Introduction:

Smartphones and mobile devices have largely become an integral part of human lives. Brands spend a huge amount of money in marketing low-quality products and compromise on product design/ development. On the one hand, they keep releasing devices one after the other throughout the year whereas the consumers are in a confused state looking at a large collection of products in the market. There are several choices for consumers today unlike in the olden days & there are a lot of cost-influencing factors like Internal memory, RAM, processor, battery capacity, display, camera pixels, etc. need to be considered before purchasing a smartphone.

Consumers look for value for money in the products and tend to spend a large amount of time on this decision-making. With too many competitors, the price marketed before a product launch is a crucial component of the outcome for any product resulting in its success or failure. It has become a great challenge for brands to retain customers, and there is a problem in identifying an optimal price for a smartphone. Therefore, there is a need for a system that predicts the prices based on the product specifications for producers and consumers to make an informed decision and lead to a successful product. Solving this problem will help both brands launch products at a competitive price in this demanding industry & satisfy customers.

In this research, the objective is to study/analyze a dataset containing details of smartphone features and their relationships for predicting its price using algorithms such as Logistic regression and K Nearest Neighbors. It is further used on unseen test data to forecast the price of the product. Multiple python libraries such as pandas, NumPy, Seaborn, Scikit-learn, and Matplotlib were used for handling multi-dimensional arrays, data manipulation, data visualization, implementing machine learning models & validating the results to get classification parameters.

Literature review:

[1] J Manasa; Radha Gupta; N S Narahari used linear regression and SVM techniques to predict housing prices with an accuracy of about 80% and they encountered an error rate of about 4% in their analysis. [2] Sameerchand Pudaruth investigated the price of used cars using different machine-learning techniques and he found that decision trees and naïve bayes could not handle target classes with numeric values. [3] Mehak Usmani; Syed Hasan Adil; Kamran Raza; Syed Saad Azhar Ali conducted a survey on stock price predictions at the Karachi stock exchange using ML algorithms such as SVM and both single layer/ MLP Artificial neural networks. They found the performance of multilayer perceptron to be the best among these algorithms. [4] Mark Holmstrom, Dylan Liu, and Christopher Vo worked on forecasting weather using Machine learning models and they came up with highly accurate results using Linear regression. [5] Nahid Quader; Md. Osman Gani; Dipankar Chaki; Md. Haider Ali have taken the historical box office data from sources like Rotten Tomatoes and IMDB, processed it and implemented Random Forest and neural networks to accurately predict the box office of upcoming movies. They concluded that the neural network was most effective with 84%. The research done by the above-mentioned authors is well suited for their datasets, but it may not be applicable to data with complex decision boundaries or too many input features often resulting in overfitting. A single algorithm cannot solve all these machine-learning problems.

Hence, algorithms such as KNN and Logistic regression are chosen for our problem with the numeric dataset to train the model with ease and retrieve consistently high-accuracy results.

Data processing:

dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	0	0	1	1
1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	1	1	0	2
1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9	1	1	0	2
0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11	1	0	0	2
0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	1	1	0	1

Figure 1. A dataset containing features of the smartphone with the price range.

The dataset has been sourced from Kaggle [6] & the labelled dataset includes information about 2000 smartphones with no missing or duplicate values. This is verified using a missingno package of python and no cleaning on the original dataset was required. There are 21 features in the dataset with 20 independent variables & price_range as the dependent target variable. price_range is represented as low-priced (0) & Expensive (3). After the data cleansing process, it is important to narrow down the columns that are relevant and useful in this modelling process to have optimal accuracy as a result. The data is split into two for the purpose of training it and testing it using the sklearn train_test_split function. The test size with a value of 0.3 indicates 70 % of the data is used for training and the rest 30 % as a testing dataset.

