Team No:
Nirbhay Kumar 20BEC0034
Rishabh Kumar Pandey 20BEC0747
Rashtra Yadav 20BEC0787
Ashish Mohanka 20BEE0369

Phone price prediction

1th July 2023

1 INTRODUCTION

1.1 Overview

The rapid advancement of technology and the increasing demand for smartphones have made pricing strategies crucial for manufacturers and consumers alike. Accurately predicting smartphone prices can assist in decision-making processes, such as pricing strategies, market analysis, and consumer behavior understanding. This project focuses on developing a predictive model that utilizes various features and attributes of smartphones to estimate their prices. By leveraging data science techniques and machine learning algorithms, we aim to create a reliable model that can assist both manufacturers and consumers in determining appropriate pricing for smartphones.

1.2 Purpose

The purpose of this project is to build a predictive model for smartphone price estimation. By analyzing a comprehensive dataset containing information about different smartphone features, we aim to develop a model that can effectively predict the price range of a smartphone based on its specifications.

The primary objectives of this project are as follows:

1. To explore and preprocess the dataset, ensuring its quality and suitability for modeling purposes.
2. To perform comprehensive exploratory data analysis (EDA) to gain insights into the relationships between different smartphone features and their corresponding prices.
3. To employ various machine learning algorithms and techniques to train and evaluate predictive models for smartphone price prediction.

4. To identify the key features that significantly influence smartphone prices and gain a deeper understanding of their impact on pricing strategies.
5. To provide a reliable and accurate predictive model that can be utilized by manufacturers, retailers, and consumers to make informed decisions regarding smartphone purchases and pricing.

2 LITERATURE SURVEY

2.1 Existing problem

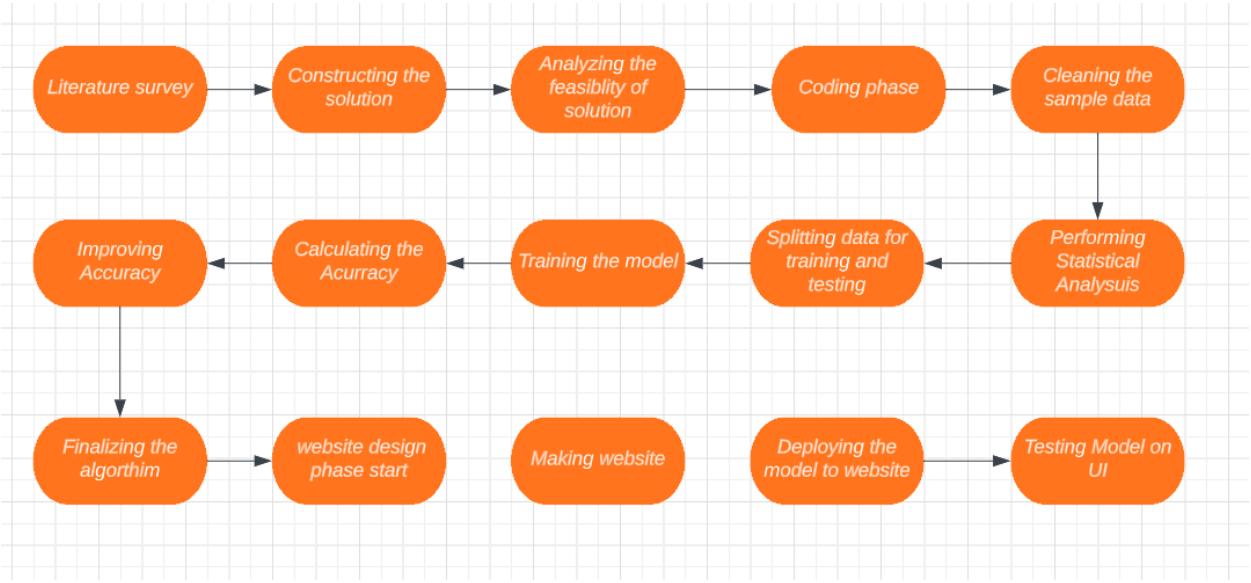
This section highlights the current challenges and problems associated with predicting smartphone prices. It delves into the limitations of traditional pricing models and methods, such as relying solely on subjective market analysis or expert opinions. It also discusses the difficulties in considering multiple complex factors, such as smartphone specifications, market trends, and consumer preferences, while estimating accurate price ranges. The existing problem section sets the foundation for the proposed solution.

