MOBILE PHONE PRICE PREDICTION

BY

ROHAN REDDY MELACHERVU

A project report submitted to

Dr. BHARADWAJA KUMAR

SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE ENGINEERING

<u>In partial fulfillment of the requirements for the course of CSE3025-</u>

<u>LARGE SCALE DATA PROCESSING</u>

VIT UNIVERSITY, CHENNAI CAMPUS

Vandalur-Kelambakkam Road

Chennai-600127

MAY-2021

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report entitled “Mobile Phone Price Prediction” is a Bonafede work of **ROHAN REDDY MELACHERVU (19BCE1191), PABBATHI SAI TEJA (19BCE1211),** who carried out the J-component under my supervision and guidance.

Dr. BHARADWAJA KUMAR

Associate Professor

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING (SCSE)

VIT UNIVERSITY, CHENNAI CAMPUS

CHENNAI-600127

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Bharadwaja Kumar,** Professor, SCSE, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to the Dean of the SENSE, VIT Chennai, for extending the facilities of the school towards our project and for the unstinting support. We also take this opportunity to thank all the faculty of the school for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

ABSTRACT

We will be downloading the dataset from Kaggle. The dataset would have over 1000 rows. We will then configure Apache Hadoop in a Virtual Machine, and then write the corresponding Mapper and Reducer codes and apply various Machine Learning Algorithms on the dataset to find out the accuracy.

In this modern era, smartphones are an integral part of the lives of human beings. When a smartphone is purchased, many factors like the display, processor, memory, camera, thickness, battery, connectivity and others are taken into account.

One factor that people do not consider is whether the product is worth the cost. As there are no resources to cross-validate the price, people fail in taking the correct decision.

This not only helps the customers decide the right phone to purchase, it also helps the owners decide what should be the appropriate pricing of the phone for the features that they offer.

This idea of predicting the price will help the people make informed choice when they are purchasing a phone in the future.

INDEX

1. INTRODUCTION

Purchase of smartphone has always been an issue encountered by all in some instance of time.

People spend a lot of time thinking and cross-checking with their peers about the product.

People are often in dilemma whether the features provided by the manufacturer of the phone are really worth the cost of buying.

Even manufacturers have the problem of deciding the price. We have seen many times where a phone has less sales because of its expensive price.

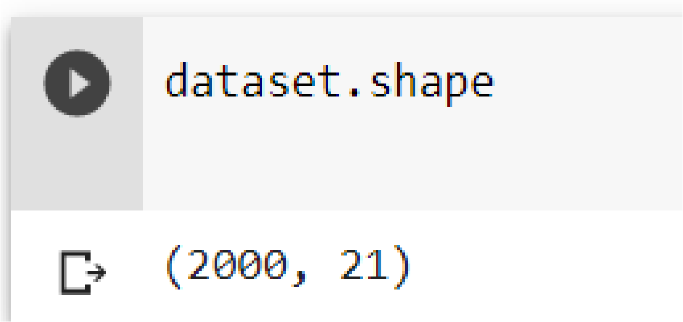
Using Hadoop Framework, we can achieve the prediction model easily.

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures. In our case, we won’t be using multiple computers for computation as in our Virtual Machine we use Apache Hadoop Pseudo mode. The Map Reduce module in the Apache Hadoop Framework helps us to easily process this huge data.

2. DATASET

On the basis of the mobile specification like battery power, 4G enabled, Wi-fi, Bluetooth, RAM etc we are predicting price range of the mobile. The dataset has 2000 rows and 21 columns It's in the form of CSV. The dataset has no NULL values and all the columns are of the form int or float so there is no requirement of data pre-processing.

The link to dataset: https://www.kaggle.com/iabhishekofficial/mobile-price-classification



```
dataset.shape
```