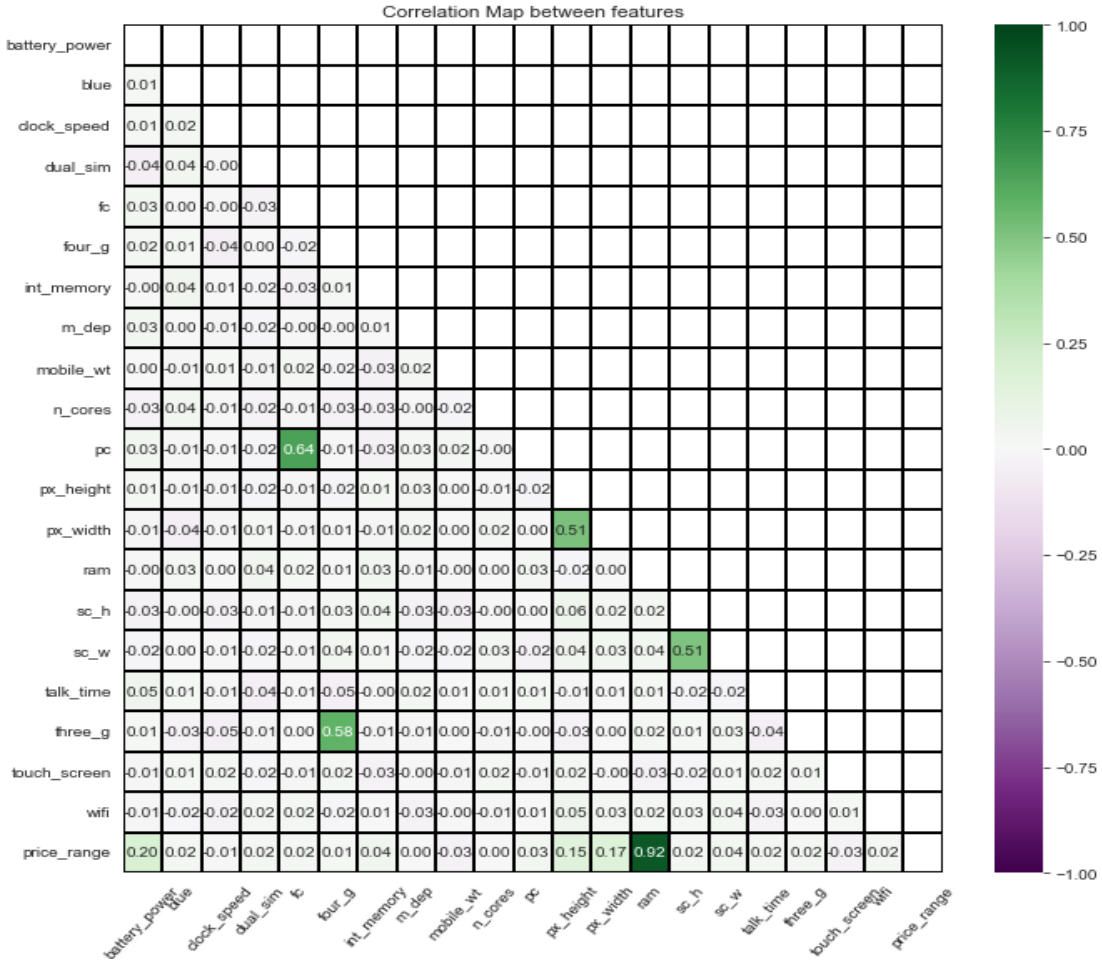


Figure 2. Correlation map between price range and features.

The correlation Map renders correlation coefficients between variables, and it is used to summarize data for more advanced analysis using the algorithms. RAM has a correlation coefficient of -1 indicating the significance of its presence in the product. Feature selection is performed using the filtering method by categorizing the input variables in numerical and categorical features. There are 13 numerical features m_dep, clock_speed, int_memory, px_width, mobile_wt, sc_w fc, ram, px_height, battery_power, sc_h, talk_time, pc and 8 categorical variables as follows four_g, n_cores touch_screen, wifi three_g, price_range, blue, dual_sim. The crucial features such as RAM, battery_power, Camera features, four_g, int_memory, n_cores, wifi, etc are selected so that model performance can accurately predict the target variable and avoid repetitive or irrelevant features. Although the model needs to train a complex dataset and it should not tend to be overfitting with high complexity.

