**CLASSIFICATION OF MOBILE PHONE PRICE DATASET USING MACHINE LEARNING ALGORITHMS**

**ABSTRACT:**

In this project, we aim to perform a classification task on a Mobile Phone Price dataset obtained from Kaggle. The dataset contains various features of mobile phones, and our objective is to build a machine learning model that can classify mobile phones into four price categories: low cost medium cost, high cost, and very high cost. To achieve this, we will employ three different machine learning algorithms: Logistic Regression, Random Forest Classifier, and a Hybrid Classifier. Firstly, Logistic Regression will be used as a baseline model for the classification task. It is a simple yet effective algorithm for binary and multiclass classification. Next, the Random Forest Classifier, a powerful ensemble learning method, will be employed. It is known for its ability to handle complex relationships in data and reduce overfitting. Finally, we will develop a Hybrid Classifier, which combines the strengths of multiple models to improve classification accuracy. This hybrid approach may involve techniques like stacking, blending, or voting. By implementing these algorithms and comparing their performance, we aim to build a robust mobile phone price classification model that can assist consumers and businesses in making informed decisions about mobile phone purchases.

**Keywords:** Random Forest, Logistic Regression., Hybrid module.

**PROBLEM STATEMENT**

The problem at hand involves the accurate classification of mobile phones into four distinct price categories: low cost, medium cost, high cost, and very high cost. Leveraging machine learning algorithms, we aim to create a model that can analyze mobile phone features and provide real-time price range predictions. This classification system will aid consumers, manufacturers, and retailers in making informed decisions, optimizing market analysis, and enhancing the efficiency of product recommendation systems in the ever-evolving mobile technology industry.

**WHY THE PARTICULAR TOPIC CHOSEN?**

The choice of the mobile phone price classification project is driven by the increasing significance of mobile phones in our daily lives. As technology evolves rapidly, consumers and businesses face a multitude of options with varying price ranges. Developing a robust classification model using Logistic Regression, Random Forest, and a Hybrid approach enables us to address this real-world problem effectively. This project seeks to empower individuals and organizations by providing a reliable tool for informed decision-making when it comes to purchasing mobile phones, ultimately enhancing the mobile device selection process.

**SCOPE:**

The scope of this project encompasses the development and evaluation of machine learning models, including logistic regression, random forest classification, and potentially hybrid approaches, for accurately classifying mobile phones into four price categories. It involves data preprocessing, model training, and validation using the provided dataset. The project's outcome will offer a practical tool for categorizing mobile devices by their features, benefiting consumers, manufacturers, and retailers in making informed decisions, market analysis, and product recommendations within the mobile phone industry.

**OBJECTIVE OF THE PROJECT:**

The primary objective of this project is to develop a machine learning-based classification system that accurately categorizes mobile phones into four distinct price ranges (low cost, medium cost, high cost, very high cost) based on their features and specifications. By leveraging advanced algorithms, we aim to create a tool that assists consumers, manufacturers, and retailers in making informed decisions, streamlining the mobile phone purchase process, and enhancing market analysis in the rapidly evolving mobile technology industry.

**SYSTEM ANALYSIS**

**Existing System**

we aim to perform a classification task on a Mobile Phone Price dataset obtained from Kaggle. The dataset contains various features of mobile phones, and our objective is to build a machine learning model that can classify mobile phones into four price categories: low cost, medium cost, high cost, and very high cost. To begin, we will utilize Logistic Regression as a baseline model for the classification task. Logistic Regression is a simple yet effective algorithm for binary and multiclass classification and will serve as our initial benchmark.