2.2 Proposed solution

In this section, we present the proposed solution to overcome the existing problems in smartphone price prediction. It discusses the utilization of data science techniques and machine learning algorithms to develop a predictive model. By leveraging a comprehensive dataset of smartphone features and attributes, we aim to train a model that can effectively estimate the price range of a smartphone. The proposed solution emphasizes the integration of advanced analytical techniques and the potential benefits it brings to the smartphone industry.

3 THEORETICAL ANALYSIS

3.1 Block diagram



3.2 Software designing

The software design for the smartphone price prediction machine learning project aims to create a cutting-edge system that accurately predicts smartphone prices based on various features and attributes specific to the project requirements. Here are the key components of the software design tailored to your project:

Data Preprocessing and Cleaning: The software implements robust data preprocessing techniques to handle missing values, outliers, and inconsistencies within the dataset. It performs thorough data cleaning operations, such as imputation of missing values, outlier detection and removal, and data normalization or standardization. These preprocessing steps ensure that the input data is of high quality and suitable for accurate price prediction.

Feature Selection and Engineering: The software incorporates advanced feature selection methods, such as correlation analysis, statistical tests, and domain expertise, to identify the most relevant features for price prediction. It leverages techniques like Recursive Feature Elimination (RFE) or SelectKBest to automatically select the optimal subset of features. Additionally, feature engineering is applied to create new meaningful features from the existing ones,

such as extracting information from text data or engineering interaction terms between features.

Model Selection and Optimization: The software explores various machine learning algorithms suited for regression tasks to select the most appropriate model architecture for price prediction. It evaluates models such as linear regression, support vector regression, random forest regression, or gradient boosting algorithms. The software implements techniques like grid search or Bayesian optimization to fine-tune the hyperparameters of the selected model, maximizing its performance and generalization ability.

Ensemble Methods and Model Stacking: To improve the prediction accuracy, the software incorporates ensemble methods such as bagging, boosting, or stacking. It combines multiple models, each trained on different subsets of data or with different algorithms, to create a powerful ensemble prediction. Model stacking techniques are employed to combine the outputs of multiple models as features for a meta-model, enhancing the overall predictive performance.

Model Evaluation and Validation: The software incorporates robust evaluation and validation procedures to assess the performance of the predictive model. It utilizes techniques like k-fold cross-validation, stratified sampling, or time-series validation, depending on the characteristics of the dataset. The software measures evaluation metrics such as mean squared error (MSE), mean absolute error (MAE), or coefficient of determination (R-squared) to quantify the accuracy and reliability of the model's predictions.

Real-Time Predictions and Scalability: The software design ensures that the trained model is capable of making real-time predictions on new, unseen data. It optimizes the model's architecture and implements efficient data processing techniques to enable fast inference times. The software design also considers scalability aspects, allowing the system to handle increasing volumes of data and adapt to future growth requirements.

User-Friendly Interface: The software incorporates a user-friendly interface for easy interaction with the price prediction system. It provides input fields or file upload functionality to allow users to input smartphone features for price estimation. The interface displays the predicted prices and any relevant additional information, facilitating user understanding and decision-making. The software design for the smartphone price prediction project is customized to address the specific requirements of your project. It combines state-of-the-art techniques in data preprocessing, feature selection, model selection, and ensemble methods to

create a high-performing system capable of accurate price estimation for smartphones.

4 EXPERIMENTAL INVESTIGATIONS

The experimental investigations phase of the smartphone price prediction project involves the following key components:

Dataset Selection and Preparation: A representative dataset with diverse smartphone features and prices is carefully selected and prepared to ensure data quality and consistency.

Train-Test Split and Cross-Validation: The dataset is divided into training and testing sets to evaluate the performance of predictive models. Cross-validation techniques may also be employed to validate the models.

