# FINAL PROJECT

***Topic*** :

Mobile price range classification using Machine Learning and Deep Learning techniques.

***Team members :***

Nitish Bharadwaj Putrevu

Bharath TVNS

Shreyas Jena

**Part 1 :** **Literature** **Review**

For a detailed literature review on the project topic, we consulted the following scholarly articles :

1) Mobile Price prediction using Machine Learning Techniques

Authors: B.Balakumar, P.Raviraj, V.Gowsalya

2) Mobile Price prediction using WEKA

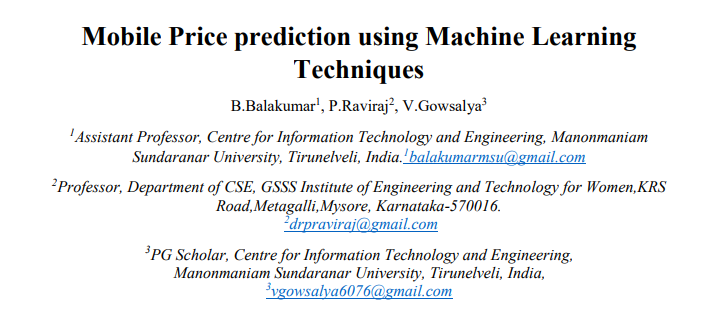
Authors: Pritish Arora, Sudhanshu Srivastava, Bindu Garg

3) Performance evaluation of different supervised learning algorithms for Mobile Price classification

Authors: Keval Pipalia, Rahul Bhadja

4) ANN for predicting Mobile Phone Price range

Authors: Ibrahim M. Nasser, Mohammed Al-Shawwa

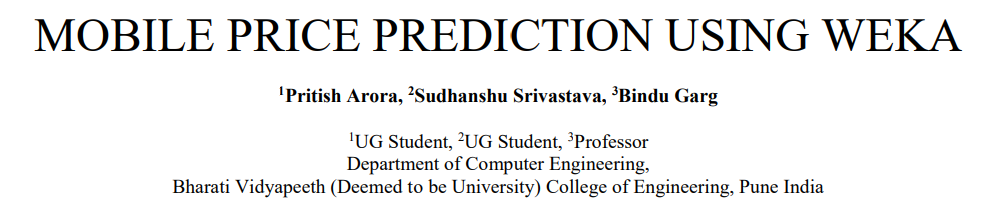
 **Link** : http://sajrest.com/Archives/vol4issue4\_2019/v4i4p3.pdf

1. Different feature selection algorithms are used to identify and remove less important and redundant features and have minimum computational complexity. Different classifiers are used to achieve as higher accuracy as possible. Results are compared in terms of highest accuracy achieved and minimum features selected. Conclusion is made on the basis of best feature selection algorithm and best classifier for the given dataset.
2. The methodology was to collect the data, preprocess it, apply an algorithm and obtain the accuracy of the result obtained.
3. Models used- Linear Regression, KNN Model .( Linear Regression was more accurate )
4. Important points -

* In forward selection, by adding irrelevant or redundant features to the data set decreases the efficiency of both classifiers.
* In backward selection if we remove any important feature from the

data set, its efficiency decreases.

* Converting a regression problem into classification problem introduces more error.
* The main reason of low accuracy rate is low number of instances in the data set.



**Link** : <https://www.ijsdr.org/papers/IJSDR2004057.pdf>

1) Specific feature selection algorithms are used to recognize and delete features that are less necessary and redundant, and have minimal complexity in computation.