Learning methods:

Supervised learning algorithms are chosen since our input variables are labelled and they are ideal for these mobile pricing predictions. They could train with the available features in the dataset and predict the expected price class at a faster rate & high accuracy consistently. The two supervised methods that are used are Logistic Regression and K Nearest Neighbours. The device's price depends on several influencing independent features such as Internal memory, RAM, core processor, battery capacity, display size, camera pixels, Bluetooth, etc. The logistic regression algorithm is used for this model-based classification since it could predict discrete variables quickly and performs well with large datasets. It can be categorised into three types namely Binary and Multiclass Logistic regression (Multi nominal and Ordinal). Logistic regression is very reliable as it uses a sigmoid function that helps to classify smartphones as either high-priced or low-priced. The sigmoid function is a mathematical representation that puts values from any real dataset in a binary representation.

KNN is an instance-based classifier that makes no assumptions for a given dataset and looks for the nearest data points, saving it to predict new datapoint by using similarities in existing data. KNN provides some flexibility for data sampling and is suitable for filling in missing values in datasets with noisy data. It is also known as a lazy learner algorithm as there is no training phase right away as it classifies in iterations. Data scaling is an important process for distance-based classification algorithms like KNN. These distances are measured using techniques such as Euclidean distance, Manhattan distance or hamming distance. The crucial step while working with the KNN algorithm is to find the correct value of k for the given dataset. Classification boundaries in KNN are complex as taken from saved data without any generalisation. These machine learning algorithms help in making a solid comparison of all mobile phones and their features accurately.

Since we are treating this as a Classification problem, both Logistic regression and KNN are ideal for this linear dataset. They are easier to train/test and can handle multiclass cases without compromising much on the performance. KNN and logistic regression can be regularised and cross-validated to avoid overfitting of data which will help in achieving consistently high accuracy.

Analysis, Testing, and Results:

As we got only a bit of information from this dataset, a few experiments are undertaken to analyse and get some brief ideas about the data. We got the below bar chart based on relationships between price and features. It is evident that RAM size capacity is highly influencing the price of the smartphone, followed by batteries and camera sensors.

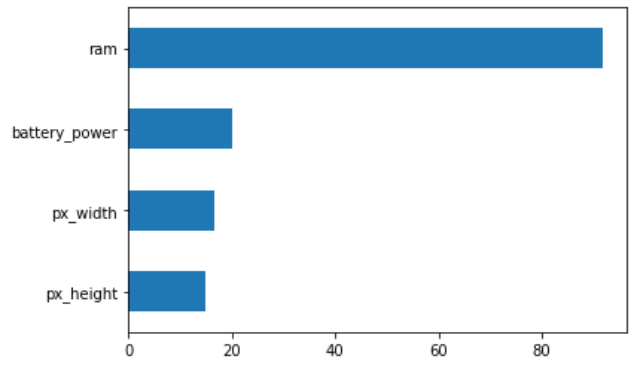


Figure 3. Features influencing overall price.

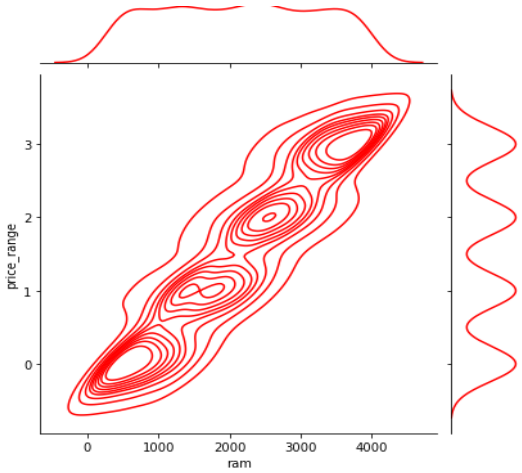


Figure 4. Effect of RAM on price range.