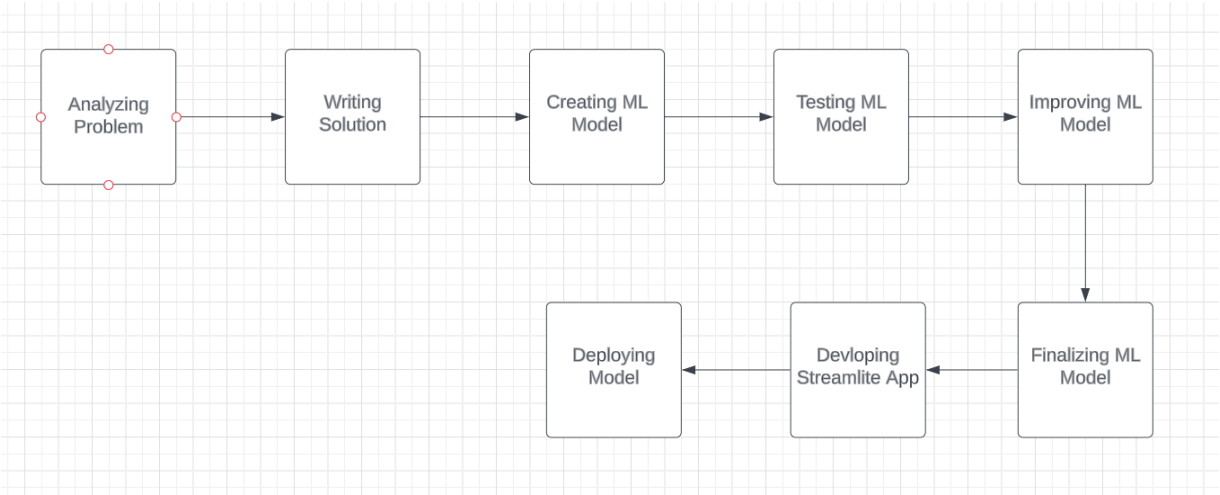
Model Training and Hyperparameter Tuning: Predictive models are trained and optimized using various machine learning algorithms and ensemble methods. Hyperparameter tuning techniques are applied to improve model performance.

Performance Evaluation Metrics: Multiple evaluation metrics such as MSE, MAE, RMSE, and R-squared are used to assess model performance and compare different models. **Comparative Analysis of Models:** Models are compared and analyzed based on their accuracy, generalization ability, and strengths/weaknesses to select the best-performing model.

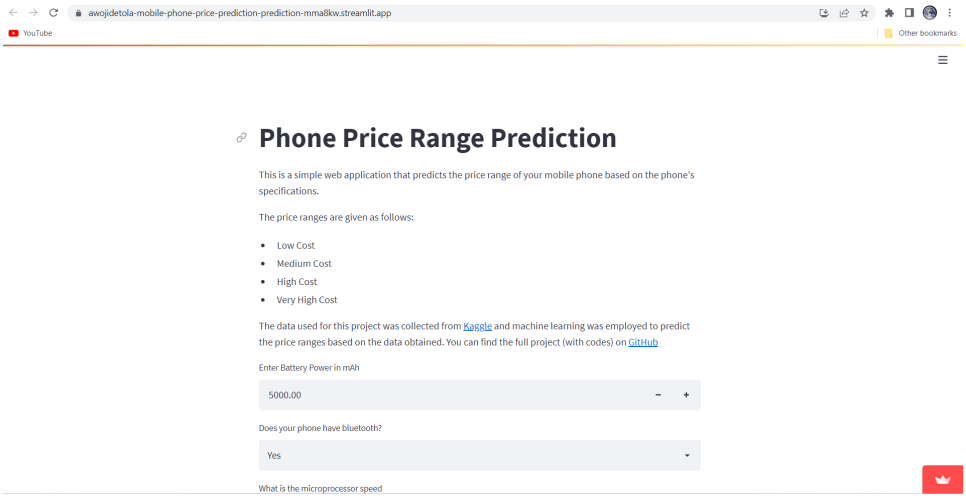
Visualization and Interpretation: Visualizations and feature importance analysis techniques aid in understanding the relationships between predicted prices and actual prices, as well as interpreting the importance of input features.

