**Mobile Price Data Classification: A Comparative Study of SVM, Random Forest, and Neural Network Models**

Mohammad Nassar, Ahmad Ghannam, and Mohammad Jomaa

Department of Computer Systems Engineering, Arab American University, Jenin, Palestine

[m.nassar4004@gmail.com](mailto:m.nassar4004@gmail.com)

[iia7mt@gmil.com](mailto:m.nassar4004@gmail.com)

[mjomaa2223@gmail.com](mailto:m.nassar4004@gmail.com)

# ***Abstract***

*A mobile phone is now one of the most ubiquitous consumer goods. Every year, thousands of new mobile phone models are introduced, each with its own set of features, specifications, and designs.* *In this study, we present a comparative analysis of three machine learning models, namely Support Vector Machines (SVM), Random Forest, and Neural Networks, for the classification of mobile price data. Machine learning algorithms are utilized when working with data that includes a predefined class label, which represents the attribute to be predicted. In the context of mobile price classification, the Mobile Price Class dataset obtained from Kaggle and we apply several algorithms to it in order to simplify it, highlight the most important elements for making a decision, and get the most accurate comparison possible within the dataset.The highest quality at the lowest possible price may be found with this tool. Python was chosen as the programming language due to its extensive machine learning libraries. Multiple classification algorithms were employed to train the model, aiming to identify the most accurate algorithm for predicting mobile price classes. The objective is to develop accurate models that can effectively predict the price range of mobile phones based on their features.*

# ***Keywords***

*Mobile price classification,      SVM,      Random Forest,      Neural Networks,      Machine Learning*

1. **Introduction**

In today's competitive market, pricing plays a pivotal role in product marketing, often determining the success or failure of a product. Therefore, having a reliable tool that can estimate prices based on various product features can be immensely beneficial for both companies and consumers. Such a tool not only aids decision-making in pricing strategies but also assists consumers in selecting the most suitable product within their budget. In this study, we focus on classifying mobile phone prices based on a range of features such as RAM, battery capacity, camera specifications, display size, and thickness. Categorizing mobile phones into distinct price ranges, such as low, medium, high, or very high, enables consumers to understand the value proposition of different devices and make informed purchasing decisions. For manufacturers, accurate classification of mobile phone prices helps them understand the key features that influence pricing, thereby driving advancements in the industry and aligning their product offerings with market demands. To achieve our objectives, we employ several popular machine learning models, namely Support Vector Machines (SVM), Random Forest, and Neural Network models. These models have been widely used in classification tasks and have demonstrated their effectiveness in various domains. By training these models on a comprehensive dataset sourced from Kaggle [1], which includes definitive class labels for price ranges, we can leverage supervised learning algorithms to make accurate predictions based on the provided mobile phone features. In our analysis, we specifically evaluate the performance of these three machine learning models using key metrics such as accuracy, precision, recall, and F1 score. Accuracy measures the overall correctness of the model's predictions, while precision quantifies the proportion of correctly predicted instances in a specific price range. Recall, on the other hand, evaluates the model's ability to identify all relevant instances within a price range, and the F1 score combines both precision and recall to provide a comprehensive evaluation of the model's performance. Based on our extensive experiments and analysis, we found that the neural network model outperformed the other two models with an impressive accuracy of 97%. This highlights the power of neural networks in capturing complex patterns and relationships within the mobile phone dataset, enabling highly accurate price range predictions. The superior performance of the neural network model establishes it as the most effective algorithm for classifying mobile phone prices among the ones evaluated in our study. By presenting these findings, our research provides valuable insights for consumers, empowering them to make more informed purchase decisions. Additionally, manufacturers can leverage our study to gain a deeper understanding of the specific features that significantly influence mobile phone prices. Armed with this knowledge, manufacturers can make data-driven decisions, align their product offerings with consumer expectations, and drive advancements in the mobile phone industry. In conclusion, our study demonstrates the effectiveness of machine learning algorithms in classifying mobile phone prices based on various features. We highlight the superiority of the neural network model, which achieved an accuracy of 97%, making it the ideal choice for predicting price ranges. Our research contributes to the existing body of knowledge in the field of pricing and product marketing, providing valuable insights for both consumers and manufacturers alike.