The dataset has been scaled using standard scalar during pre-processing and both Logistic regression and KNN models were run for the processed data. We could see that both the models classified the price of the mobile phone with good accuracy as shown in the table. Accuracy score, Confusion matrix and parameters such as are the metrics used to evaluate the efficiency of the ML algorithms. We further run the model for some iterations, and it is tested for the unseen test dataset. The accuracy results are obtained for both the train and test datasets as shown in Figures 5 and 6. In these supervised algorithms, Logistic regression has happened to predict the price range of mobile phones slightly more accurately compared to KNN.

	ML Algorithms	Accuracy %
0	Logistic Regression	0.945000
1	KNN	0.933333

Figure 5. Accuracy of train dataset.

```
LogisticRegression
Accuracy of Train DataSet:94.85714285714286
Accuracy of Test DataSet:93.33333333333333
```

Figure 6. Train & Test accuracy after several evaluations

It is important to analyse the cause of price increases in certain sets of smartphones. Batteries have become largely bigger and more efficient due to their unique power storage component known as Lithium. This allows an extended device screen on time (SOT) on a single battery charge and Lithium allows the large capacity battery to be charged at a rapid speed within 20 minutes from 0-100%. Supporting smartphones with fast charging has made manufacturers put their bet to make a successful product. The relationship between battery power and price range in Figure 7. shows that expensive mobile phones are equipped with high-power batteries. Because of the Covid pandemic & lockdowns, there has been an imbalance in the demand/supply of semiconductor design/manufacturing. This has put tension on the electronics supply chain leading to a shortage of these chips that require niche expertise

to build this hardware. RAM & wireless transmitters like 4G and Wi-Fi modems have become a necessity today, these shortages are increasing the prices of the products as indicated in Fig8.

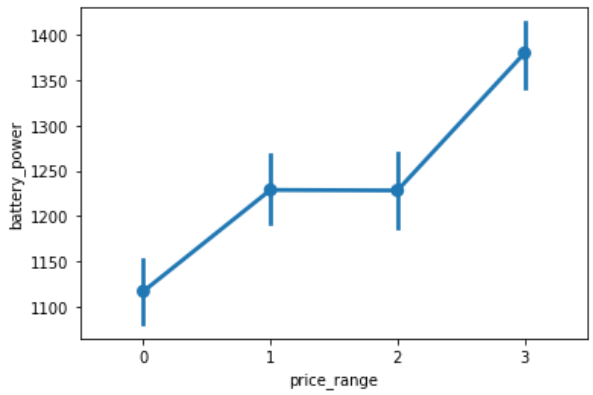


Figure 7. Effect of Lithium batteries on price range

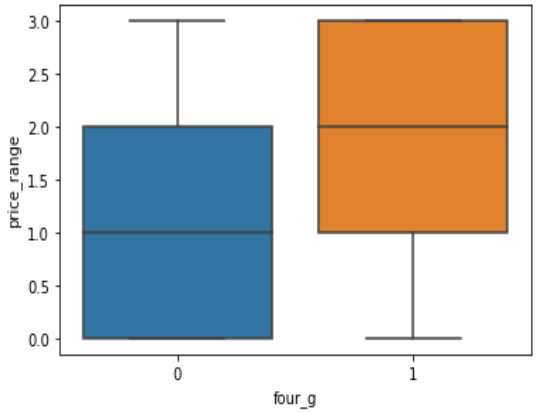


Figure 8. Effect of 4G modem on price range

Cross-validation is done for assessing the efficiency of the model and to avoid overfitting. Stratified k-Fold Cross Validation is performed by taking k=10 for 10 splits of the dataset and the scores have been obtained as indicated in the bar chart. The cross-validation score of the logistic regression model is 97% and the score for the KNN algorithm is 93.5%. The model efficiency can be significantly improved if there are more records in the dataset for it to train better and perform well with unseen data as well.

