

# Data Collection and Preprocessing Phase

Date	June 2024
Team ID	739861
Project Title	<b>FrappeActivity: Mobile Phone Activity Classification Using Machine Learning</b>
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 SVM, 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.
<b>Data Preparation</b>	
Collect the dataset	Please refer to the link given below to download the dataset. Link: <a href="https://github.com/irecsys/CARSSKit/blob/master/context-aware_data_sets/Mobile_Frappe.zip">https://github.com/irecsys/CARSSKit/blob/master/context-aware_data_sets/Mobile_Frappe.zip</a>

<p>Importing the libraries</p>	<pre># 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</pre>
<p>Loading Data</p>	<p>We use the code Link: <a href="https://github.com/irecsys/CARSKit/blob/master/context-aware_data_sets/Mobile_Frappe.zip">https://github.com/irecsys/CARSKit/blob/master/context-aware_data_sets/Mobile_Frappe.zip</a> For reading the dataset</p>

Handling missing values

```
# Checking for null values

df.isna().sum()

[17] ✓ 0.1s

... user      0
    item      0
    cnt       0
    daytime   0
    weekday   0
    isweekend  0
    homework  0
    cost      0
    weather   0
    country   0
    city      0
    name      0
    dtype: int64
```

There are no null values

<p>Handling Categorical values</p>	<pre>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'])</pre> <p>✓ 0.1s</p>
<p>Handling Duplicate Values</p>	<pre># Checking duplicate values df.duplicated().sum()</pre> <p>[206] ✓ 0.0s</p> <p>... 0</p> <p>There are no duplicate values in our dataset</p>