Frappe Activity: Mobile Phone Activity Classification

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATION

In

COMPUTER SCIENCE AND ENGINEERING (MCA)

Submitted By

SAISRI MANDHALA **23UK1F0016**

Under the guidance of

DR. R. NAVEEN KUMAR

Professor, Head CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(MCA)

VAAGDEVI ENGINEERING COLLEGE (AUTONOMOUS)

Affiliated to JNTUH, HYDERABD,, WARANGAL (T.S) 506005

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (MCA) VAAGDEVI ENGINEERING COLLEGE (AUTONOMOUS)

CERTIFICATE OF COMPLETION PROJECT WORK REVIEW-I

This is to certify that the UG Project Phase-1 entitled MOBILE PHONE ACTIVITY CLASSIFICATION is being submitted by M. SaiSri (23UK1F0016) in partial fulfilment of the requirements for the award of the degree of Master of computer application in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2023- 2024.

Project Guide	**HOD**
DR. R. Naveen Kumar	**DR. R. NaveenKumar**
(Professor)	(Professor)

External

ACKNOWLEDGEMENT

I wish to take this opportunity to express our sincere gratitude and deep sense of respect to my beloved **DR Syed Musthak Ahmed,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

I extend my heartfelt thanks to **DR. R. Naveen Kumar**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

I express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

I express heartfelt thanks to the guide, **DR. R. Naveen Kumar,** professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, I express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

SAISRI MANDHALA **23UK1F0016**

ABSTRACT

Mobile Phone Activity Classification project aims to develop a system for accurately classifying user activities based on mobile phone sensor data. Utilizing data from various sensors, such as accelerometers, gyroscopes, and GPS, the system seeks to identify and categorize activities like walking, running, driving, and stationary states. By leveraging machine learning algorithms and data preprocessing techniques, the project aims to create a robust model that can be deployed in mobile applications to enhance user experience through activity recognition. Potential applications include fitness tracking, context-aware services, and personalized recommendations. The project underscores the importance of real-time data processing and the challenges of handling noisy and heterogeneous sensor data.

TABLE OF CONTENTS:-

1.INTRODUCTION

1.1. OVERVIEW

Mobile Phone Activity Classification is an advanced technology designed to predict user activities by analysing sensor data from mobile devices. This project specifically focuses on classifying activities into known, unknown, and working categories, including specific tasks like using Instagram or shopping.

The core of this technology relies on various sensors embedded in mobile devices, such as accelerometers, gyroscopes, GPS, magnetometers, and proximity sensors. These sensors collect data on movement, orientation, location, and nearby objects, providing a comprehensive dataset for activity recognition. The data is continuously collected and sampled at regular intervals to capture user activities in real time.

Data preprocessing is a crucial step in this process. It involves filtering out noise to remove irrelevant data, segmenting the continuous data stream into manageable windows, and extracting features that are indicative of different activities, such as mean, variance, and energy. These steps ensure that the data is clean, structured, and ready for analysis.

Machine learning algorithms play a key role in this classification system. Supervised learning algorithms are trained on labelled datasets where activities are pre-identified, including decision trees, support vector machines (SVM), and neural networks. These models learn patterns in the sensor data that correspond to specific activities. In cases where activities are not pre-labelled, unsupervised learning algorithms, such as K-means clustering, are used to identify patterns and cluster similar activities together.

1.2. PURPOSE

The purpose of the Mobile Phone Activity Classification project is to develop a sophisticated system that accurately identifies and categorizes user activities using sensor data from mobile devices. This system is designed to classify activities into known, unknown, and working categories, and to detect specific tasks such as using Instagram or shopping.

The main objectives of the project are:

1. **Enhancing User Experience**: Improve mobile applications by adapting features based on real-time activity classification, such as switching to a specific mode during work or offering tailored recommendations.
2. **Monitoring Health and Fitness**: Track and analyse physical activities to provide personalized fitness suggestions and insights into daily routines, helping users achieve their health goals.
3. **Ensuring Safety and Monitoring**: Implement features that detect unusual activities or patterns, particularly useful for safety applications like elderly care, by alerting caregivers to potential issues.
4. **Supporting Behavioural Research and Analysis**: Provide tools for researchers and psychologists to study user behaviour patterns, offering insights into daily routines, habits, and behavioural changes for mental health studies, lifestyle analysis, and user experience research.
5. **Enabling Context-Aware Services**: Facilitate the creation of context-aware applications that adjust device settings and functionalities based on the user's current activities, enhancing overall application usability.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

1. **Technical Challenges**

   - **Sensor Data Accuracy**: The accuracy of sensor data varies across different mobile devices and can be affected by hardware limitations, environmental conditions, and user behaviour. For example, accelerometers and gyroscopes may have calibration issues, leading to inaccurate activity detection.

2. **Practical Issues**

   - **User Privacy Concerns**: Collecting and analysing sensor data raises significant privacy issues. Users must trust that their data is handled securely and used responsibly, which can be a barrier to the adoption of activity classification technologies.
   - **Battery Consumption**: Continuous sensor data collection and processing can lead to increased battery consumption. Designing systems that balance the need for data accuracy with minimal battery drain is a critical practical issue.

3. **Ethical and Legal Considerations**

   - **Data Security and Consent**: Ensuring that data collection complies with legal regulations and obtaining informed consent from users is essential. Mismanagement of user data or lack of transparency can lead to legal issues and loss of user trust.
   - **Bias in Machine Learning Models**: Machine learning models can inherit biases from training data, leading to unfair or inaccurate activity classification for certain user groups. Addressing these biases to ensure fair and equitable technology is a significant challenge.

4. **Scalability and Maintenance**

   - **Scalability of Solutions**: Developing systems that can scale to support millions of users while maintaining performance and accuracy is a significant challenge. As the user base grows, the system must handle increased data loads and diverse activity patterns.

2.2 PROPOSED SOLLUTION

1. Technical Challenges

- **Calibration Algorithms**: Implement advanced calibration techniques to correct sensor inaccuracies. Regularly recalibrate sensors to maintain accuracy.
- **Sensor Fusion**: Combine data from multiple sensors to improve overall accuracy. Techniques like Kalman filtering or sensor fusion algorithms can merge accelerometer, gyroscope, and magnetometer data for more reliable activity detection.

2. Practical Issues

- **Data Encryption**: Encrypt sensor data during collection, transmission, and storage to protect user privacy.
- **Anonymization Techniques**: Implement data anonymization techniques to ensure that personal information is not exposed.

3. Ethical and Legal Considerations

- **Clear Consent Processes**: Develop clear, user-friendly consent forms that explain what data is collected, how it is used, and how users can opt out.
- **Compliance with Regulations**: Ensure that data collection practices comply with regulations such as GDPR and CCPA.