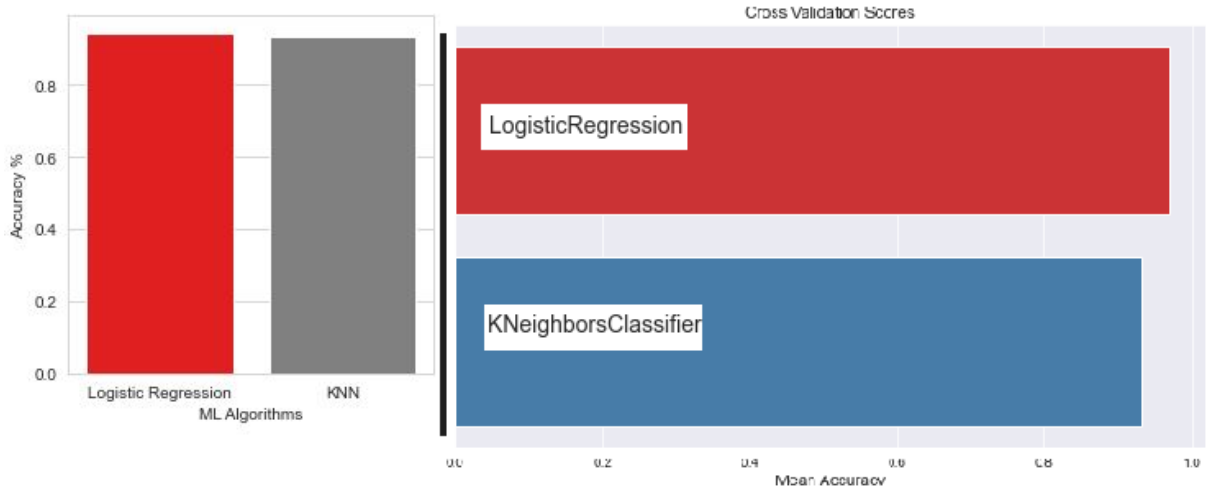
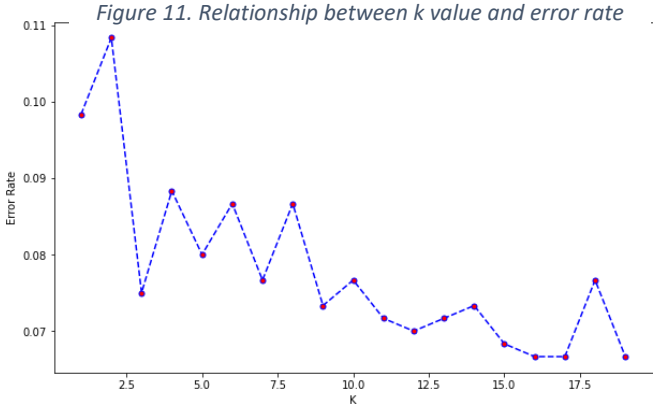


Figure 10. Accuracy and cross-validation bar graph for Logistic regression & KNN models

Figure 11. Relationship between k value and error rate



As we could find from Fig. 11, lower values of k are having high error rates while the error rate is very low when the K value is high. Choosing the right value of k was a challenge for the given data and KNN had difficulty in classifying distant data points, accuracy was low for smaller values of k.

Conclusion:

In conclusion, we were able to note that both Logistic regression and KNN had their own strengths/ weakness. Logistic regression was quicker to train with our mid-sized dataset and suited for problems with linear boundaries, but it has its limitations to classify with large complex datasets. In contrast, the challenge I faced with KNN is that it was a bit slow with its predictions due to its one-by-one iteration of data points and impacting the model performance for large datasets. Having said that KNN can be used for regression problems as well, where Logistic regression will not be very useful. It is found that a single algorithm cannot solve all these machine-learning problems and each problem needs its own study, data processing, analysis, prediction, regularisation, etc.

