Mobile Price Prediction Application

Gaurvi Bhardwaj

1. Abstract:

The goal of any organization is to make their product to get succeed and compete with other products in the market where pricing of their products plays a vital role. To sell any product in market, the most important aspect is to determine the price. There are many traditional and new methods for estimating before pricing their products, and a method is chosen which gives more appropriate result.

In today's fiercely competitive mobile phone market, pricing strategies have become increasingly intricate, influenced by a multitude of factors. From RAM and internal memory to camera quality, battery power, and screen size, numerous variables now dictate the cost of a mobile device. To unravel the complexities of pricing dynamics, a dataset has been provided, delineating each aspect of mobile phones as distinct variables. This dataset serves as the foundation for a comprehensive study aimed at elucidating the factors that significantly impact the price range of mobile phones.

AI/ML-powered mobile price range prediction offers businesses a powerful tool for understanding market dynamics, optimizing pricing strategies, and driving revenue growth. By leveraging these technologies effectively, businesses can gain a competitive advantage and deliver greater value to customers.

2. Problem statement :

Price is an important factor in marketing and shopping. Customers like to check price of any any commodity if it is expensive or cheap rather than finding out its features, quality ,etc. For mobile companies, it is important to judge the humans mind and range where the sell is high. Moreover, for stability they have to fight in the competitive mobile phone market, companies aim to decipher the sales trends and discern the underlying factors influencing pricing dynamics. The goal is to unveil correlations between various mobile phone features (e.g., RAM, Internal Memory) and their corresponding price. This approach seeks to provide insights into the relative pricing levels, indicating the degree of affordability or premiumness associated with different sets of features.

3. Market / Customer/ Business Need Assessment:

Customer Need Assessment: Mobile phones are the best selling electronic devices as people keep updating their cell phones whenever they find new features in a new device. Thousands of mobiles are sold daily, in such a situation it is a very difficult task for someone who is planning to set up their own mobile phone business to decide what the price of the mobile should be. For consumers, purchasing a mobile phone is a significant investment, and they want to ensure they are getting the best value for their money. Understanding the factors influencing the price range of mobile phones is crucial for making informed choices that align with their budget and preferences. By providing a predictive model for mobile price ranges, businesses can cater to the customer's need for transparency and enable them to compare and evaluate different devices effectively.

Market Need Assessment: In today's fast-paced mobile phone market, consumers are inundated with a plethora of choices, each offering a unique combination of features and functionalities. However, amidst this abundance, there exists a pressing need for clarity and transparency in pricing. Consumers seek guidance in understanding the rationale behind mobile phone prices and desire a reliable method to anticipate the price range of their desired device. By accurately predicting mobile price ranges, businesses can address this market need and empower consumers to make informed purchasing decisions.

Business Need Assessment: For mobile phone manufacturers and retailers, pricing strategy is paramount in maintaining competitiveness and maximizing profitability. However, determining the optimal price point requires a nuanced understanding of market trends, consumer preferences, and product attributes. By developing a robust predictive model for mobile price ranges, businesses can gain insights into the factors driving pricing decisions and optimize their strategies accordingly. This enables them to align their product offerings with market demand, effectively manage inventory, and enhance overall business performance. Moreover, a comprehensive understanding of price dynamics empowers businesses to respond swiftly to market fluctuations and gain a competitive edge in the dynamic mobile phone industry.

4. Target Specifications and Characterization:

- To analyze a dataset containing various features of mobile phones.
- Build a predictive model that can accurately classify the price range of a given phone as low, medium, high, or very high
- Adherence to strict data privacy and security standards to protect sensitive information.
- High accuracy in predicting mobile phone price ranges and ability to handle large datasets and accommodate growing data volumes effectively.

5. External Searches (Information searches):

5.1 Applications of Machine Learning in Mobile Price Prediction

Machine learning algorithms analyze historical sales data, market trends, and consumer behavior to predict future demand for specific smartphone models. By identifying patterns and correlations in large datasets, these algorithms can anticipate fluctuations in demand and adjust pricing strategies accordingly. It also enables the analysis of competitor pricing strategies, product features, and market positioning. By gathering and analyzing data from competitors' websites, social media platforms, and other sources, machine learning algorithms can provide insights into competitive pricing dynamics, helping companies formulate effective pricing strategies to maintain competitiveness. Machine learning algorithms optimize pricing strategies by considering various factors such as production costs, demand elasticity, and market conditions. Through iterative learning and optimization techniques, these algorithms determine the optimal price points that maximize revenue while satisfying consumer demand and maintaining profitability. ML enables personalized pricing strategies tailored to individual consumer preferences, purchasing history, and demographic characteristics. By analyzing customer data and behavior patterns, companies can offer personalized pricing incentives, discounts, or promotions to enhance customer loyalty and drive sales.