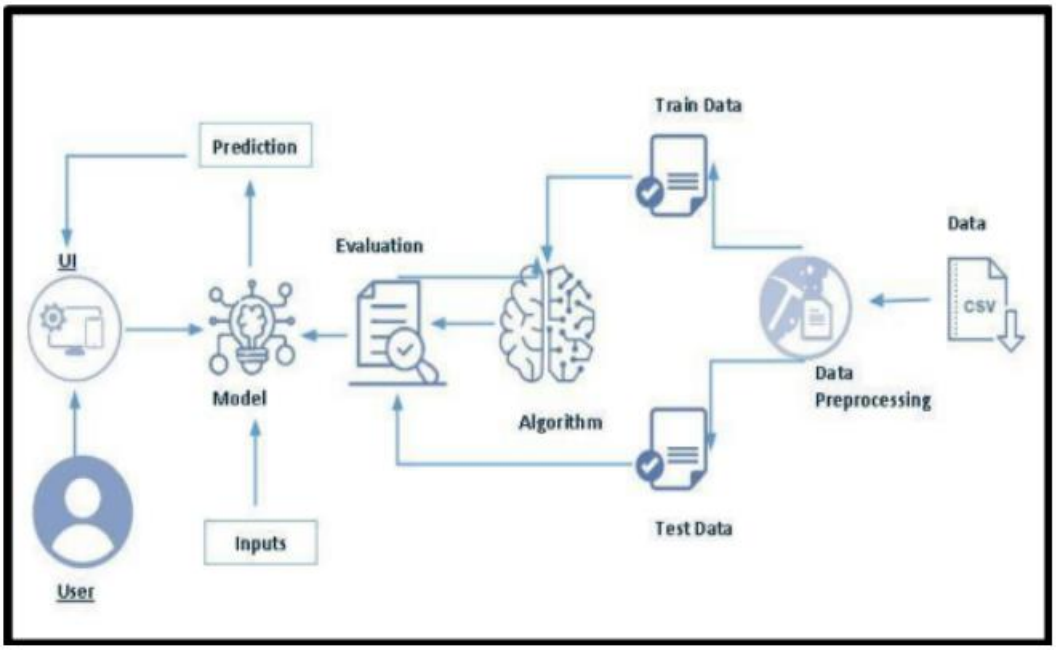
4. Scalability and Maintenance

- **Cloud-Based Solutions**: Use cloud services to manage increasing data loads and user bases, ensuring that the infrastructure can scale effectively.
- **Modular System Design**: Design systems with modular components that can be independently scaled or updated.

- Addressing the challenges in mobile phone activity classification involves a range of solutions, from technical approaches like sensor fusion and advanced algorithms to practical measures such as privacy protection and energy-efficient design. By implementing these solutions, developers can create more accurate, reliable, and user-friendly activity classification systems.

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



BLOCK DIAGRAM

3.2. SOFTWARE DESIGNING

The following is the Software required to complete this project:

- **Google Collab**: Google Collab will serve as the development and execution environment for your predictive modelling, data preprocessing, and model training tasks. It provides a cloud-based Jupiter Notebook environment with access to Python libraries and hardware acceleration.

- **Dataset (CSV File)**: The dataset in CSV format is essential for training and testing your predictive model. It should include historical air quality data, weather information, pollutant levels, and other relevant features.

- **Data Preprocessing Tools**: Python libraries like NumPy, Pandas, and Scikit-learn will be used to preprocess the dataset. This includes handling missing data, feature scaling, and data cleaning.

- **Feature Selection/Drop**: Feature selection or dropping unnecessary features from the dataset can be done using Scikit-learn or custom Python code to enhance the model's efficiency.

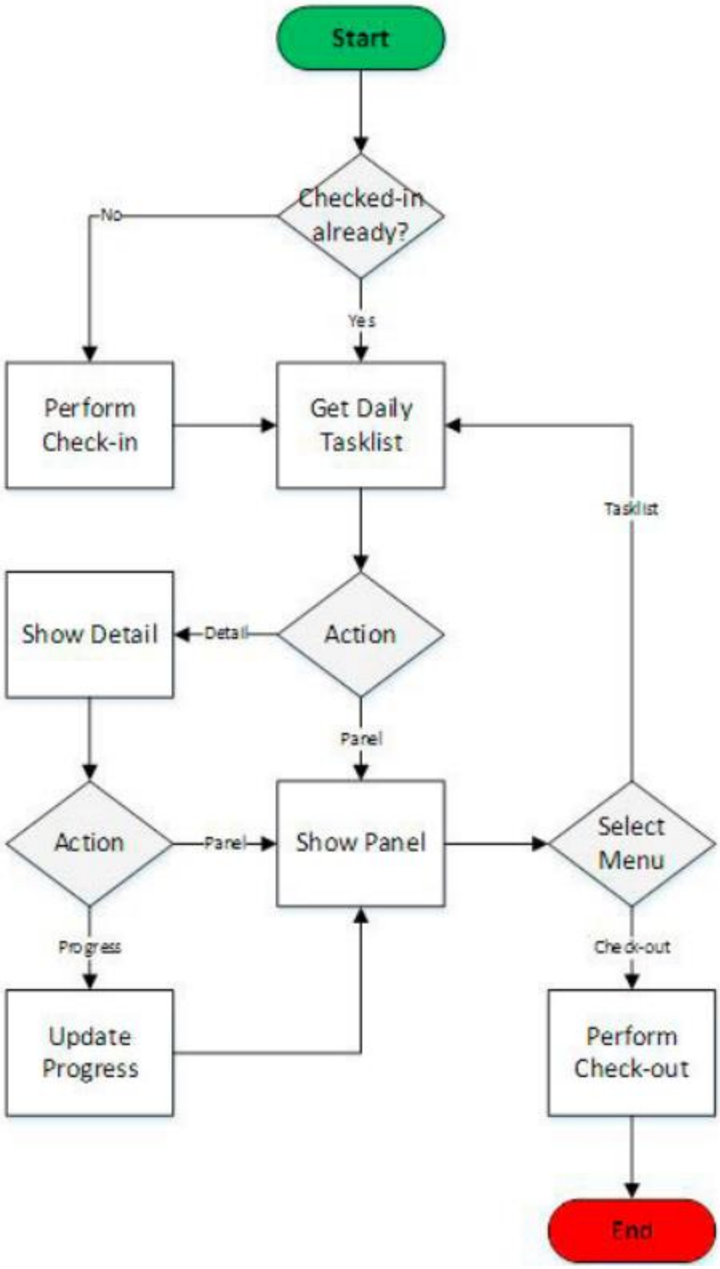
- **Model Training Tools**: Machine learning libraries such as Scikit-learn, TensorFlow, or Torch will be used to develop, train, and fine-tune the predictive model. Regression or classification models can be considered, depending on the nature of the AQI prediction task.