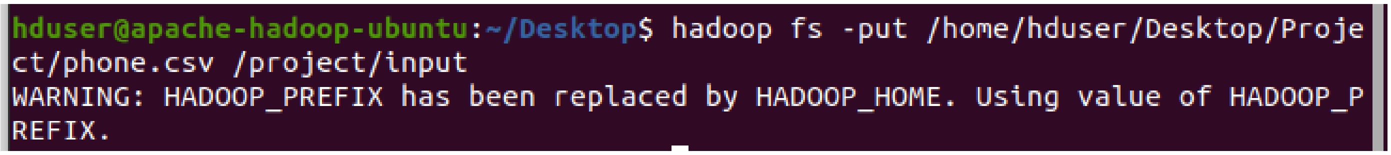
```
(2000, 21)
```

3. METHODOLOGY

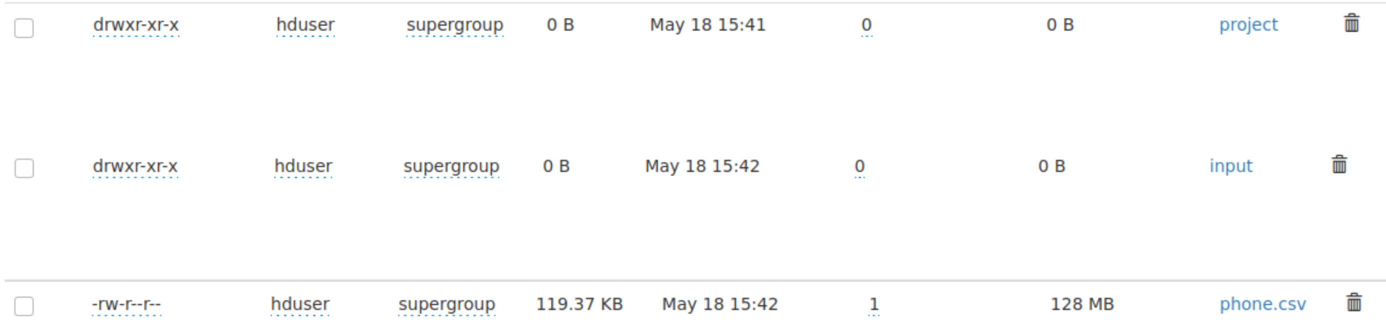
3.1 STRATEGY:

We have downloaded the dataset in our Virtual Machine, the main strategy here is to upload the dataset into the Hadoop Distributed File System and then use Mapper to send the big data to the Reducer, the Reducer then will take the output of Mapper as input, parse it and then perform Machine-Learning algorithms on it.

Uploading the dataset into HDFS:



```
hduser@apache-hadoop-ubuntu:~/Desktop$ hadoop fs -put /home/hduser/Desktop/Proje
ct/phone.csv /project/input
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_P
REFIX.
```



□	drwxr-xr-x	hduser	supergroup	0 B	May 18 15:41	0	0 B	project	🗑
□	drwxr-xr-x	hduser	supergroup	0 B	May 18 15:42	0	0 B	input	🗑
□	-rw-r--r--	hduser	supergroup	119.37 KB	May 18 15:42	1	128 MB	phone.csv	🗑

4. CODE

4.1 Mapper Code: The Mapper will read the CSV file from the HDFS line by line and sends it to the Reducer.

```
#!/usr/bin/env python3
"""mapper.py"""
import sys
count = 0
for line in sys.stdin:
    line = line.strip()
    print('%s\t' % count,line)
    count = count+1
```

4.2 Reducer Code: The Reducer takes output from the Mapper as input, parses it and by importing various Python libraries, perform Machine Learning algorithms on it to find out the accuracy.

```
#!/usr/bin/env python3
"""reducer.py"""
from operator import itemgetter
import sys
import pandas as pd
import numpy as np
from sklearn.metrics import classification_report as cr
from sklearn.metrics import accuracy_score as As
import warnings
```

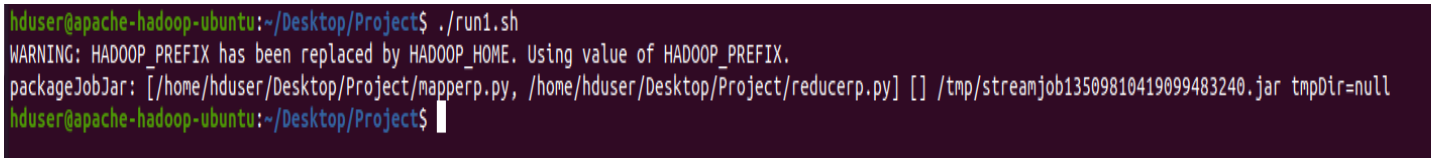
```
warnings.filterwarnings("ignore")
y = []
c = 0
container = []
row = []
for line in sys.stdin:
    line = line.strip()
    count,word = line.split('\t', 1)
    words = word.split(",")
 np_words = np.array(words).astype("float")
 container.append(np_words)
df = pd.DataFrame(container)
X = df.iloc[:,1:-1].values
y = df.iloc[:,-1].values
y = y.astype(int)

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = .2 , random_state = 0)

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
from sklearn.neighbors import KNeighborsClassifier

neigh = KNeighborsClassifier(n_neighbors=3)

neigh.fit(X_train, y_train)

print("KNN Model")

print(cr(y_test,neigh.predict(X_test)))

from sklearn.linear_model import LogisticRegression

clf = LogisticRegression(random_state=0).fit(X_train, y_train)

print("Logistic Regression model")

print(cr(y_test,clf.predict(X_test)))

from xgboost import XGBClassifier

xg = XGBClassifier(eval_metric="logloss")

xg.fit(X_train, y_train)

print("XGB Model")

print(cr(y_test,xg.predict(X_test)))
```