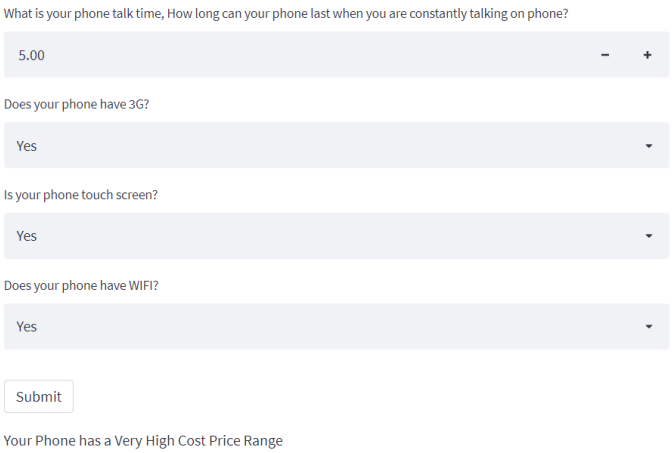
Model Validation and Generalization: Models are evaluated on a separate testing dataset to validate their performance on unseen data and assess their generalization capability. The experimental investigations phase provides valuable insights into the performance, accuracy, and generalization of the smartphone price prediction models.

5 Flow Chart



6 Result





7 ADVANTAGES & DISADVANTAGES

Advantages:

Improved Price Accuracy: Smartphone price prediction models help in achieving more accurate price estimations, enabling consumers to make informed purchasing decisions and sellers to set competitive prices.

Time and Cost Savings: By utilizing predictive models, businesses can save time and resources by automating the price prediction process, eliminating the need for manual analysis and research.

Market Insights: The analysis conducted during the prediction process provides valuable insights into market trends, customer preferences, and factors influencing smartphone prices, helping businesses gain a competitive edge.

Optimized Pricing Strategy: Accurate price predictions enable businesses to develop effective pricing strategies, ensuring optimal balance between profitability and customer demand.

Enhanced Revenue Generation: By setting competitive prices based on accurate predictions, businesses can attract more customers, increase sales, and maximize revenue.

Risk Mitigation: Price prediction models help businesses mitigate the risk of overpricing or underpricing their products, reducing the likelihood of financial losses.

Adaptability: Predictive models can be continuously trained and updated with new data, allowing businesses to adapt to changing market conditions and evolving customer preferences.

Disadvantages:

Data Limitations: The accuracy and effectiveness of price prediction models heavily rely on the availability and quality of data. Incomplete or biased datasets can result in inaccurate predictions.

Model Complexity: Developing and implementing accurate price prediction models requires expertise in data analysis, machine learning, and statistical modeling, making it a complex and resource-intensive process.

Assumption Dependencies: The accuracy of price predictions is based on certain assumptions about market dynamics, consumer behavior, and external factors. Deviations from these assumptions can affect the reliability of predictions.

Market Volatility: Rapid changes in market conditions, technological advancements, or unforeseen events can challenge the predictive capabilities of the models, leading to less accurate price estimations.

Competitive Landscape: Price prediction models may face limitations in highly competitive markets where pricing strategies and fluctuations are influenced by multiple factors and dynamic market forces.

Model Interpretability: Some complex predictive models may lack interpretability, making it difficult to understand the underlying factors influencing price predictions and limiting the ability to make informed decisions based on the model's output.

Data Privacy and Ethics: The use of customer data for price prediction purposes raises concerns about privacy and ethical considerations, requiring businesses to ensure proper data handling and compliance with regulations.

8 APPLICATIONS

E-commerce Platforms: E-commerce platforms can utilize smartphone price prediction models to provide real-time price recommendations to sellers, optimize pricing strategies, and enhance the overall user experience for buyers.

Retail and Online Marketplaces: Retailers and online marketplaces can leverage price prediction to set competitive prices for smartphones, attract customers, and increase sales while maximizing profitability.