- **Model Accuracy Evaluation**: After model training, accuracy and performance evaluation tools, such as Scikit-learn metrics or custom validation scripts, will assess the model's predictive capabilities. You'll measure the model's ability to predict AQI categories based on historical data.

- **UI Based on Flask Environment**: Flask, a Python web framework, will be used to develop the user interface (UI) for the system. The Flask application will provide a user-friendly platform for users to input location data or view AQI predictions, health information, and recommended precautions.

- Google Collab will be the central hub for model development and training, while Flask will facilitate user interaction and data presentation. The dataset, along with data preprocessing, will ensure the quality of the training data, and feature selection will optimize the model. Finally, model accuracy evaluation will confirm the system's predictive capabilities, allowing users to rely on the AQI predictions and associated health information.

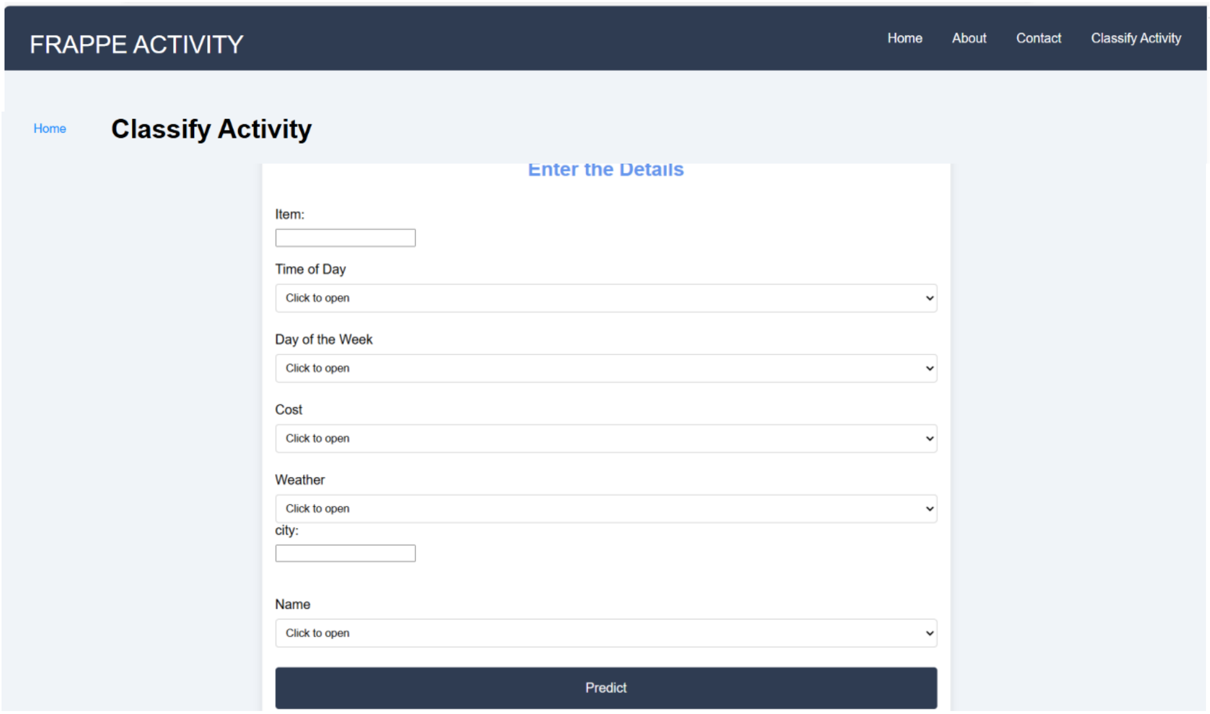
4.FLOWCHART



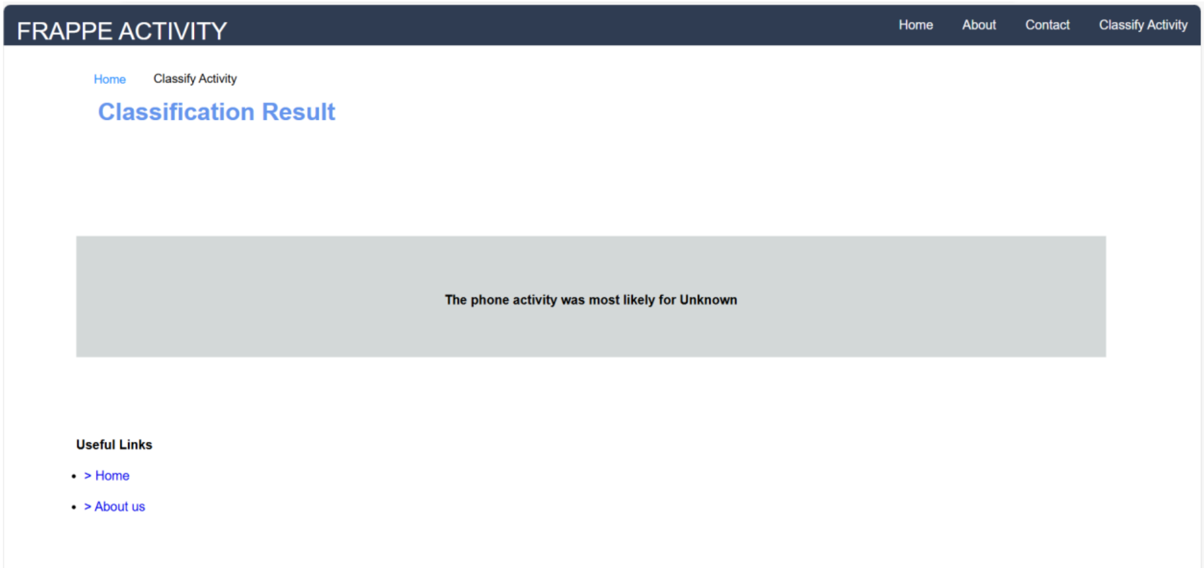
5.RESULT

HOME PAGE

Predict page



Prediction page



6.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

1. Enhanced User Experience

- **Context-Aware Services**: These systems enable mobile applications to adapt based on user activities, such as switching to a “driving mode” or providing contextual recommendations.

- **Personalization**: Applications can offer tailored experiences and suggestions, enhancing user satisfaction and engagement.

2. Health and Fitness Monitoring

- **Activity Tracking**: Monitors physical activities such as walking, running, and cycling, helping users meet their fitness goals.

- **Health Insights**: Provides feedback on activity patterns and suggests improvements for a healthier lifestyle.

3. Behavioural Research and Analysis

- **Detailed Insights**: Offers valuable data for understanding user behaviour patterns, which can be used in mental health research and lifestyle studies.

- **Data for Research**: Provides data for academic research in psychology, sociology, and human-computer interaction.