4.3 Run command: This is the .sh command we use to run the Mapper and Reducer codes on the file which is in the Hadoop Distributed File System.

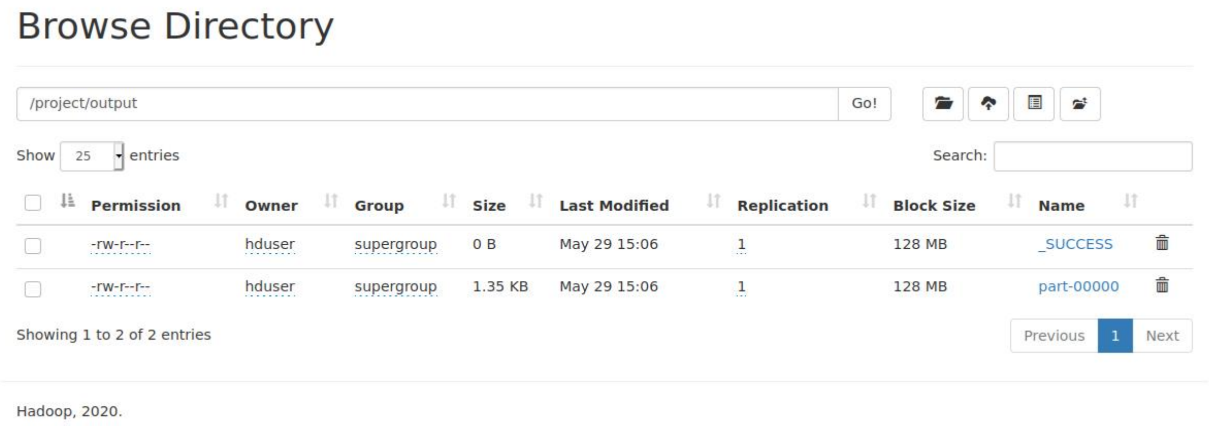
```
hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.0.jar -file

/home/hduser/Desktop/Project/mapperp.py -mapper /home/hduser/Desktop/Project/mapperp.py -file

/home/hduser/Desktop/Project/reducerp.py -reducer /home/hduser/Desktop/Project/reducerp.py -input

/project/input/phone.csv -output /project/output
```

4.4 Executing the run command



```
hduser@apache-hadoop-ubuntu:~/Desktop/Project$ ./run1.sh
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
packageJobJar: [/home/hduser/Desktop/Project/mapperp.py, /home/hduser/Desktop/Project/reducerp.py] [] /tmp/streamjob13509810419099483240.jar tmpDir=null
hduser@apache-hadoop-ubuntu:~/Desktop/Project$ 
```