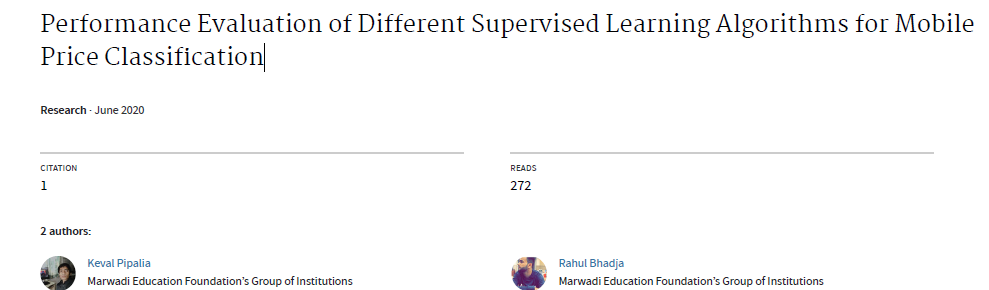
2) The methodology was to collect data, reduce it’s dimensions (7) ,feature selections, extract features that are variations of the original dimensions (Principal Component Analysis), forward selection, backward selection and classification.

3) Forward selection began with no variables and then they were added one by one, most decreasing the error, until any addition does not decrease the error (or only slightly decreases it).

4) Backward Selection began with all variables and remove them one by one, removing at each step the one that most decreases the error (or only slightly increases it), before any further removal significantly increases the error.

5) Accuracy was the metric for classification.

6) The models used were ZeroR algorithm, Naive Bayes algorithm, J48 decision tree algorithms, out of which J48 decision tree algorithms were the most effective.



1) Methodology was to collect and pre-process data, analyse it, select a model and implement it, evaluate the model and compare the result.

2) For pre-processing, the given data had been cleaned and null values had been removed.

Incomplete data such as data that which is lacking attribute values, missing values within the records were deleted from the data set. Outlier analysis was performed. In WEKA a filter called Interquartile-Range was used to perform outlier analysis.

3) A Correlation Matrix Heat Map and Feature description of columns was done to analyze the data.

4) Supervised learning algorithms such as Logistic Regression, K-Nearest Neighbours (KNN), Decision Tree, Support Vector Machine (SVM), and Gradient Boosting were used to classify the range of Mobile price.

5) Accuracy ,F1 score, Misclassification Rate , Precision, Recall , Specificity confusion matrix and Classification Report were used to evaluate the algorithms.

6) Even on the less training data, Gradient Boosting and SVM algorithms classifies very well and the accuracy can be increased by using big datasets. The main reason of low accuracy rate for some algorithms is low number of instances in the data set.

1) An Artificial Neural Network (ANN) model, was developed and tested for predicting the price range of a mobile phone.

2) A model based on the Multilayer Perceptron Topology was developed and trained.

3) A neural architecture was designed and a back-propagation algorithm was initialized.

4) The learning algorithm was able to determine the input variables importance.

5) The model used feed forward backpropagation algorithm for training. The factors for the model were obtained from data set represents mobile phones specifications.

6) The model was tested and the total result was 96.31% accuracy.

**How is our work different from the rest?**

1) The models we are going to use are Linear Regression, Random Forest Classification, XGBoost Classifier, Light GBM Classifier, Artificial Neural Networks, Convolutional Neural Networks, GRU, LSTM and a few pre- trained models like- VGG16 and Resnet50.

2) The evaluation metrics will be F1 score, log\_loss and accuracy and confusion matrix for the machine learning models and accuracy for the deep learning models.

3) Our analysis stands different from the previous work done, because we have used different metrics to compare Linear Regression, Random Forest Classification,XGBoost Classifier and Light GBM Classifier, Support Vector Machines ,Artificial Neural Networks, ,LSTM,and a MLP classifier. After this we compared whether deep neural architectures or standard machine learning models provide us maximum accuracy.

**Part 2: Observations**

*EDA :*

1) Initial evaluation of the “train” data set showed that it was a reasonably-sized data set with 2000 rows and 21 columns.

2) The data set had only numerical values; however some categorical variables had been encoded in the form of 0s and 1s.

3) There were no duplicated rows or null values in the data set.