4. Scalability and Adaptability

- **Scalable Solutions**: Cloud-based architectures allow the system to handle increasing user data and activity patterns.
- **Adaptable Models**: Transfer learning techniques enable models to adapt to new environments and user groups

DISADVANTAGES:

1. Technical Challenges

- **Sensor Data Accuracy**: Variations in sensor accuracy can lead to incorrect activity classifications.
- **Real-Time Processing Requirements**: Ensuring that algorithms process data in real-time without draining the battery.

2. Privacy Concerns

- **Data Privacy Issues**: Collecting and storing personal sensor data raises privacy concerns.
- **Compliance Challenges**: Ensuring adherence to data protection regulations like GDPR and CCPA.

3.Battery Consumption

- **Issue:** Continuous sensor data collection and processing can significantly drain the mobile device's battery.
- **Impact:** Frequent battery depletion can lead to a poor user experience.

4. Model Bias

- **Issue:** Machine learning models may inherit biases from training data, leading to unfair or inaccurate activity classifications.
- **Impact:** Bias can result in unequal performance across different user groups

7.APPLICATIONS

1. **Personal Assistant Applications**: Mobile phone activity classification systems can be used to enhance personal assistant applications by providing context-aware features. These assistants can adjust their functionalities based on the detected activity, offering relevant services or information.

2. **Safety and Emergency Services:** Activity classification systems can be used for fall detection and emergency alert systems for elderly individuals.

3. **Behavioural Analysis and Research:** Activity classification systems can be used for research into human behaviour, habits, and lifestyle pattern

4.**Enterprise Solutions:** Activity classification systems can be used in enterprise solutions to improve productivity and manage workflows.

5. **Educational Tools:** Using activity classification for educational purposes, including learning tools and teaching aids.

8.CONCLUSION

Mobile phone activity classification systems represent a significant advancement in the way we understand and interact with technology. By leveraging sensor data and machine learning algorithms, these systems can accurately detect and categorize a wide range of user activities, leading to a multitude of applications across various domains. From personal assistant applications that enhance daily convenience to sophisticated fitness and wellness apps that help users achieve health goals, the potential uses are diverse and impactful. In safety and emergency services, these systems offer vital tools for fall detection and high-risk activity monitoring, providing peace of mind and improving safety for individuals in vulnerable situations.

Additionally, they play a crucial role in behavioural research and user experience optimization, offering valuable insights into human behaviour and enabling the development of more effective and engaging technologies. Enterprise solutions benefit from these systems through enhanced productivity and streamlined customer support, while educational tools leverage activity data for interactive learning experiences and effective classroom management. Despite facing challenges such as sensor data accuracy, privacy concerns, and computational demands, the advantages of mobile phone activity classification systems are substantial. They enable the creation of innovative solutions that improve our daily lives, advance research, and support a wide range of professional applications.

9.FUTURE SCOPE

The future of mobile phone activity classification systems is rich with potential, driven by advances in technology and evolving user needs. As these systems continue to develop, several key areas of growth and innovation are on the horizon

1. **Deep Learning Models:** Future advancements will likely involve the integration of more sophisticated deep learning models that can handle complex activity patterns and improve classification accuracy.

2. **Real-Time Activity Recognition:** Future systems will aim to provide real-time activity recognition with minimal latency, making them more effective for dynamic applications

3. **Expanded Application Areas**: Expanding the use of activity classification systems for comprehensive health monitoring and medical diagnosis.

4. **Advanced Privacy Protection Techniques:** Implementing advanced privacy protection techniques to address concerns about data security and user consent.

10.APPENDIX

Model building:

1)Dataset

2)Jupyter Notebook and VS code Application Building

1. HTML file (home file, Index file, Predict file)
2. Models in pickle format

SOURCE CODE:

Home.html

```
<!DOCTYPE html>
<html lang="end">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Frappe Activity</title>
   <link rel="stylesheet" href="styles.css">
<style>
   . body {
      overflow: hidden;
 }
header {
   background-color: #2F3C52; color: white;
padding: 1rem 0;
}
nav ul {
   list-style-type: none;   padding: 14px 16px;
   text-align: right;
```

```
}
nav ul li {
   display: inline-block; justify-content: right;
   margin: 0 1rem; text-align: right;
}
nav ul li a {
   color: white; text-decoration: none;
}
. content {
   padding-top: 205px;
   padding-left: 60px; background-color: #2F3C52;
}
. content >p1{
   font-weight: bold; font-size: 40px;
   margin: 10px;}
. content p2{
   padding-top: 180px;}
. button {
   margin: 40px; background-color: cornflower blue; border-radius: 8px;
   padding: 4px; margin-left: 90px;
}
.img {
   margin-left: 800px; margin-top: -350px;
}
</style>
<body>
   <header>
      <nav>
         <ul>
            <li style="float: left; font-size: 30px;">FRAPPE ACTIVITY</li>
            <li><a href="/home">Home</a></li>
            <li><a href="/about">About</a></li>
            <li><a href="/contact">Contact</a></li>
            <li style=padding:5px ;><a href="/predict">Classify Activity</a></li>
```

```
        </ul>
      </nav>
    </header>
     <div class=" content">
      <p1>Classify mobile activities <br> with the Mobile Frappe<br>app</p1><br>
      <p2>Focused on homework activity classification, the project<br>employs predictive algorithms to discern patterns in user<br>behavior. </p2> <br>
      <a href="/predict">
      <button type="button" class="button">Classify Activity</button>
      </a>
      <div class="img">
        <img src="https://arramton.com/_next/image?url=%2Fassets%2Fimages%2FGroup%2016324.webp&w=640&q=75">
    </div>
    </body>
</html>
```

Predict.Html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Frappe Activity</title>
   <link rel="stylesheet" href="styles.css">
   <style>
      body {
         font-family: Arial, sans-serif;   }
         header {
         background-color: #2F3C52; color: white;
         padding: 1rem 0;
      }
      nav ul {
```

```
    list-style-type: none;
    padding: 14px 16px;
    text-align: right;
}
nav ul li {
    display: inline-block; justify-content: right;
    margin: 0 1rem;
}
nav ul li a {
    color: white; text-decoration: none;
}
main {
    padding: 2rem; background-color: #f0f4f8;
}
. logo {
    display: flex; align-items: center;
}
. logo a {
    padding: 5px; font-size: 15px;
    color: dodger blue; text-decoration: none;
}
. classify-activity {
    background-color: white;
    padding: 1rem; max-width: 800px;
    margin: 2rem auto; border-radius: 8px;
    box-shadow: 0 0 10px rgba (0, 0, 0, 0.1);
}
. classify-activity h2 {
    text-align: center;
    margin-bottom: 2rem; color: cornflower blue;
```