5.2 Mobile dataset :

Dataset Information

In [51]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   battery_power  2000 non-null   int64  
 1   blue           2000 non-null   int64  
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64  
 4   fc             2000 non-null   int64  
 5   four_g         2000 non-null   int64  
 6   int_memory     2000 non-null   int64  
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64  
 9   n_cores        2000 non-null   int64  
 10  pc             2000 non-null   int64  
 11  px_height      2000 non-null   int64  
 12  px_width       2000 non-null   int64  
 13  ram            2000 non-null   int64  
 14  sc_h           2000 non-null   int64  
 15  sc_w           2000 non-null   int64  
 16  talk_time      2000 non-null   int64  
 17  three_g        2000 non-null   int64  
 18  touch_screen   2000 non-null   int64  
 19  wifi           2000 non-null   int64  
 20  price_range    2000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

Figure: Depicts Dataset information

6 Business Model:

In the dynamic landscape of smartphone market growth, the integration of Artificial Intelligence (AI) holds paramount importance, particularly in the realm of price prediction. AI algorithms play a pivotal role in analyzing vast amounts of data related to consumer behavior, market trends, and technological advancements. By leveraging AI-powered predictive models, smartphone manufacturers and retailers can forecast pricing strategies with greater accuracy and efficiency.

These AI-driven systems enable companies to assess various factors influencing smartphone prices, such as supply chain dynamics, competitor pricing strategies, and consumer preferences. Through advanced machine learning algorithms, AI can detect subtle patterns and correlations within datasets, offering insights into pricing trends and optimal pricing points.

Moreover, AI enhances the ability to adapt to rapidly changing market conditions and consumer demands. By continuously learning from real-time data inputs, AI algorithms can refine price predictions and adjust strategies accordingly, ensuring competitiveness and profitability in a dynamic market environment.

In essence, AI empowers stakeholders in the smartphone industry to make informed decisions regarding pricing, optimizing revenue streams, and enhancing overall market competitiveness. Its role extends beyond mere prediction, driving strategic initiatives and fostering innovation to meet the evolving needs of consumers in an increasingly digitalized world.

7 Concept Generation:

The project follows a structured workflow, encompassing data wrangling, exploratory data analysis (EDA), hypothesis testing, feature engineering, and machine learning model implementation. Here's a brief overview of each step:

1.Data Wrangling: Handle missing values and unique value checks. Replace erroneous 0 values for pixel resolution height and screen width with the mean values to ensure data consistency.

2.Exploratory Data Analysis (EDA): Gain insights into the relationships between variables and the price range. Discover patterns, correlations, and distributions in the data. Identify significant factors affecting the price range, such as battery capacity, RAM, and pixel quality.

```
from sklearn.model_selection import train_test_split

X = df.drop(['price_range'], axis = 1)
y = df['price_range']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Training set shape:", X_train.shape, y_train.shape)
print("Testing set shape:", X_test.shape, y_test.shape)
```

```
Training set shape: (1600, 20) (1600,)
Testing set shape: (400, 20) (400,)
```

Figure: Data Splitting

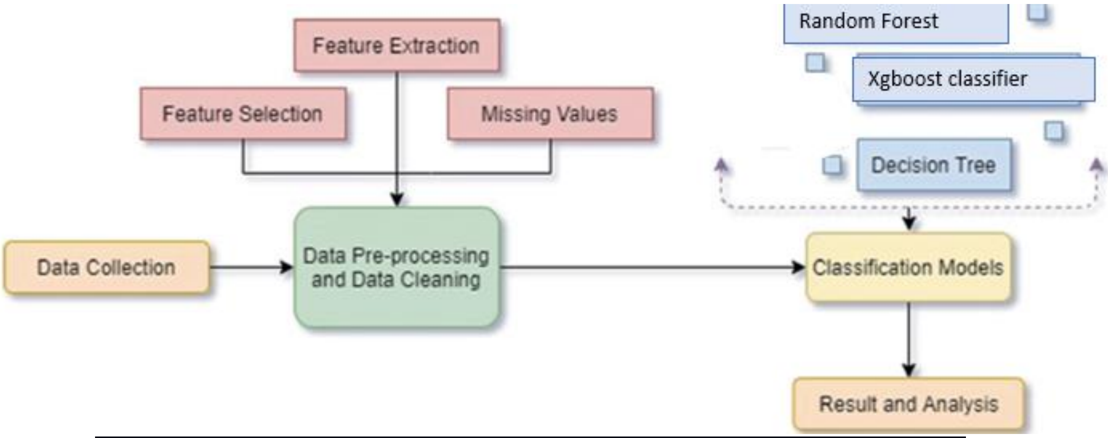
3.Hypothesis Testing: Perform statistical tests to validate hypotheses and identify significant factors. Handle outliers that may affect the model's performance.