4) An initial evaluation of the numerical data was done, first by using describe() function and then pictorially using histograms. While some variables like “int\_memory” and “mobile\_wt” showed uniform distributions, others like “clock\_speed” and “n\_cores” peaked at distinct values, indicating discrete distributions.

5) An initial evaluation of the categorical data was done by plotting frequency distribution using bar plots. Barring “three\_g”, all the other categorical variables showed uniform distribution among different cases.

6) Then an attempt was made to visualize all the data variables using box plots.

7) To get an idea of the dependencies between various variables, we used a heat map. It clearly showed that the dependent variable “price\_range” only shows significant dependency on a few key variables, namely -

“ram”, “battery\_power”, “px\_height” and “px\_width”.

This was further confirmed by using the background\_gradient() function.

8) We then did individual analysis of each of these key variables. First we checked out “battery\_power” using a scatter plot. Since the dependency value of price\_range w.r.t. battery\_power is only 0.20, a significant pattern was not observed.

9) But the scatter plot of price\_range w.r.t. “ram” showed a clear increasing pattern. ( because it shows high dependency with it, around 0.92 )

10) Then we tried to find out whether combining “ram” with other categorical parameters like “blue”, “three\_g” gave rise to new, interesting patterns. The result was negative, because the box plots when split based on these categorical variables, did not show any difference between each other.

11) We then tried a more direct approach to finding patterns between “price\_range” and some variables which we felt could be important based on our intuition. But the box plots did not reveal any new patterns.

12) In order to get a overall view of the dependencies, we also tried out the PairGrid(), but that did not reveal any new patterns.

13) We also tried a pair plot on a smaller subset of the train data set by including only selected columns.

14) We tried to represent categorical features by pie charts to make it look more appealing.

15) Finally, we tried out the distplot(), lineplot() and pointplot() to look for dependencies.

*Modelling:*

Model 1- ANN :

Hyperparameters used : epochs=300, verbose=1, loss='sparse\_categorical\_crossentropy', optimizer='adam', metrics=['accuracy'], input\_shape=(20)

Model 2- Pipeline model using MLPClassifier :

Hyperparameters used :

solver='adam', alpha=1e-5,hidden\_layer\_sizes=(70,70,70), random\_state=1

Model 3- LSTM :

Hyperparameters used :

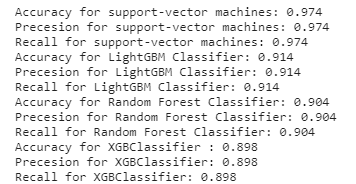
recurrent\_dropout=0.2,return\_sequences=True,loss='sparse\_categorical\_crossentropy',optimizer='adam',metrics=['accuracy'], epochs=5, batch\_size=32,verbose=1,

Model 4- XGBClassifier :

Hyperparameters used :

colsample\_bylevel=0.9,colsample\_bytree=0.8,gamma=0.99,max\_depth=5,min\_child\_weight=1,n\_estimators=10,nthread=4,random\_state=2,silent=True

**Evaluation**







**Conclusions**

1) The data set was easy to use because of its reasonable size and minimum pre processing requirements.

2) The data set, even though being reasonably-sized, was found to be lacking in showing many interesting relations between the independent variable “price\_range” and the other variables. Extensive EDA only managed to show us a few dependencies.

3)In terms of modelling, Support Vector Machines turned out to be the most accurate and the least time consuming model.

4)Deep Learning Models turned out to provide the least accurate results when the data is small, This is because deep learning algorithms need a large amount of data to understand it perfectly.

CODE LINK- https://drive.google.com/drive/folders/1xgThnOtIVosDU1b8-KF6vXM91ZuXv\_hE?usp=sharing

**Contributions**

**Nitish Bhardwaj Putrevu** : Contributed to the modelling and preprocessing of data. Also wrote the literature review.

**Bharath TVNS** : Contributed to the EDA and writing of final project report.

**Shreyas Jena** : Contributed to the EDA and writing of final project report.