```
        }
        . form-group {
           margin-bottom: 1.5rem;
        }
        . form-group label {
           display: block; margin-bottom: 0.5rem;
        }
        . form-group select {
           width: 100%; padding: 0.5rem;
         border: 1px solid #ccc; border-radius: 4px;
        }
        . predict-button {
           width: 100%; padding: 1rem;
           background-color: #2F3C52; color: white;
           border: none; border-radius: 4px;
           font-size: 1rem; cursor: pointer;
        margin-top: 1.5rem;
        }
        . predict-button: hover {
           background-color: #1f2a38;
        }
    </style>
</head>
<body>
    <header>
        <nav>
           <ul>
              <li style="float: left; font-size: 30px;">FRAPPE ACTIVITY</li>
              <li><a href="/home">Home</a></li>
              <li><a href="/about">About</a></li>
```

```
          <li><a href="/contact">Contact</a></li>
          <li><a href="/predict">Classify Activity</a></li>
        </ul>
      </nav>
    </header>
    <main>
      <div class="logo">
        <a href="/home">Home</a>
        <h1 style="margin-left: 50px;">Classify Activity</h1>
      </div><section class="classify-activity">
        <h2>Enter the Details</h2>
        <form action="/prediction page" method="POST">
          <div class="form-group">
            <label for="item">Item:</label>
            <input type="number" id="item" name="item" required><br><br>
            <label for="daytime">Time of Day</label>
            <select name="daytime" id="daytime">
              <option value="">Click to open</option>
              <option value="Morning">morning</option>
              <option value="Afternoon">afternoon</option>
              <option value="Evening">evening</option>
              <option value="Night">night</option>
            </select></div>
          <div class="form-group">
            <label for="weekday">Day of the Week</label>
            <select name="weekday" id="weekday">
              <option value="">Click to open</option>
              <option value="Sunday">sunday</option>
              <option value="Monday">monday</option>
              <option value="Tuesday">tuesday</option>
```

```html
        <option value="Wednesday">wednesday</option>
        <option value="Thursday">thursday</option>
        <option value="Friday">friday</option>
        <option value="Saturday">saturday</option>
    </select></div>
<div class="form-group">
    <label for="cost">Cost</label>
    <select name="cost" id="cost">
        <option value="">Click to open</option>
        <option value="Free">free</option>
        <option value="Paid">paid</option>
    </select></div>
<div class="form-group">
    <label for="weather">Weather</label>
    <select name="weather" id="weather">
        <option value="">Click to open</option>
        <option value="Sunny">sunny</option>
        <option value="Rainy">rainy</option>
        <option value="Cloudy">cloudy</option>
        <option value="Snowy">snowy</option>
    </select>
    <label for="city">city:</label>
    <input type="number" id="city" name="city" required><br><br>
</div>
<div class="form-group">
    <label for="sname">Name</label>
    <select name="sname" id="sname">
        <option value="">Click to open</option>
        <option value="Any.Do To-do & Task List">Any.Do To-do & Task List</option>
    </select>   </div>
```

```
                    <button type="submit" class="predict-button">Predict</button>
            </form>
        </section>
    </main>
</body>
</html>
```

Prediction Page

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Frappe Activity</title>
    <style>
        body {
            overflow: hidden;
            font-family: Arial, sans-serif;     }
        header {
            background-color: #2F3C52; color: white;
            padding: 1rem 0;
        }
        nav ul {
            list-style-type: none; text-align: right;
        }
        nav ul li {
            display: inline-block;
            margin: 0 1rem;
        }
        nav ul li a {
            color: white; text-decoration: none;
```

```
        }
        main {
          padding: 2rem;
        }
. cont p {
          height: 5pc; width: 80pc;
          margin-left: 90px; margin-top: 90px; padding-top: 70px;
          text-align: center; background-color: rgb (211, 216, 216);
          font-weight: bold;
        }
        . below p {
          margin-top: 100px; margin-left: 90px;
          font-weight: bold;
        }
        . below li {
          margin-top: 20px; margin-left: 60px;999
            }
        . below a {
          text-decoration: none;
        }
    </style>
</head>
<body>
    <header>
        <nav>
          <ul>
            <li style="float: left; font-size: 30px;">FRAPPE ACTIVITY</li>
            <li><a href="/home">Home</a></li>
            <li><a href="/about">About</a></li>
            <li><a href="/contact">Contact</a></li>
```

```html
            <li><a href="/predict">Classify Activity</a></li>
          </ul>
        </nav>
    </header>
    <main>
        <div class="logo">
            <a href="#" style="padding: 30px; font-size: 15px; margin-left: 50px; color: dodger blue; text-decoration: none;">Home</a>
            <label for="Classify Activity" style="font-size: 15px;">Classify Activity</label>
        </div>
        <div class="below" style="padding: 15px; margin-left: 70px; font-size: 30px; font-weight: bold; color: cornflowerblue;">
            <label for="Classification Activity">Classification Result</label>
        </div>
    </main>
    <div class="cont">
        <p> {{prediction text}} </p>
    </div>
    <section class="below">
        <p>Useful Links</p>
        <ul>
            <li><a href="/home"> > Home</a></li>
            <li><a href="/about"> > About us</a></li>
        </ul>
    </section>
</body>
</html>
```

CODE SNIPPETS

MODEL BUILDING

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn .preprocessing import StandardScaler
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

from sklearn.model_selection import RandomizedSearchCV
```

```python
import pandas as pd
df = pd.read_csv("frappe.csv")
```

```python
df.head()
```

```python
df.head()
```

	user	item	cnt	daytime	weekday	isweekend	homework	cost	weather	country	city
0	0	0	1	morning	sunday	weekend	unknown	free	sunny	United States	0
1											
2											
3											
4											

```python
meta_app=pd.read_csv("meta.csv",sep='\t')
```

```python
meta_app.head()
```

	item	package	category	downloads	developer	icon	language	description	name	price	rating	short desc
0	0	com.anydo	Productivity	1,000,000 - 5,000,000	Any.DO	http://d2lh3rxs7crswz.cloudfront.net/com.anydo...	en	Meet Any.DO, the best way to-do To-do's on And...	Any.DO To-do & Tasks List	Free	4.5	Any.DO helps you remember everything you have ...
1	1	com.yahoo.mobile.client.android.yahoo	News & Magazines	1,000,000 - 5,000,000	Yahoo! Inc.	http://d2lh3rxs7crswz.cloudfront.net/com.yahoo...	en	With Yahoo! for Android, you'll stay connected...	Yahoo!	Free	4.2	With Yahoo! for Android, you'll stay connected...
2	2	com.compasspro	Tools	1,000,000 - 5,000,000	Mobile Essentials	http://d2lh3rxs7crswz.cloudfront.net/com.compa...	en	Professional Compass for Android. Simple and p...	Compass PRO	Free	4.1	Professional Compass for Android. Simple and p...
3	3	com.instagram.android	Social	50,000,000 - 100,000,000	Instagram	http://d2lh3rxs7crswz.cloudfront.net/com.insta...	en	Instagram – A beautiful way to share your worl...	Instagram	Free	4.6	Instagram – A beautiful way to share your worl...
4	4	com.shoppinglist	Shopping	10,000 - 50,000	Kiwi3	http://d2lh3rxs7crswz.cloudfront.net/com.shopp...	en	Shopping List is the most easy way to organize...	Shopping List	Free	4.5	The most easy Shopping List. In English, for f...