**Literature Review**

Ibrahim nasir, muhammad al shawa, and sami saad abu nasir published [2] a paper titled “development of an artificial neural network for predicting mobile phone price range,” And they used a dataset referring to abhishek sharma. Use the data to predict the price based on the neural network and its learning algorithm. Their forecasting average accuracy obtains 96.31%

In anil kukret's [3] paper titled "The application of machine learning algorithms to a dataset of mobile phone prices for classification," The author collected data from their environment based on predetermined criteria. They employed multiple models for classification, and the random forest regressor model demonstrated the highest accuracy. The study identified screen size, battery capacity, internal memory, and ram as the most significant features for predicting mobile phone prices.

In the paper titled "Mobile price prediction using ml algorithms" By gautam dhall and dr. S.Nithiya,[4] the authors obtained data from kaggle. They employed support vector machines (svm) for classification and achieved an average accuracy of 95.00% in their predictions.

In a paper titled "Prediction of mobile phone price class using supervised machine learning techniques" By varun kiran and dr. Jebakumar r [5], the mobile price class dataset sourced from kaggle was used. The study evaluated several machine learning algorithms for classifying instances based on the price range. Lda achieved the highest accuracy of 95%, followed by knn with an accuracy of 92.75%. However, decision tree and naïve bayes classifiers had lower accuracies of 75.75% and 52.25% respectively. Random forest achieved an accuracy of 87%.

Keval pipalia and rahul bhadja conducted a study titled "Performance evaluation of different supervised learning algorithms for mobile price classification." [6] they utilized a dataset of mobile price available on kaggle to compare the performance of five supervised machine learning classifiers for mobile price classification. The study found that the gradient boost classifier achieved the highest accuracy of 90% based on the f1 score, while k-nearest neighbors (knn) had the lowest accuracy of 55%. The results suggest that gradient boosting and support vector machine (svm) algorithms performed well even with limited training data, and utilizing larger datasets could further improve their accuracy. The lower accuracy rates observed for certain algorithms may be attributed to the limited number of instances in the dataset. The study recommends future research to explore additional supervised machine learning algorithms to enhance the accuracy of the model.

In the paper titled "Mobile price prediction using machine learning techniques" By b. Balakumar, p. Raviraj, and v. Gowsalya,[7] the authors utilized linear regression and k-nearest neighbors (knn) algorithms for mobile price prediction. The accuracy achieved by the linear regression model was 91.32%, while the knn model achieved an accuracy of 92.12%. These results indicate that both algorithms were effective in predicting mobile prices with high accuracy.

1. **Methodology**

This Data set provides around 2000 examples of mobile phones with specific attributes and their prices. The number of attributes is 20. The output is with four ranges 0,1,2,3 from lower to higher prices. We will use Python with the sklearn library to train and test the model. We will use google colab as SAAS. The research was carried out in Google Colab’s Python kernel. The general workflow diagram of supervised ML tasks is as follows [5]:

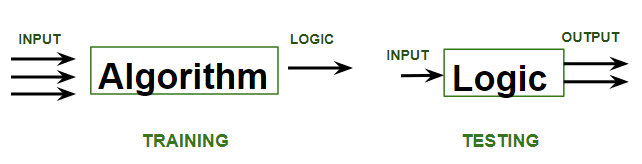


Figure 1.Supervised ML.

The training dataset is portioned into two – train for training the model and validation for its evaluation. The computer tries to comprehend the logic behind the pricing of a mobile based on its features and uses it to forecast future instances as correctly as possible.

**3.1 Attributes Definition**

In this dataset, there are multiple features that have a major effect on the price as follows:

Table 1.Attributes Definition.