Mobile Network Operators: Mobile network operators can utilize price prediction models to optimize pricing plans for smartphones, offering personalized and competitive subscription packages based on predicted device prices.

Secondary Market Platforms: Platforms facilitating the buying and selling of used smartphones can benefit from price prediction to determine fair market values for pre-owned devices, facilitating smoother transactions and negotiations.

Price Comparison Websites: Price comparison websites can incorporate smartphone price prediction to provide users with up-to-date and accurate price comparisons across different sellers and platforms, aiding consumers in making informed purchase decisions.

Mobile App Developers: Developers of mobile applications can leverage price prediction to determine the optimal price points for their apps, ensuring competitiveness while maximizing revenue generation.

Financial Institutions: Financial institutions may utilize smartphone price prediction to assess device values for loan or insurance purposes, enabling more accurate risk assessment and appropriate pricing of financial products.

9 CONCLUSION

In conclusion, smartphone price prediction models offer significant benefits such as improved price accuracy, time and cost savings, optimized pricing strategies, market insights, enhanced revenue generation, risk mitigation, and adaptability. However, challenges exist, including data limitations, model complexity, market volatility, and ethical considerations. Despite these challenges, the applications of price prediction in industries like e-commerce, retail, mobile networks, and financial institutions are valuable. By leveraging price prediction, businesses can make informed pricing decisions, attract customers, increase sales, and stay competitive in a dynamic marketplace. The field of smartphone price prediction holds promise for revolutionizing pricing strategies and enhancing business outcomes.

10 FUTURE SCOPE

The future scope of smartphone price prediction is promising, with potential advancements and opportunities on the horizon. Some areas of future development and exploration include:

Enhanced Predictive Models: Continued advancements in machine learning algorithms, data analysis techniques, and computing power will lead to more accurate and sophisticated price prediction models. These models can incorporate additional variables, such as market trends, consumer sentiment, and product features, to further improve price estimations.

Integration of Real-Time Data: Integrating real-time data sources, such as online market trends, competitor pricing, and consumer demand, can enhance the timeliness and accuracy of price predictions. By incorporating live data feeds, businesses can dynamically adjust prices to align with market conditions and optimize revenue.

Personalized Pricing: The future of smartphone price prediction may involve personalized pricing strategies tailored to individual consumers. By leveraging customer data, including past purchase history, preferences, and demographics, businesses can offer personalized pricing options to maximize customer satisfaction and loyalty.

Automated Pricing Systems: Automated pricing systems driven by predictive models can enable businesses to dynamically adjust prices in real-time based on market dynamics, demand-supply factors, and competitor pricing. This automation can save time and resources while ensuring optimal pricing strategies.

Integration with Augmented Reality (AR): The integration of smartphone price prediction with AR technology can enable consumers to visualize and compare product prices in real-world environments. AR applications can provide real-time pricing information when scanning or interacting with smartphone products, enhancing the shopping experience.

Ethical and Privacy Considerations: As the field progresses, ensuring ethical data handling practices and privacy protection will become increasingly important. Future developments will focus on maintaining transparency, data anonymization, and complying with regulations to protect consumer privacy while utilizing predictive models.

11a BIBLIOGRAPHY

[How to Make Predictions with scikit-learn - MachineLearningMastery.com]

[Machine Learning Algorithms - Javatpoint]

