Data Collection and Preprocessing Phase

Date	JULY 10 2024
Team ID	865503
Project Title	FrappeActivity: Mobile Phone Activity Classification Using Machine Learning
Maximum Marks	6 Marks

PreparationTemplate

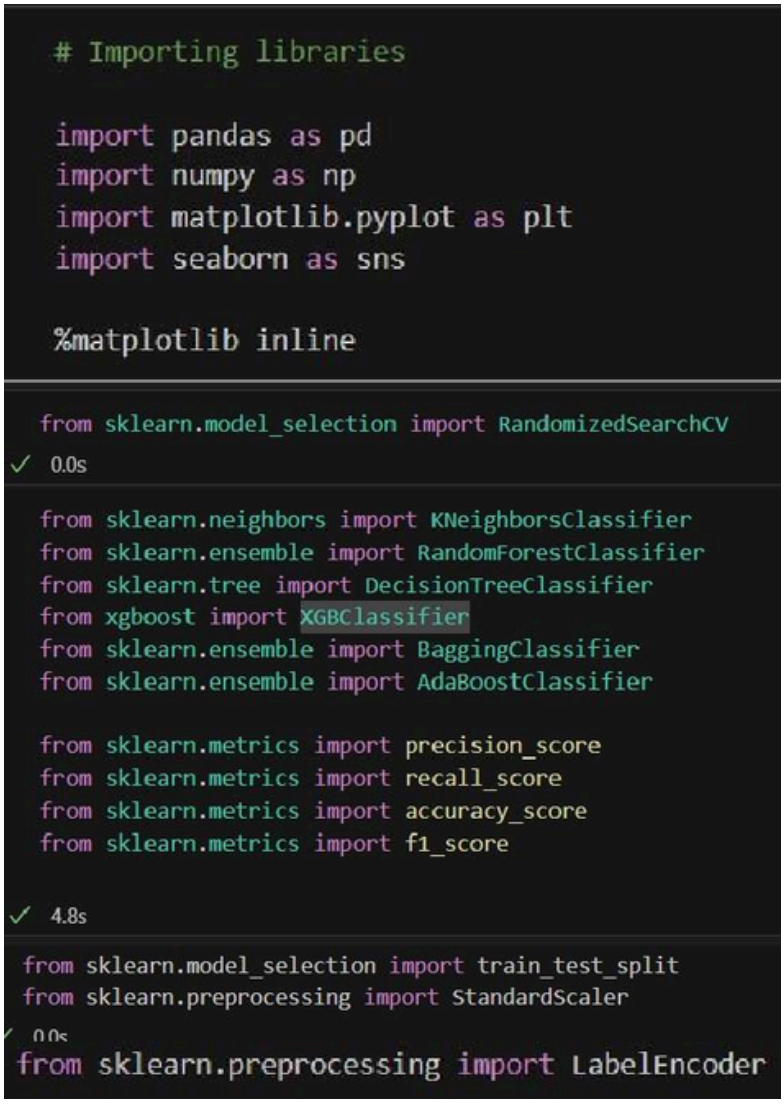
To classify mobile phone activities using machine learning, define the activities, collect sensor data, preprocess by segmenting and extracting features, analyze data distributions, choose and train models like Random Forest or Bagging Classifier, validate with accuracy metrics, deploy in a mobile app, optimize for performance, test in real-world scenarios, and document thoroughly for transparency and future enhancements.

Section	Description

Data Overview	There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc. In this project we have used .csv data.
Data Preparation	These are the general steps of pre-processing the data before using it for machine learning
Handling missing values	We use Handling missing values For checking the null values
Handling categorical data	As we can see our dataset has categorical data we must convert the categorical data to integer encoding or binary encoding
Handling Outliers in Data	With the help of boxplot, outliers are visualized. And here we are going to find upper bound and lower bound of numerical features with some mathematical formula.

Data Preparation

Collect the dataset	Please refer to the link given below to download the dataset. Link: https://github.com/irecsys/CARSKit/blob/master/c ontext-aware_data_sets/Mobile_Frappe.zip

Importing the libraries	

```
# Importing libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

```
from sklearn.model_selection import RandomizedSearchCV
```

✓ 0.0s

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import AdaBoostClassifier

from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
```