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Datatype |
| battery\_power | battery capacity in milliampere-hours (mAh) | decimal |
| blue | is Bluetooth available? | boolean |
| clock\_speed | The CPU clock speed in Ghz | float |
| dual\_sim | is the dual sim? | boolean |
| fc | the front camera in megapixels | decimal |
| four\_g | is the fourth generation of networks supported? | boolean |
| int\_memory | internal memory size in GB | decimal |
| m\_dep | mobile depth in cm | float |
| mobile\_wt | mobile weight in gram | decimal |
| n\_cores | number of CPU cores | decimal |
| pc | primary camera in megapixels | decimal |
| px\_height | the vertical resolution of the phone's display screen. | decimal |
| px\_width | horizontal resolution of the phone's display screen. | decimal |
| ram | Temporary memory size in GB | decimal |
| sc\_h | height of the phone's display in cm | decimal |
| sc\_w | width of the phone's display in cm | decimal |
| talk\_time | battery life of the phone during continuous voice calls over hours | decimal |
| three\_g | is the third generation of networks supported? | boolean |
| touch\_screen | Does it have a touch screen? | boolean |
| wifi | Does it have wifi? | boolean |

**3.2 Data Preprocessing**

To create a good-performing model we need to depend on good data, the main idea of machine learning is to try to do the patterns that already exist in the dataset. But if the data have some missing fields or have data so far from other values (outliers), this will lead to bad training and bad prediction from the model. So we have some steps to do to make the data higher quality.

**3.2.1 Enough  Features and Records**

Features (columns) and records must be a good amount so the algorithm that trains the model detects the patterns and evaluate the target function accurately. We have 2000 record 0.08 of this for training and the remaining for validation.

**3.2.2 Handling Missing Values**

|  |
| --- |
| data.isnull().sum() |

There are no missing values in the dataset.

**3.2.3 Remove Invalid Values**

Now, we have some zero values of screen width and pixel resolution height in our data, but the mobile dimensions cannot be zero. set the mean of sc\_w instead invalid value.

|  |
| --- |
| data['sc\_w'][data[data.sc\_w == 0].index] = data.sc\_w.mean()  data['px\_height'][data[data.px\_height == 0].index] = data.px\_height.mean() |

**3.2.4 Data Splitting**

The data should be split into two parts, the first part is the training data, in which the model will be trained. We split this part into training data and validation data. So we decided to split it into 80% of the total dataset for training and 20% for testing. Then we have unseen data for testing the ML model.

|  |
| --- |
| x\_train, x\_valid, y\_train, y\_valid = train\_test\_split(x, y, test\_size = 0.2, random\_state = 0,stratify = y) |

**3.2.5 Standardization**

Feature scaling is a preprocessing step that normalizes and standardizes the feature values. Providing various benefits such as better convergence, improved model performance, and enhanced interpretability. It ensures fair comparisons between features and helps machine learning algorithms make more reliable and accurate predictions.

|  |
| --- |
| sc=StandardScaler() col=data.iloc[:,:20].columns x\_train=sc.fit\_transform(x\_train) x\_valid = sc.fit\_transform(x\_valid) |

The fit\_transform method (for sc instance) computes the mean and standard deviation of each feature in the training set and then scales the features based on these statistics. This make Improvement of algorithm convergence as SVM-based distance makes computing faster and reduces higher value effect, and improves the visualization.

**3.3 Training Algorithms**

There are three types of learning:

Supervised Learning: The model learns from labeled data to predict the correct output given new inputs.

Unsupervised Learning: The model discovers patterns and relationships within unlabeled data without predefined labels or target values.

Reinforcement Learning: The agent learns to take actions in an environment to maximize cumulative rewards based on feedback received through rewards or punishments.

In This labeled dataset, we will use Supervised Learning algorithms to train the model. The Algorithms we will use are  Random Forest, SVM, and ANN which all works for supervised learning.

**3.3.1 Random Forest**

Is an ensemble learning algorithm that combines multiple decision trees for classification tasks. In a Random Forest, each decision tree is constructed using the "divide and conquer” process and each  decision tree in the forest is a subset of the training data, randomly selecting data points with replacement from the original training set through a process called bootstrap sampling. Introduces randomness by using different subsets of training data and considering random subsets of features for splitting. This randomness reduces overfitting and enhances the diversity of the individual trees. By aggregating the predictions of the trees, Random Forest provides a final prediction.