5. Output



Browse Directory

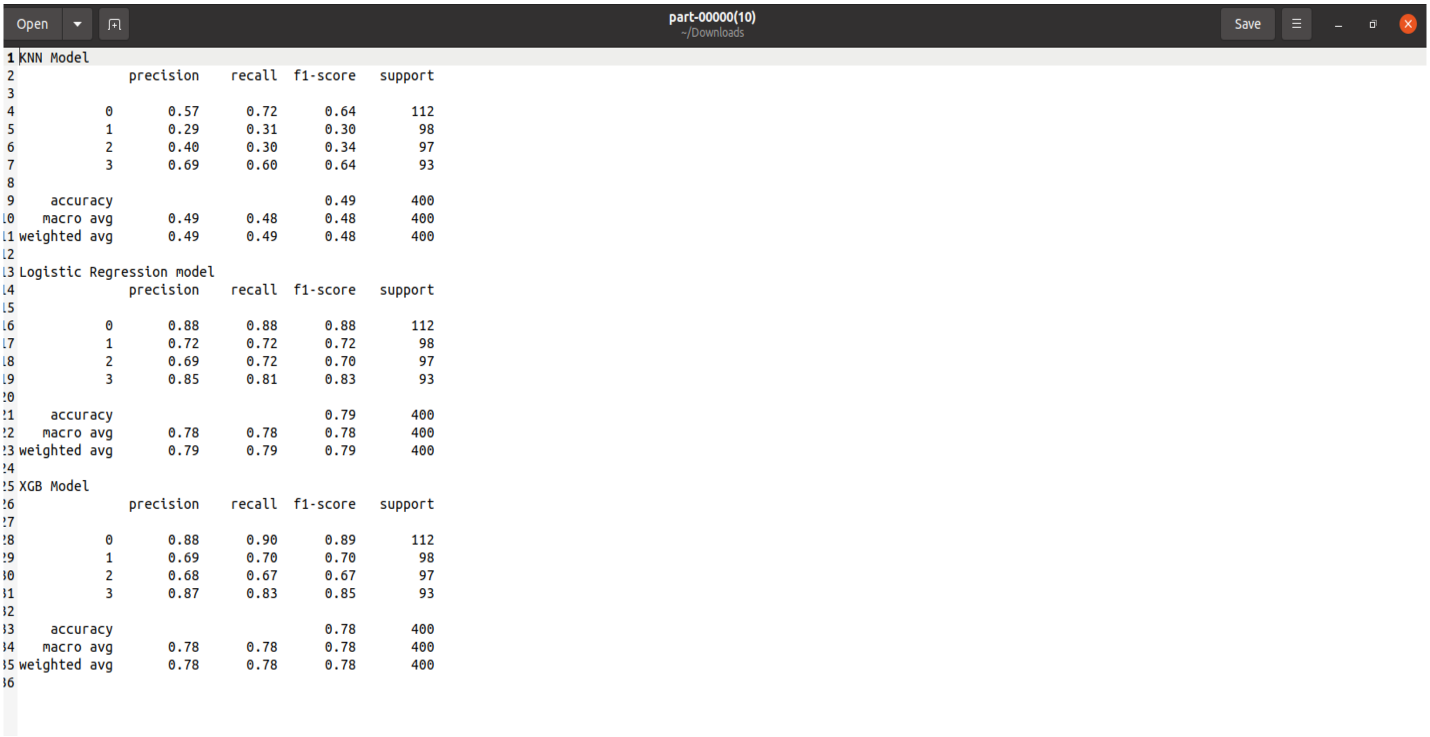
/project/output Go!

Show 25 entries Search:

☐	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
☐	-rw-r--r--	hduser	supergroup	0 B	May 29 15:06	1	128 MB	_SUCCESS
☐	-rw-r--r--	hduser	supergroup	1.35 KB	May 29 15:06	1	128 MB	part-00000

Showing 1 to 2 of 2 entries

Hadoop, 2020.



```
KNN Model
              precision    recall  f1-score   support

           0       0.57      0.72      0.64       112
           1       0.29      0.31      0.30        98
           2       0.40      0.30      0.34        97
           3       0.69      0.60      0.64        93

    accuracy                           0.49       400
   macro avg       0.49      0.48      0.48       400
weighted avg       0.49      0.49      0.48       400

Logistic Regression model
              precision    recall  f1-score   support

           0       0.88      0.88      0.88       112
           1       0.72      0.72      0.72        98
           2       0.69      0.72      0.70        97
           3       0.85      0.81      0.83        93

    accuracy                           0.79       400
   macro avg       0.78      0.78      0.78       400
weighted avg       0.79      0.79      0.79       400

XGB Model
              precision    recall  f1-score   support

           0       0.88      0.90      0.89       112
           1       0.69      0.70      0.70        98
           2       0.68      0.67      0.67        97
           3       0.87      0.83      0.85        93

    accuracy                           0.78       400
   macro avg       0.78      0.78      0.78       400
weighted avg       0.78      0.78      0.78       400
```

We achieved 49% accuracy with the KNN model, and 78% accuracy with the Logistic Regression and XGB Models.

6. CONCLUSION

Using Hadoop, we could easily process the large dataset, and then apply Machine Learning algorithms on the dataset to get a decent accuracy. In this Project, we learnt how to configure, use and code in Hadoop framework, how Hadoop processes large datasets with ease and how to code using Hadoop framework to apply your own logic.