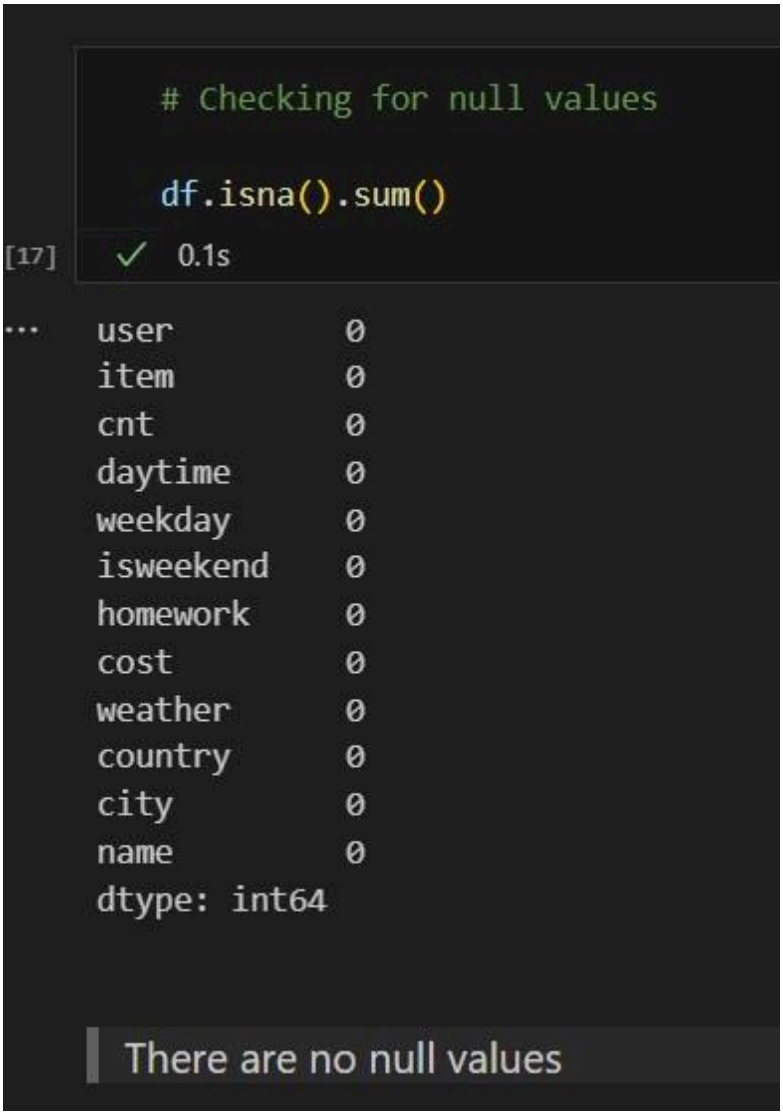
✓ 4.8s

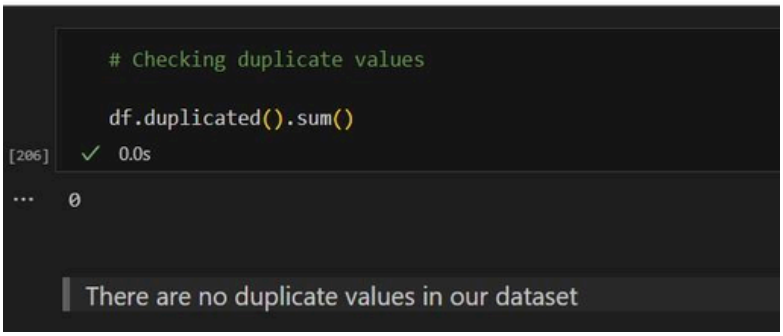
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

✓ 0.0s

```
from sklearn.preprocessing import LabelEncoder
```

Loading Data	We use the code Link: https://github.com/irecsys/CARSKit/blob/master/c__ontext-aware_data_sets/Mobile_Frappe.zip For reading the dataset

Handling missing values	

Handling Categorical values	
Handling Duplicate Values	

```
dt_encoder=LabelEncoder()
dt_encoder.fit(df['daytime'])
df['daytime']=dt_encoder.transform(df['daytime'])
wd_encoder=LabelEncoder()
wd_encoder.fit(df['weekday'])
df['weekday']=wd_encoder.transform(df['weekday'])
wknd_encoder=LabelEncoder()
wknd_encoder.fit(df['isweekend'])
df['isweekend']=wknd_encoder.transform(df['isweekend'])
hw_encoder=LabelEncoder()
hw_encoder.fit(df['homework'])
df['homework']=hw_encoder.transform(df['homework'])
c_encoder=LabelEncoder()
c_encoder.fit(df['cost'])
df['cost']=c_encoder.transform(df['cost'])
w_encoder=LabelEncoder()
w_encoder.fit(df['weather'])
df['weather']=w_encoder.transform(df['weather'])
n_encoder=LabelEncoder()
n_encoder.fit(df['name'])
df['name']=n_encoder.transform(df['name'])
✓ 0.1s
```

```
# Checking duplicate values

df.duplicated().sum()
[206] ✓ 0.0s
... 0
```

There are no duplicate values in our dataset