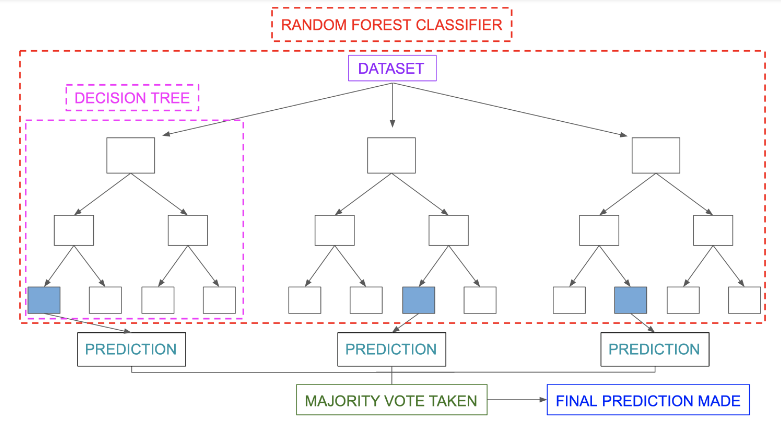


Figure 2.Random Forest Classifier Prediction.

|  |
| --- |
| rf = RandomForestClassifier(n\_estimators = 100, random\_state=0,criterion = 'entropy',oob\_score = True)  rf.fit(x\_train, y\_train)  y\_pred\_rf = rf.predict(x\_valid)  acc\_rf = accuracy\_score(y\_valid, y\_pred\_rf) |

**3.3.2 Support Vector Machine (SVM)**

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for binary Classification problems in Machine Learning. SVMs aim to find an optimal hyperplane that separates two classes, maximizing the margin between them.

To extend binary SVMs to handle multi-class classification, several approaches can be employed. One popular method is the One-vs-Rest (OvR) strategy. In this approach, we train a separate binary SVM for each class, treating it as the positive class while considering the remaining classes as the negative class. During prediction, the class associated with the SVM that yields the highest decision value is selected.

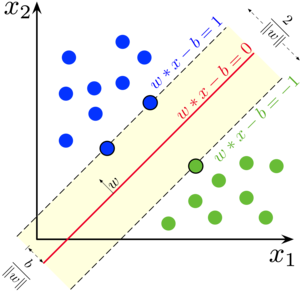


Figure 3.Support Vector Machine Classifier.

|  |
| --- |
| svm = SVC(kernel='linear',random\_state=0)  svm.fit(x\_train,y\_train)  y\_pred\_svm = svm.predict(x\_valid)  acc\_svm = accuracy\_score(y\_valid, y\_pred\_svm) |

**3.3.3 Artificial Neural Networks (ANN)**

The network is a multilayer perceptron neural network using the linear sigmoid activation function [2] as seen in Figure 4.

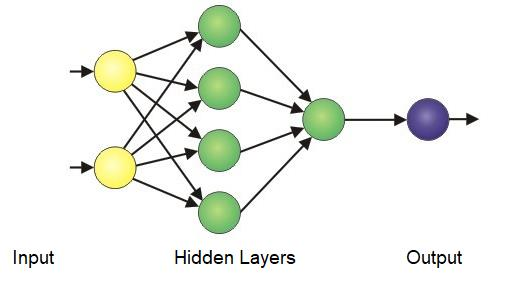


Figure 4.Artificial Neural System Architecture.