The consistent performance of the classification model is measured with the confusion matrix, it has precisely gathered instances in the diagonal of the matrix and wrongly classified instances are in the rest of the 4x4 matrix. The below confusion matrix of Logistic regression indicates that we have correctly predicted the price of 140 devices with a range of 0 class level and misclassified 4 samples. Similarly, the same is noted for price ranges 1(Medium), 2(High), and 3(Expensive) classes.

```
Model: LogisticRegression(random_state=101)
Accuracy: %97.0
MSE: 0.04
ROC AUC: 0.9986023478673542
Recall: 0.96
Precision: 0.9607577849574407
F1-Score: 0.9598402316352899
Best Estimator: LogisticRegression(C=29.763
========================================
Fitting 10 folds for each of 360 candidates
Model: KNeighborsClassifier()
Accuracy: %93.5
MSE: 0.056666666666666664
ROC AUC: 0.9950819494884469
Recall: 0.9433333333333334
Precision: 0.9448233157205306
F1-Score: 0.9433951983418901
Best Estimator: KNeighborsClassifier(leaf_size=1, metric='manhattan', n_neighbors=18,
                     weights='distance')
------------------------------------------------------------------------
```

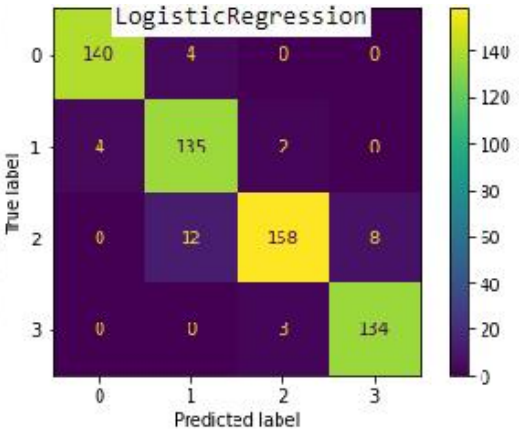


Figure 9. Classification Performance Report of the Logistic regression & KNN Models

The sklearn. metrics module is implemented to calculate metrics such as Recall, Precision, F1- score, Accuracy, and AUC-ROC curves are evaluated as shown in Fig 9. The precision score measures the number of correct positive predictions made with overall positive predictions. For logistic regression, it is 96.07% slightly higher than KNN with 94.4%. Adding to that, the Recall score shows measures the number of correct positive predictions with respect to total positive instances in the dataset itself. Logistic regression performed better again with 96% compared to KNN with 94.3%. F1-score is a harmonic mean of both precision and recall

scores, they also indicate better performance of Logistic regression over KNN. In the aspect of mean squared error, Logistic regression has a lower error rate than the KNN as noted above.

The computational time complexity would also be the right parameter to assess the performance of the algorithm. For logistic regression with 2000 data samples, the Big O notation for training time complexity can be represented as O (1400*21) and test time complexity would be O (600*21). On the other hand, KNN doesn't have a training phase, all computational time complexity during prediction is O (1) with the brute force approach and using the K-d tree it is O (1400*21*Log (1400)).

In this activity, we had taken hardware information from only a few specific manufacturers and found that RAM has the most influence on the pricing, but the business in the market works differently in reality today. Each brand has target consumers who purchase products from 200 GBP to 1200 GBP which in turn offers value in a range of areas including software and security updates. Software drives these businesses today and has a greater influence on pricing now. Therefore, the model must be fed with more real data by including details software/security versions to be useful for a larger set of consumers. On top of this, this tool can be extended with several other businesses/products like laptops, automobiles like electric cars, smart wearables, housing/real estate etc so that there is always transparency about the quality of a product and its pricing to the public.

References

1. https://ieeexplore.ieee.org/abstract/document/9074952

2. https://ripublication.com/irph/ijict_spl/ijictv4n7spl_17.pdf

3. https://ieeexplore.ieee.org/abstract/document/7783235/

4. https://cs229.stanford.edu/proj2016/report/HolmstromLiuVo-MachineLearningAppliedToWeatherForecasting-report.pdf

5. https://ieeexplore.ieee.org/abstract/document/8281839/

6. https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification?datasetId=11167

7. https://www.ijcaonline.org/archives/volume179/number29/asim-2018-ijca-916555.pdf

8. https://ieeexplore.ieee.org/abstract/document/10007128

9. https://ieeexplore.ieee.org/abstract/document/9604550