4.Feature Engineering: Engineer new features or transform existing ones to enhance the model's predictive power. Select relevant features that contribute most to the price range prediction.

5.Machine Learning Models: Implement machine learning algorithms to build the predictive model. Evaluate various models, including logistic regression, random forest, and XGBoost, to determine the best-performing algorithm. Tune hyperparameters for improved model performance.

11 Final Product prototype:

This machine learning project provides insights into the factors influencing the price range of mobile phones. By leveraging the dataset's features, I built a predictive model capable of accurately classifying mobile phone price ranges. The project highlights the significance of variables such as RAM, battery power, and pixel quality in determining the price range. This project can be deployed on the website or app.



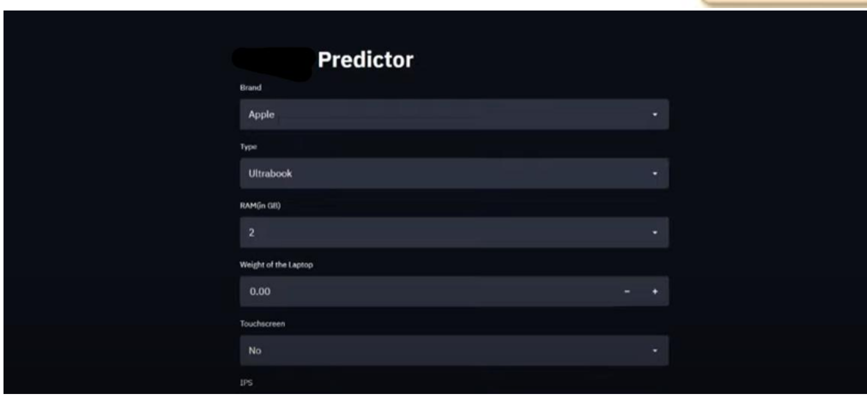


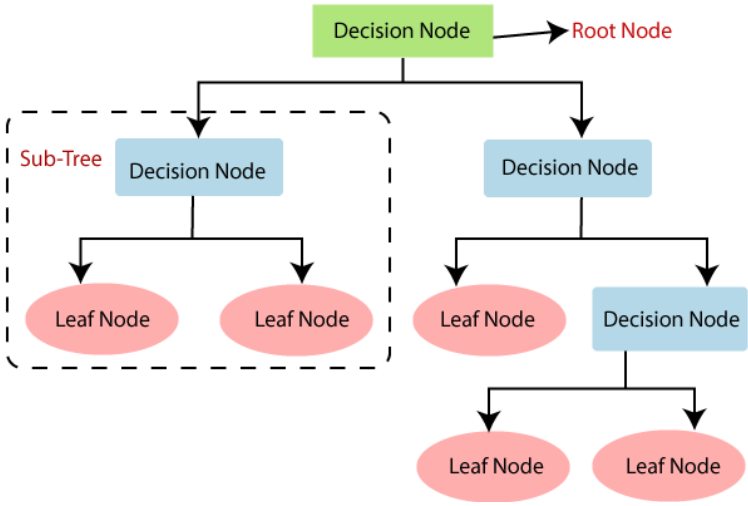
Figure: Prototype of deploying

12 Product details:

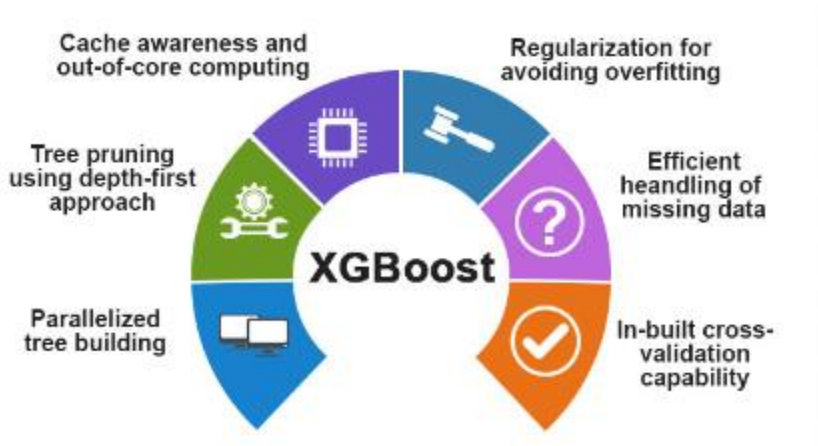
12.1 Algorithm:

Classification model contains different algorithms which can be used to classify malignant or benign tumors based on the predefined cancer dataset.