Here are the Back-propagation algorithm:

**∙** Initialize each wi to some small random value .

**∙** Until the termination condition is met, Do

**∙** For each training example <(x1,…xn),t > Do

**∙** Forward Propagation: OK = activation\_function (Σ(wi \* xi) + b)

**∙** For each output unit k: δk=ok(1-ok)(tk-ok)

**∙** For each hidden unit h: δh=oh(1-oh) Σ k wh,k δk

**∙** For each network weight wj Do wi,j=wi,j+Δwi,j,where Δwi,j= η δj xi,j and ηis the learning rate.

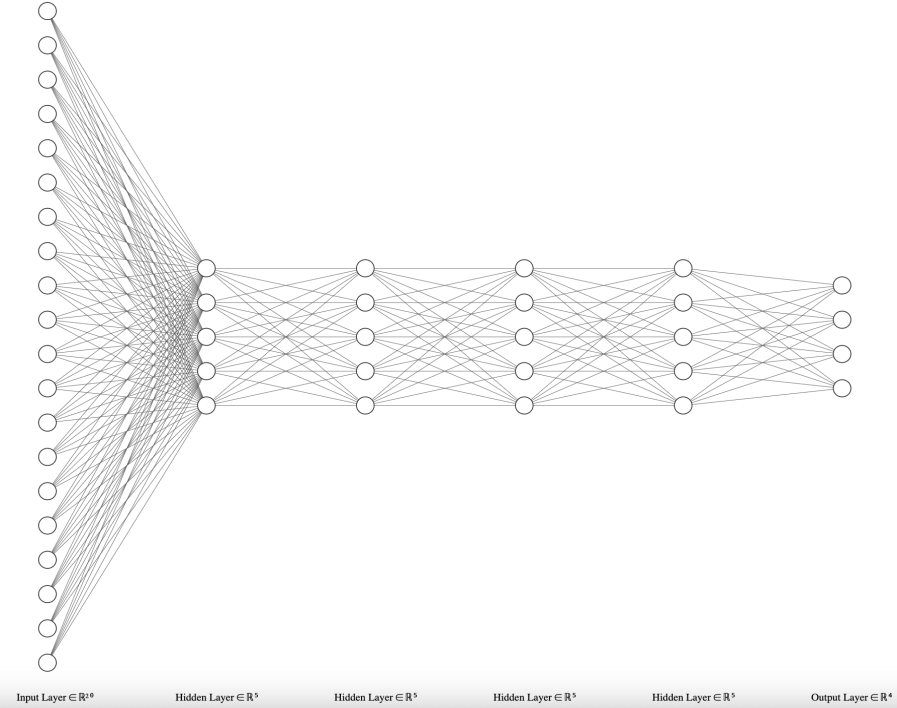


Figure 5.Design of the Neural Network.

|  |
| --- |
| ann=MLPClassifier(hidden\_layer\_sizes=(4,5),random\_state=0)  ann.fit(x\_train, y\_train)  y\_pred\_ann = ann.predict(x\_valid)  acc\_ann = accuracy\_score(y\_valid, y\_pred\_ann) |

**4. RESULTS AND DISCUSSION**

Accuracy score gives the accuracy of the trained model after evaluating it using validation data, for which we have sampled 20% of the dataset training.

|  |
| --- |
| models = ['RF', 'SVM', 'ANN']  acc\_scores = [acc\_rf, acc\_svm, acc\_ann] |

|  |  |
| --- | --- |
| Models | Accuracy |
| Random Forests | 0.9125 |
| Support Verctor Machine (SVM) | 0.9625 |
| Artificial Neural Network (ANN) | 0.97 |

The best algorithm (ANN):

A confusion matrix has the total count of the accurately grouped occurrences along its cross and the count of the Incorrectly classified instances in the rest of the matrix. We have used 4 class values so, the matrix generated is a 4\*4 matrix.