**Disadvantages:**

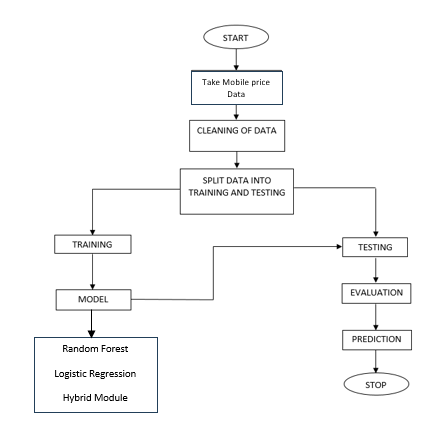
1. **Limited Complexity:** Logistic Regression assumes linear relationships, which may not capture complex patterns in mobile phone data.
2. **Sensitivity to Outliers:** Outliers can significantly impact model performance due to the likelihood-based loss function.
3. **Inability to Handle Non-linearity:** It struggles with non-linear relationships between features and target classes.
4. **Multiclass Extension Complexity:** Extending Logistic Regression to multiclass problems requires additional techniques like one-vs-all.
5. **Vulnerability to Overfitting:** Logistic Regression can overfit with high-dimensional feature spaces or insufficient data preprocessing.

**Proposed System**

we plan to tackle the mobile phone price classification task using three distinct machine learning algorithms: Logistic Regression, Random Forest Classifier, and a Hybrid Classifier. While Logistic Regression serves as our baseline model due to its simplicity and effectiveness, the Random Forest Classifier will leverage ensemble learning to handle complex data relationships and reduce overfitting. In addition to these two approaches, we will introduce a Hybrid Classifier, which will harness the combined strengths of both Logistic Regression and Random Forest. This hybrid module may incorporate advanced techniques such as stacking, blending, or voting, aiming to further enhance classification accuracy. By comparing the performance of these three approaches, we aim to construct a robust mobile phone price classification model, empowering consumers and businesses with valuable insights for informed purchasing decisions.

**Advantages:**

1. **Model Diversity:** Using three distinct models increases the likelihood of capturing different aspects of the data.
2. **Improved Decision-Making:** Consumers and businesses benefit from more accurate and reliable price classifications.
3. **Feature Interpretation:** The hybrid approach offers both interpretability and complexity handling.
4. **Adaptability:** Techniques like stacking and voting can adapt to changing data and model landscapes.
5. **Enhanced Performance:** By comparing models, you can select the best-performing one for the task at hand, optimizing decision support.

 **Block Diagram:**

**Fig 1. Block Diagram of Proposed System**

**HARDWARE & SOFTWARE REQUIREMENTS**

# **H/W CONFIGURATION:**

# Processor - I7/Intel Processor

* Hard Disk -160GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* RAM - 8Gb

**S/W CONFIGURATION:**

* Operating System : Windows 11
* Server side Script : Python, HTML, MYSQL, CSS, Bootstrap.
* Libraries : PANDAS, Django
* IDE : PyCharm (or) VS code
* Technology : Python 3.10

**Modules:**

1. **User**:
   1. **Register:**

Users can register for the Mobile Phone Price Prediction web application here.

* 1. **Login:**

After registering, the user can access his portal.

* 1. **View Data:**

View data after preprocessing (cleaned dataset)

* 1. **Input :**

User will give the input values.

* 1. **Result History:**

After giving the inputs, model will predict the result which it was set according to performance, it will predict that the Mobile Phone Price if low cost, high cost, very high cost are medium cost.

**Take Dataset:**

The dataset for the Mobile Phone Price data is collected from the kaggle website (kaggle.com).

The size of overall dataset is 124 KB (1,26,976 bytes)

**Pre-processing:**

* In preprocessing first of all we will check whether there is any Nan values.
* If any Nan values is present we will fill the Nan values with different fillna techniques like bfill, ffill, mode, and mean.
* Here we used the ffill (front fill) technique on our project.

**Training the data:**

Irrespective of the algorithm we select the training is the same for every algorithm**.**

Given a dataset we split the data into two parts training and testing, the reason behind doing this is to test our model/algorithm performance just like the exams for a student the testing is also exam for the model.

We can split data into anything we want but it is just good practice to split the data such that the training has more data than the testing data, we generally split the data.

And for training and testing there are two variables X and Y in each of them, the X is the features that we use to predict the Y target and same for the testing also.

Then we call the .fit ( ) method on any given algorithm which takes two parameters i.e., X and Y for calculating the math and after that when we call the .predict ( ) giving our testing X as parameter and checking it with the accuracy score giving the testing Y and predicted X as the two parameters will get our accuracy score and same steps , these are just checking for how good our model performed on a given dataset.