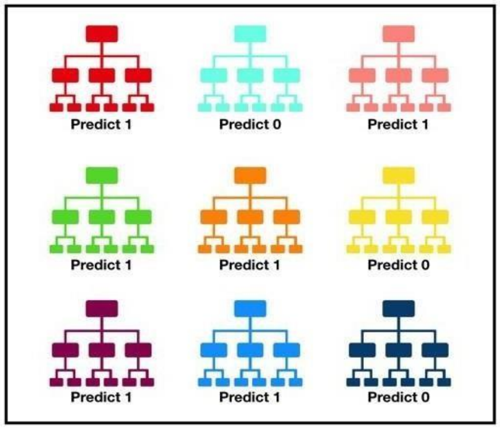
DecisionTreeClassifier: The DecisionTreeClassifier is a tree-based classification algorithm that recursively splits the dataset based on feature conditions to create a tree-like structure. It makes decisions by traversing the tree from the root to a leaf node, where each leaf represents a class label.Decision trees inherently provide a measure of feature importance based on how often a feature is used to split the data. Features used near the top of the tree contribute more to the model's decision-making process.



XGBoost : XGBoost is an ensemble learning algorithm that combines the predictions of multiple weak learners (typically decision trees) to create a more robust and accurate model. It is known for its high predictive performance and efficiency.



Random Forest: Random Forest is an ensemble learning method that builds a multitude of decision trees during training and outputs the mode of the classes for classification or the average prediction for regression. It introduces randomness by training each tree on a random subset of the data and using a random subset of features for each split.Random Forest provides a more robust measure of feature importance compared to a single decision tree. The feature importance is computed as the average of the feature importance across all trees in the forest.



Tally: Six 1s and Three 0s
Prediction: 1

12.2 Python-libraries:

- Pandas: Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant.

- Scikit Learn: Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library used based on python programming language in spyder IDE. It features various classification, regression and clustering algorithms including (SVM) support vector machines , random forests, gradient boosting, k-means and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

- Seaborn: Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures. This is a visualization tool used to demonstrate the count of benign and malignant cells through the predefined dataset.

- Xgboost: XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. It’s vital to an understanding of XGBoost to first grasp the machine learning concepts and algorithms that XGBoost builds upon: supervised machine learning, decision trees, ensemble learning, and gradient boosting.

Algorithm with highest accuracy among classification algorithms is chosen as the best algorithm for Mobile price prediction.

12.3 Team required to develop:

1. Machine learning engineering
2. Business analyst
3. Software developer
4. Cloud engineer
5. Data Researcher

13. Code Implementation:

EDA:

```
In [66]: # how many instances are in each price range category (0, 1, 2, 3)
price_range_counts = df['price_range'].value_counts()

print("Number of instances in each price range category:")
for category, count in price_range_counts.items():
    if category == 0:
        price_category = "Low Cost"
    elif category == 1:
        price_category = "Medium Cost"
    elif category == 2:
        price_category = "High Cost"
    elif category == 3:
        price_category = "Very High Cost"
    else:
        price_category = "Unknown"

    print(f"{price_category}: {count}")
```

```
Number of instances in each price range category:
Medium Cost: 500
High Cost: 500
Very High Cost: 500
Low Cost: 500
```

ML Modelling:

ML Model - 1 Random Forest classifier

In [90]:
```
# ML Model - 1 Implementation

# Import necessary libraries
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score


# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=300, random_state=42)

# Train the classifier on the training set
rf_classifier.fit(X_train, y_train)

# Predict the labels for the training and testing sets
train_predictions = rf_classifier.predict(X_train)
test_predictions = rf_classifier.predict(X_test)

# Calculate accuracy for the training and testing sets
train_accuracy = accuracy_score(y_train, train_predictions)
test_accuracy = accuracy_score(y_test, test_predictions)

# Print the results
print("Training Accuracy: {:.2f}%".format(train_accuracy * 100))
print("Testing Accuracy: {:.2f}%".format(test_accuracy * 100))
```

```
Training Accuracy: 100.00%
Testing Accuracy: 88.75%
```