```python
df.info()
```

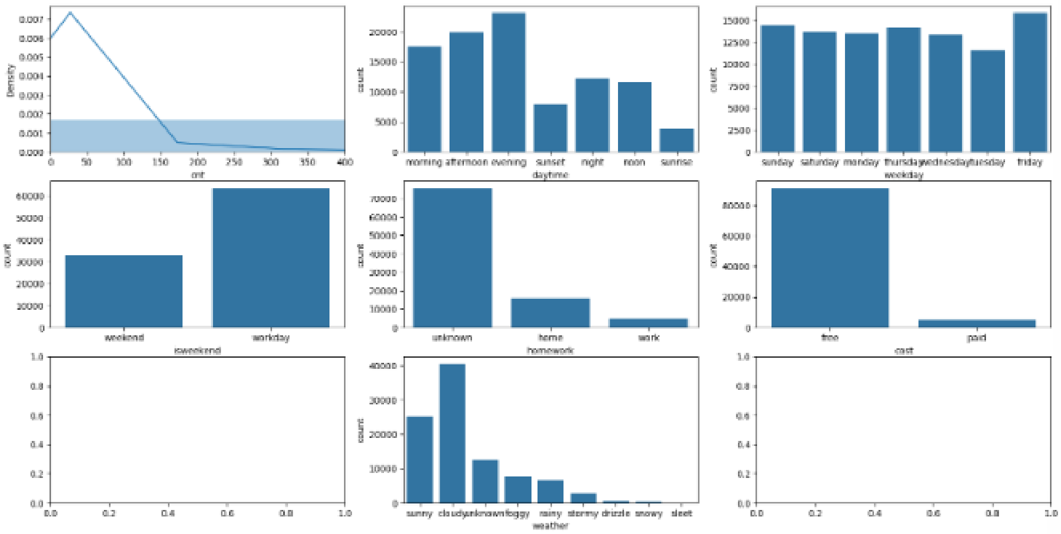
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 0 entries
Data columns (total 12 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   user       0 non-null      int64
 1   item       0 non-null      int64
 2   cnt        0 non-null      int64
 3   daytime    0 non-null      int32
 4   weekday    0 non-null      int32
 5   isweekend  0 non-null      int32
 6   homework   0 non-null      int32
 7   cost       0 non-null      int32
 8   weather    0 non-null      int32
 9   country    0 non-null      object
 10  city       0 non-null      int64
 11  name       0 non-null      int32
dtypes: int32(7), int64(4), object(1)
memory usage: 132.0+ bytes
```

```
df.isna().sum()
```

```
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
```

```
from sklearn.preprocessing import LabelEncoder
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()
```

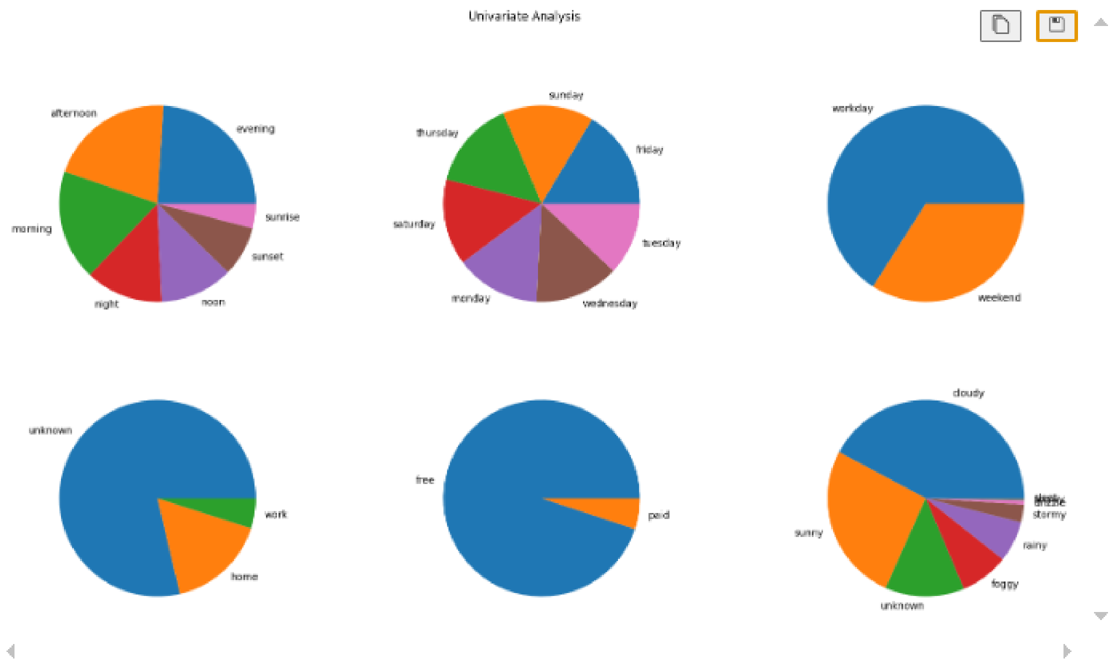
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv("frappe.csv")
df = df['user\titem\tcnt\tdaytime\tweekday\tisweekend\thomework\tcost\tweather\tcountry\tcity'].str.split('\t', expand=True)
df.columns = ['user', 'item', 'cnt', 'daytime', 'weekday', 'isweekend', 'homework', 'cost', 'weather', 'country', 'city']
fig,axes=plt.subplots(3,3,figsize=(20,10))
fig.suptitle("univariate Analysis")
sns.distplot(df['cnt'],ax=axes[0,0])
axes[0,0].set_xlim(0,400)
sns.countplot(x=df['daytime'],ax=axes[0,1])
#df = df['user\titem\tcnt\tdaytime\tweekday\tisweekend\thomework\tcost\tweather\tcountry\tcity'].str.split('\t', expand=True)
#df.columns = ['user', 'item', 'cnt', 'daytime', 'weekday', 'isweekend', 'homework', 'cost', 'weather', 'country', 'city']
sns.countplot(x=df['weekday'],ax=axes[0,2])
sns.countplot(x=df['isweekend'],ax=axes[1,0])
sns.countplot(x=df['homework'],ax=axes[1,1])
sns.countplot(x=df['cost'],ax=axes[1,2])
sns.countplot(x=df['weather'],ax=axes[2,1])
```




