://realpython.com/knn-python/
4. https://www.kaggle.com/datasets/anas123siddiqui/mobiles
5. https://medium.datadriveninvestor.com/random-forest-pros-and-cons-c1c42fb64f04
6. https://www.mygreatlearning.com/blog/knn-algorithm-introduction/