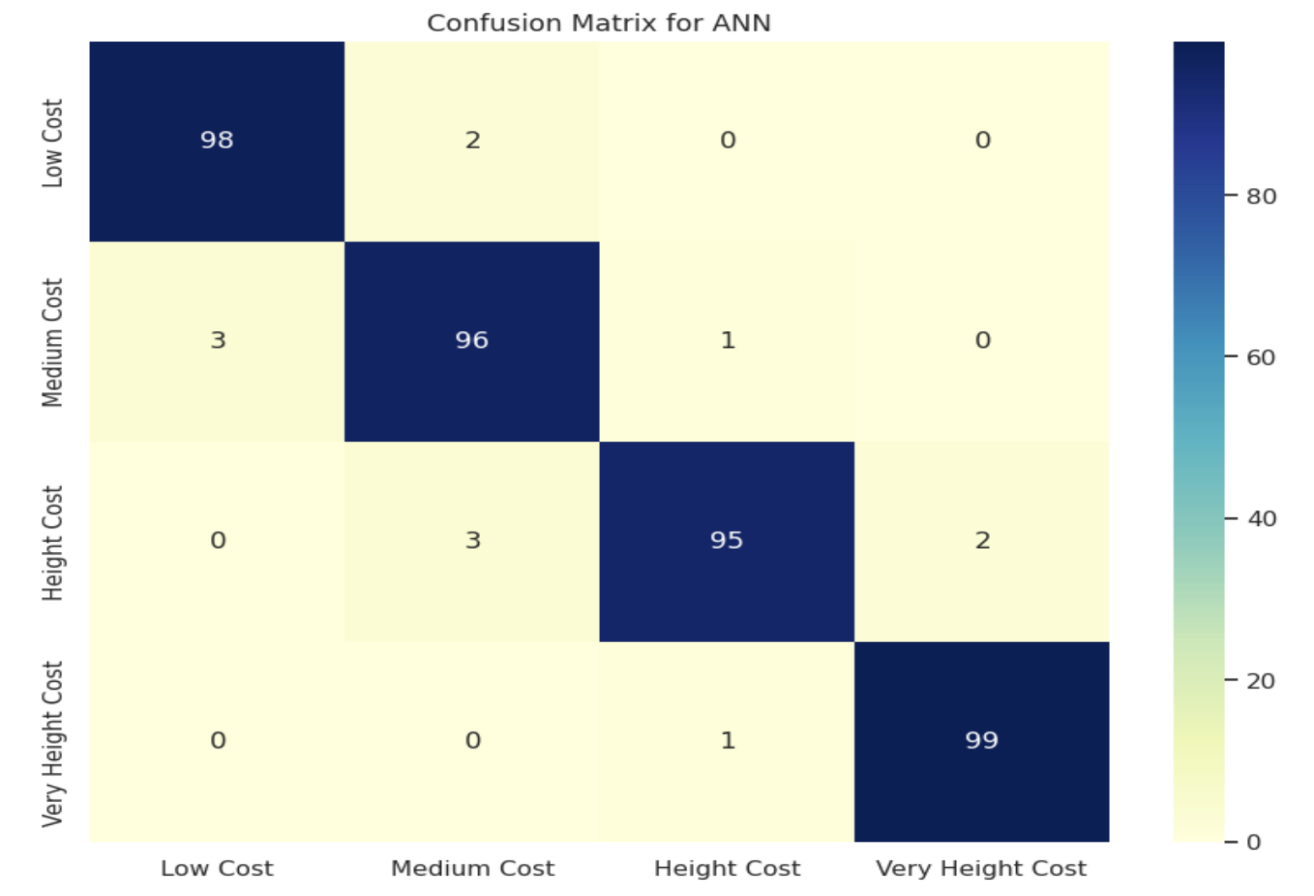
****

Figure 6.Confusion Matrix for ANN.

A classification report for the best algorithm we have (ann) gives the full report of the classification with parameters like accuracy, recall, precision, f1-score.