11b APPENDIX

```
import streamlit as st
import pandas as pd
```

```
import numpy as np
import sklearn
import pickle
model = pickle.load(open('model.pkl','rb'))
st.title("Phone Price Range Prediction")
st.markdown(" This is a simple web application that predicts the price
range of your mobile phone based on the phone's specifications.")

st.markdown("The price ranges are given as follows: ")

st.markdown(
"""
* Low Cost
* Medium Cost
* High Cost
* Very High Cost
""")

st.markdown("The data used for this project was collected from
[Kaggle](https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-cl
assification) and machine learning was employed to predict the price
ranges based on the data obtained.")



battery_power = st.number_input("Enter Battery Power in mAh")
bluetooth = st.selectbox("Does your phone have bluetooth?", ("Yes","No"))
if bluetooth == "Yes":

    blue = 1

else:

    blue = 0

clock_speed = st.number_input("What is the microprocessor speed",
value=2.0)

sim = st.selectbox("Does your phone have Dual Sim?",("Yes","No"))

if sim == "Yes":

    dual_sim = 1

else:
```

```
    dual_sim = 0
fc = st.number_input("What is your front camera mega pixels?")
speed= st.selectbox("Is your phone 4G?",("Yes","No"))
if speed == "Yes":
    four_g = 1
else:
    four_g = 0
int_memory = st.number_input("What is your internal memory in GB?")
m_dep = st.number_input("What is your phone's mobile depth in cm?")
mobile_wt = st.number_input("What is the weight of your phone?")
n_cores = st.number_input("What is the number of cores of processor?")
pc = st.number_input("What is the main camera quality in mega pixels?")
px_height = st.number_input("What is the pixels resolution height?")
px_width = st.number_input("What is the pixels resolution width?")
ram = st.number_input ("What is the ram in MB?")
sch = st.number_input("What is the screen height in cm?")
scw = st.number_input("What is the screen width in cm?")
talk_time = st.number_input("What is your phone talk time, How long can
your phone last when you are constantly talking on phone?")
speed_2 = st.selectbox("Does your phone have 3G?",("Yes","No"))
if speed_2 == "Yes":
    three_g = 1
else:
    three_g = 0
ts= st.selectbox("Is your phone touch screen?",("Yes","No"))
if ts == "Yes":
```

```
    touch_screen = 1
else:
    touch_screen = 0
w = st.selectbox("Does your phone have WIFI?",("Yes","No"))
if w == "Yes":
    wifi = 1
else:
    wifi = 0
hide_streamlit_style = """
            <style>
            footer {visibility: hidden;}
            </style>
            """
st.markdown(hide_streamlit_style, unsafe_allow_html=True)
data = [[battery_power, blue, clock_speed, dual_sim, fc, four_g,
int_memory, m_dep, mobile_wt, n_cores, pc, px_height, px_width,ram, sch,
scw, talk_time, three_g, touch_screen, wifi]]
result = model.predict(data)
if result[0] == 0:
    result_2 = "Low Cost"
elif result[0] == 1:
    result_2 = "Medium Cost"
elif result [0] == 2:
    result_2 = "High Cost"
else:
    result_2 = "Very High Cost"
```

```
if st.button('Submit'):
    st.write("Your Phone has a {} Price Range".format(result_2))
else:
    st.write('')
```

CODE EXPLANATION:

The code begins by importing necessary libraries such as Streamlit, Pandas, NumPy, scikit-learn, and pickle. These libraries are commonly used for web development, data manipulation, numerical operations, machine learning, and serializing Python objects, respectively.
The pickle.load() function is used to load a pre-trained machine learning model stored in a file called 'model.pkl'. This model will be used to make predictions on new data.
The code sets up the title and introductory text for the web application using st.title() and st.markdown() functions from the Streamlit library.
The markdown section provides an overview of the price ranges and the source of the data used for the project.
The code then collects user input for various phone specifications such as battery power, Bluetooth availability, clock speed, dual SIM support, front camera megapixels, 4G support, internal memory, mobile depth, weight, number of processor cores, main camera quality, pixel resolution height and width, RAM, screen height and width, talk time, 3G support, touch screen availability, and Wi-Fi availability. It uses Streamlit's input components like st.number_input() and st.selectbox() to gather this information.
A hidden Streamlit style is added to remove the default footer from the application's appearance using the st.markdown() function with HTML code.
The user input data is collected and stored in the data variable as a list of lists.
The model.predict() function is used to make a prediction on the input data. The predicted result represents the price range of the phone.
Based on the predicted result, the code assigns a corresponding price range label to the variable result_2.
If the user clicks the 'Submit' button, the predicted price range is displayed using st.write().
If the user does not click the 'Submit' button, nothing is displayed.
Overall, this code sets up a web application that takes user input for various phone specifications and uses a pre-trained machine learning model to predict the price range of the phone based on those specifications.