In [91]:
```
# classification report for Test Set
from sklearn.metrics import classification_report

# Use the trained Random Forest classifier to make predictions on the test set
y_pred = rf_classifier.predict(X_test)

# Generate the classification report
report = classification_report(y_test, y_pred)

# Print the classification report
print("Classification Report for Test Set:\n", report)
```

```
Classification Report for Test Set:
               precision    recall  f1-score   support

           0       0.94      0.96      0.95       105
           1       0.88      0.85      0.86        91
           2       0.79      0.86      0.82        92
           3       0.94      0.88      0.91       112

    accuracy                           0.89       400
   macro avg       0.89      0.89      0.88       400
weighted avg       0.89      0.89      0.89       400
```

ML Model - 2 DecisionTreeClassifier

In [94]:

```
# ML Model - 2 Implementation

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a Decision Tree Classifier
clf = DecisionTreeClassifier(random_state=42)

# Train the model
clf.fit(X_train, y_train)

# Make predictions on the test set
y_pred = clf.predict(X_test)
```

In [95]:

```

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

# Print the evaluation metrics
print(f'Accuracy: {accuracy}')
print('Classification Report:\n', classification_rep)
print('Confusion Matrix:\n', conf_matrix)
```

```
Accuracy: 0.8375
Classification Report:
               precision    recall  f1-score   support

           0       0.91      0.88      0.89       105
           1       0.75      0.85      0.80        91
           2       0.80      0.71      0.75        92
           3       0.87      0.90      0.89       112

    accuracy                           0.84       400
   macro avg       0.83      0.83      0.83       400
weighted avg       0.84      0.84      0.84       400

Confusion Matrix:
 [[ 92  13   0   0]
 [  9  77   5   0]
 [  0  12  65  15]
 [  0   0  11 101]]
```

ML Model - 3 XGBClassifier

In [99]:
```
# ML Model - 3 Implementation
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from xgboost import XGBClassifier

# Assuming X and y are your features and labels
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create an XGBoost classifier
xgb = XGBClassifier(max_depth = 5, learning_rate = 0.1)

# Fit the Algorithm
xgb.fit(X_train, y_train)


# Predict on the model
y_pred = xgb.predict(X_test)
```

In [100]:
```
# Evaluate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.9075
```

In [101]:
```
# Evaluation metrics for Test set
score = classification_report(y_test, y_pred)
print('Classification Report for XGBoost(Test set)= ')
print(score)
```

```
Classification Report for XGBoost(Test set)=
              precision    recall  f1-score   support

           0       0.94      0.94      0.94       105
           1       0.85      0.92      0.88        91
           2       0.89      0.84      0.86        92
           3       0.94      0.92      0.93       112

    accuracy                           0.91       400
   macro avg       0.91      0.91      0.90       400
weighted avg       0.91      0.91      0.91       400
```

Code Link-
https://github.com/Gaurvi-bhardwaj/Mobile-price-prediction/blob/main/Mobile_price_prediction.ipynb

13 Conclusion:

One of the key advantages of machine learning in mobile price prediction is its ability to process large amounts of data quickly and efficiently, delivering insights in seconds rather than the hours or days it may take for manual analysis. This rapid turnaround time enables businesses to make timely and informed decisions, optimizing pricing strategies and maximizing profitability.

Moreover, machine learning offers the potential for personalized pricing strategies tailored to individual consumer preferences and market dynamics. By analyzing historical sales data, competitor pricing, and other relevant factors, ML models can recommend optimal pricing points that resonate with target customers and drive sales.

While there are still challenges to overcome, such as data limitations and biases in models, the trajectory is clear: machine learning is the future of mobile price prediction. As technology continues to evolve and datasets expand, AI-powered systems will play an increasingly prominent role in shaping pricing strategies and market dynamics in the smartphone industry.

13 References:

[1] M. Çetın and Y. Koç, "Mobile Phone Price Class Prediction Using Different Classification Algorithms with Feature Selection and Parameter Optimization," 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2021, pp. 483-487, doi: 10.1109/ISMSIT52890.2021.9604550.

[2] P. Arora, S. Srivastava and B. Garg, mobile price prediction using weka, 2020.

[3] Gandomi, A., & Haider, M. (2015). "Beyond the hype: Big data concepts, methods, and analytics." International Journal of Information Management, 35(2), 137-144.

[4] Wu, H., & Lu, H. (2019). "Mobile App User Behavior Prediction: A Business Perspective." IEEE Transactions on Knowledge and Data Engineering, 31(7), 1307-1320.

[5] https://www.kaggle.com/datasets/mohannapd/mobile-price-prediction