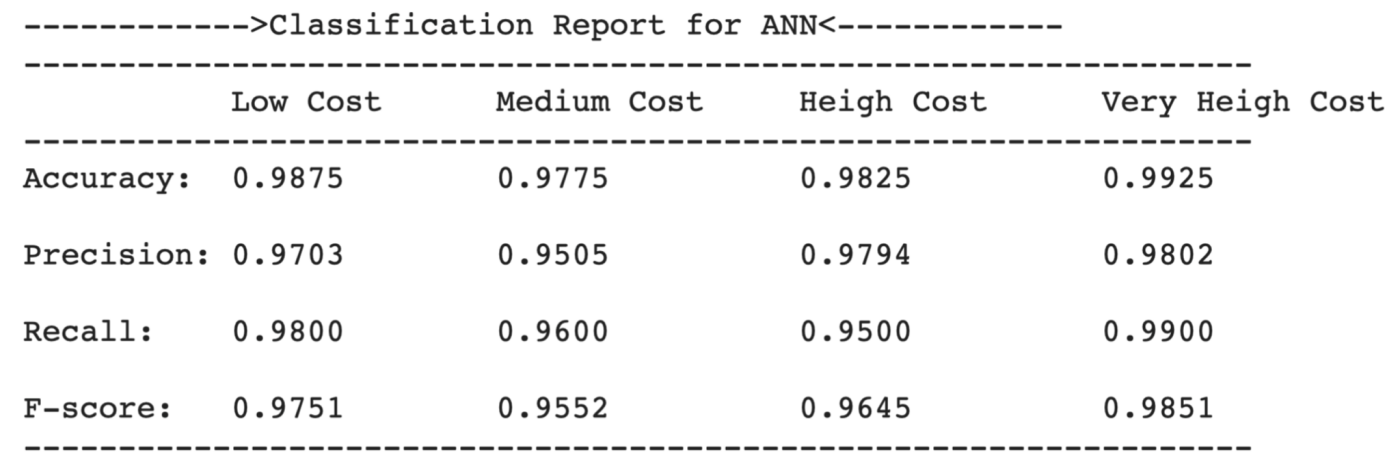
****

Figure 7.Classification Report.

In figure 8. We applye unseen data to ANN model.

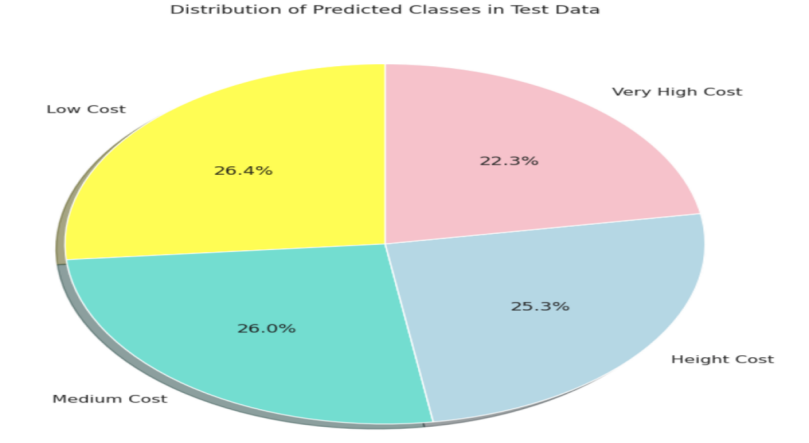
****

Figure 8.Distribution of predicted classes on test data.

**5. Conclusions**

Machine learning may be so helpful and easy to use in some scenarios, but it may be not essential in other cases. The dataset should be big enough so the algorithm can get the pattern of the data and learn the model properly. Otherwise, the model may become overfitted and then not produce good outputs.

When we made the valid 30% and the train 70% it produced excellent results but we changed it to 20%, 80% it produced more better results.

We compared three machine learning algorithms: Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Network (ANN). We used these algorithms to predict the price range of mobile phones based on various features.

After training and evaluating the models, we observed the following results:

- RF achieved an accuracy of 91.25% on the validation set. It also performed well but slightly lower than ANN.

- SVM achieved an accuracy of 96.25% on the validation set. It showed a good performance in predicting the price range of mobile phones.

- ANN achieved the highest accuracy of 97% on the validation set. It outperformed both SVM and RF in predicting the price range.

Based on these results, we can conclude that the Artificial Neural Network (ANN) model is the most accurate and reliable for predicting the price range of mobile phones in our dataset.

However, it's important to note that the performance of these models may vary depending on the dataset and the specific problem at hand. It's always recommended to experiment with different algorithms and evaluate their performance before drawing final conclusions.