```
fig,axes=plt.subplots(2,3,figsize=(20,10))
fig.suptitle("Univariate Analysis")
col=['daytime','weekday','isweekend','homework','cost','weather']
k=0
for i in range(2):
  for j in range(3):
    d=dict(df[col[k]].value_counts())
    axes[i,j].pie(d.values(),labels=d.keys())
    k+=1
```

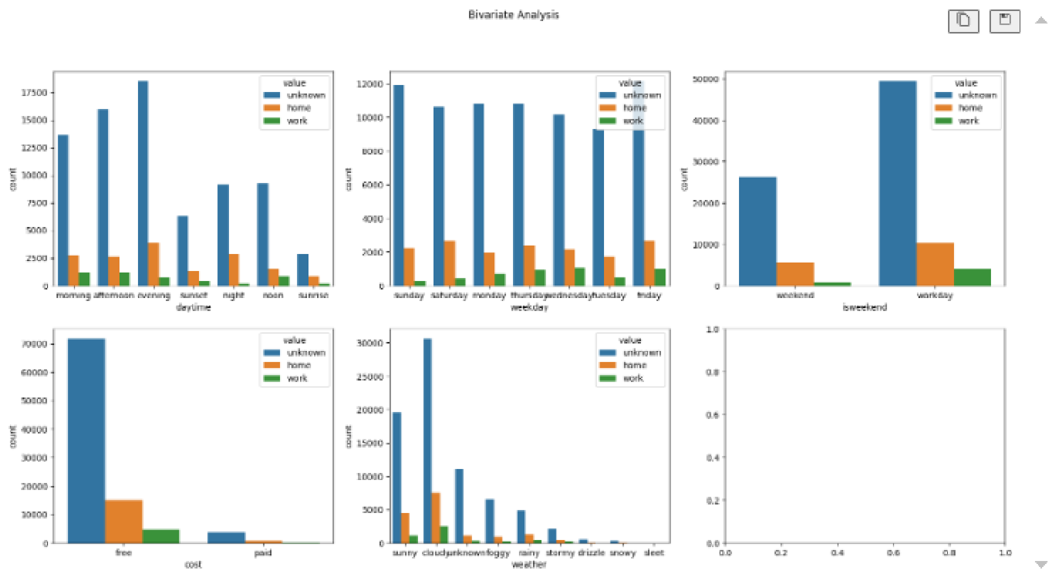
[28]

Python



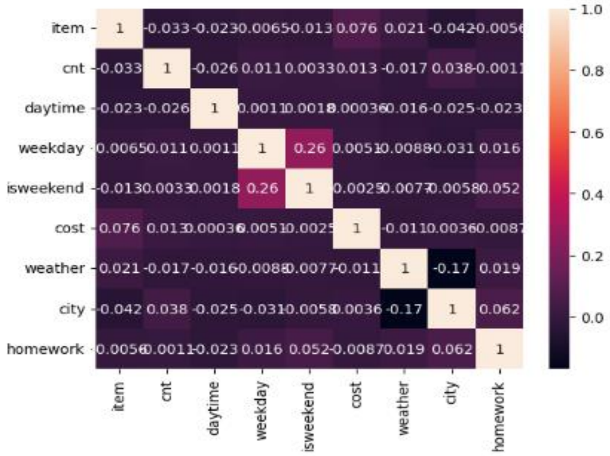

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

fig,axes=plt.subplots(2,3,figsize=(20,10))
fig.suptitle("Bivariate Analysis")

# Melt the dataframe to long-form
df_melted = pd.melt(df, id_vars=['daytime', 'weekday', 'isweekend','cost','weather'], value_vars=['homework'])

# Use the melted dataframe in the countplots
sns.countplot(x='daytime', hue='value', data=df_melted, ax=axes[0,0])
sns.countplot(x='weekday', hue='value', data=df_melted, ax=axes[0,1])
sns.countplot(x='isweekend', hue='value', data=df_melted, ax=axes[0,2])
sns.countplot(x='cost' , hue='value', data=df_melted, ax=axes[1,0])
sns.countplot(x='weather', hue='value',data=df_melted,ax=axes[1,1])
```

[29]



```
sns.heatmap(c,annot=True)
```

```
<Axes: >
```



```
from imblearn.over_sampling import SMOTE
smote_sampler = SMOTE(random_state=42)
X_smote, y_smote = smote_sampler.fit_resample(x, y)
smote_data = pd.concat([X_smote, y_smote], axis=1)
smote_data.shape
```

```
(227010, 8)
```

```
smote_data['homework'].value_counts()
```

```
homework
1    75670
0    75670
2    75670
Name: count, dtype: int64
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X_smote,y_smote,train_size=0.8,random_state=42)
```

```
x_test.shape
```

```
(45402, 7)
```

```
y_test.shape
```

```
(45402, 1)
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler() # Use the correct class name with capitalization
scaler.fit(x_train)
x_train = scaler.transform(x_train)
x_test = scaler.transform(x_test)
```

```
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
```

```
models=[]
models.append(('KNeighborsClassifier',KNeighborsClassifier()))
models.append((' DecisionTreeClassifier', DecisionTreeClassifier()))
models.append(('RandomForestClassifier',RandomForestClassifier()))
models.append(('XGBClassifier',XGBClassifier()))
models.append(('BaggingClassifier',BaggingClassifier()))
models.append(('AdaBoostClassifier',AdaBoostClassifier()))
```

```
model_perform={}
```

```
for   name ,model in models:
 from sklearn.metrics import precision_score
 from sklearn.metrics import recall_score
 from sklearn.metrics import accuracy_score
 from sklearn.metrics import f1_score
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
p=precision_score(y_test,y_pred,average='micro')
r=recall_score(y_test,y_pred,average='micro')
a=accuracy_score(y_test,y_pred)
f=f1_score(y_test,y_pred,average='micro')
s={'precision':p,'Recall':r,'Accuracy':a,'F1 Score':f}
model_perform[name]=s
```

```python
for model in model_perform:
    print(model)
    print("precision:",model_perform[model]['precision'])
print("Recall:",model_perform[model]["Recall"])
print("Accuracy:",model_perform[model]["Accuracy"])
print("F1 Score:",model_perform[model]["F1 Score"])
```

```
AdaBoostClassifier
precision: 0.5588079820272235
Recall: 0.5588079820272235
Accuracy: 0.5588079820272235
F1 Score: 0.5588079820272235
```

```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import RandomizedSearchCV

# Create a decision tree classifier object
dt_classifier = DecisionTreeClassifier()

random_search = RandomizedSearchCV(estimator=dt_classifier,
                                   param_distributions=param_grid,
                                   scoring='accuracy',
                                   cv=3,
                                   n_iter=100,
                                   random_state=42)
```

```python
random_search.fit(x_train,y_train)
print("Best Parameters:",random_search.best_params_)
print("Best Score:",random_search.best_score_)
```

```
Best Parameters: {'splitter': 'best', 'min_samples_split': 2, 'min_samples_leaf': 2, 'min_impurity_decrease': 0.0, 'max_features': None, 'max_depth': None, 'criterion': 'entropy', 'ccp_alpha': 0.0}
Best Score: 0.7997445046473723
```