Overall, this paper demonstrates the effectiveness of machine learning algorithms in predicting the price range of mobile phones, and the potential application of these models in the mobile industry for pricing and market analysis.

**References**

[1] Mobile Price Classification. (n.d.). Kaggle. Retrieved June 30, 2023, from <https://www.kaggle.com/iabhishekofficial/mobile-price-classification>

[2] Nasser, I. M., Al-Shawwa, M. O., & Abu-Naser, S. S. (2019). Developing Artificial Neural Network for Predicting Mobile Phone Price Range.‏

[3] Kukreti, A. The Application of Machine Learning Algorithms to a Dataset of Mobile Phone Prices for Classification.‏

[4] Ghaffar Nia, N., Kaplanoglu, E. & Nasab, A. Evaluation of artificial intelligence techniques in disease diagnosis and prediction. Discov Artif Intell 3, 5 (2023).

[5] Kiran, A. V., & Jebakumar, R. (2022). Prediction of Mobile Phone Price Class using Supervised Machine Learning Techniques. International Journal of Innovative Science and Research Technology, 7, 248-251.‏

[6] Pipalia, K., & Bhadja, R. (2020). Performance evaluation of different supervised learning algorithms for mobile price classification. In International Journal for Research in Applied Science & Engineering Technology (IJRASET) Dept. of Computer Engineering (Vol. 8, No. VI). Marwadi Education Found.

[7] Asim, M., & Khan, Z. (2018). Mobile price class prediction using machine learning techniques. *International Journal of Computer Applications*, *179*(29), 6-11.‏

[8] (n.d.). scikit-learn: machine learning in Python — scikit-learn 1.3.0 documentation. Retrieved June 30, 2023, from <https://scikit-learn.org/stable/>

[9] Welcome To Colaboratory - Colaboratory. (n.d.). Google Research. Retrieved June 30, 2023, from <https://colab.research.google.com/>

[10] muhammedessa/Machine-Learning-with-Python. (n.d.). GitHub. Retrieved June 30, 2023, from <https://github.com/muhammedessa/Machine-Learning-with-Python>

[11] Wiktionary. Retrieved June 30, 2023, from <https://www.youtube.com/watch?v=IlzGWhmX2-0&list=PLMYF6NkLrdN_UGFAUX2qKaMa_FZuJGoRA&ab_channel=MuhammedEssa>

[12] Mobile Price Classification: SVM, Random Forest, and Neural Network <https://github.com/M7mdNassar/Mobile-Price-Classification>

[13] Kalaivani, K. S., Priyadharshini, N., Nivedhashri, S., & Nandhini, R. (2021, November). Predicting the price range of mobile phones using machine learning techniques. In AIP Conference Proceedings (Vol. 2387, No. 1). AIP Publishing.‏

[14] Sakib, A. H., Shakir, A. K., Sutradhar, S., Saleh, M. A., Akram, W., & Biplop, K. B. M. B. (2022, January). A hybrid model for predicting Mobile Price Range using machine learning techniques. In 2022 The 8th International Conference on Computing and Data Engineering (pp. 86-91).‏

[15] Chattopadhyay, S., & Kishore, S. (2021). Classification of Mobile Price Range with Different Machine Learning Algorithms and Optimized Hyperparameters. American Journal of Electronics & Communication,

[16] Gupta, A. A., & Vijaykumar, S. (2020). Mobile price prediction by its features using predictive model of machine learning. Studies in Indian Place Names, 40(35), 906-913.‏

[17] Hu, N. (2022, July). Classification of Mobile Phone Price Dataset Using Machine Learning Algorithms. In 2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML) (pp. 438-443). IEEE.‏

[18] AYDIN, S. (2022). Using Machine Learning Algorithms in the Classification of Prices on MobilePhones. International Research in Science and Mathematics, 201.

[19] Our work in Google Colab : <https://colab.research.google.com/drive/18SC7ux9mVdEcdiQIr2h38BCCLhA1IcZJ?usp=sharing>