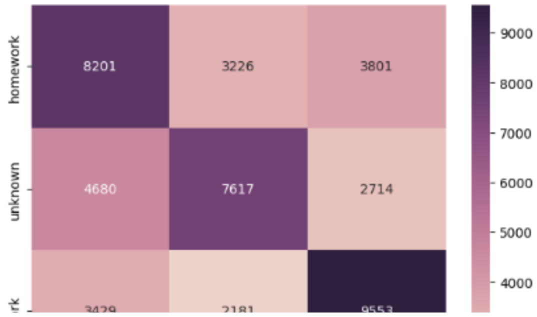
```python
dt_final=random_search
```

```python
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred, digits=4))
```

```
              precision    recall  f1-score   support

           0     0.5028    0.5385    0.5201     15228
           1     0.5848    0.5074    0.5434     15011
           2     0.5945    0.6300    0.6118     15163

    accuracy                         0.5588     45402
   macro avg     0.5607    0.5587    0.5584     45402
weighted avg     0.5606    0.5588    0.5584     45402
```

```python
from sklearn.metrics import classification_report, confusion_matrix # Import both function

y_pres=dt_final.predict(x_test)
print(dt_final.score(x_test, y_test))
cm = confusion_matrix(y_test, y_pred, labels=[0, 1, 2])
df_cm = pd.DataFrame(cm, index=[i for i in ["homework", "unknown", "work"]],
                     columns=[i for i in ["homework", "unknown", "work"]])
plt.figure(figsize=(7, 5))
sns.heatmap(df_cm,annot=True,fmt='g',cmap=sns.cubehelix_palette(as_cmap=True))
```

```
0.8193251398616801

<Axes: >
```




```python
dt_classifier = DecisionTreeClassifier()

# Define the parameter grid
param_grid = {
    'criterion': ['gini', 'entropy'],
    'splitter': ['best', 'random'],
    'max_depth': [None, 2, 4, 6, 8, 10],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': [None, 'sqrt', 'log2'],
    'min_impurity_decrease': [0.0, 0.1, 0.2],
    'ccp_alpha': [0.0, 0.1, 0.2]
}

# Create the RandomizedSearchCV object with 3 folds for cross-validation
random_search = RandomizedSearchCV(estimator=dt_classifier,
                                   param_distributions=param_grid,
                                   scoring='accuracy',
                                   cv=3,  # Specify the number of folds here
                                   random_state=42)

# Fit the RandomizedSearchCV object with your training data
random_search.fit(x_train, y_train)
```

```python
print("Best Parameters:",random_search.best_params_)
print("Best Score:",random_search.best_score_)
```

```
Best Parameters: {'splitter': 'best', 'min_samples_split': 5, 'min_samples_leaf': 4, 'min_impurity_decrease': 0.2, 'max_features': None, 'max_depth': 10, 'criterion': 'gini', 'ccp_alpha': 0.0}
Best Score: 0.33401061627241085
```

```python
print("Best Score:",random_search.score(x_test,y_test))
```

```
Best Score: 0.33062420157702305
```

```python
bc_final=random_search
```

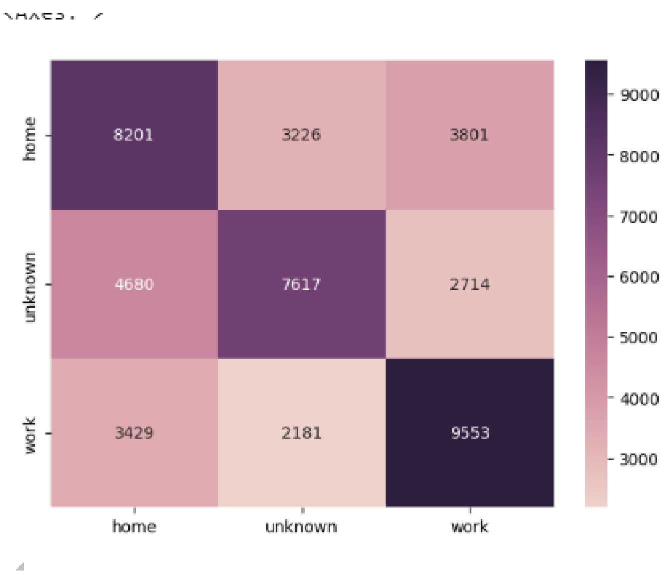
```python
print(classification_report(y_test,y_pred,digits=4))
```



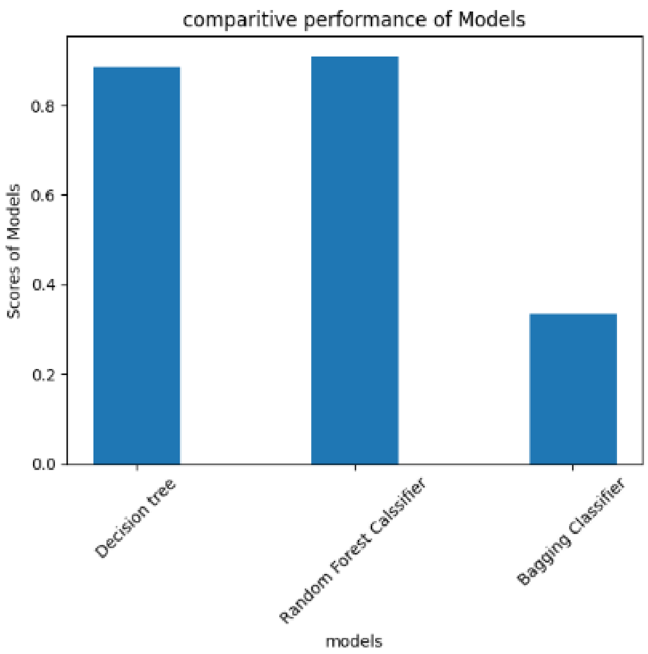
```
              precision    recall  f1-score   support

           0     0.5028    0.5385    0.5201     15228
           1     0.5848    0.5074    0.5434     15011
           2     0.5945    0.6300    0.6118     15163
```

```python
from sklearn.metrics import classification_report, confusion_matrix # Import both functions

y_pres=dt_final.predict(x_test)
print(dt_final.score(x_test, y_test))
cm = confusion_matrix(y_test, y_pred, labels=[0, 1, 2])
df_cm = pd.DataFrame(cm, index=[i for i in ["home", "unknown", "work"]],
                     columns=[i for i in ["home", "unknown", "work"]])
plt.figure(figsize=(7, 5))
sns.heatmap(df_cm,annot=True,fmt='g',cmap=sns.cubehelix_palette(as_cmap=True))
```

```
0.8198537509360821
```




```python
train_score={'Decision tree':dt_final.score(x_train,y_train),'Random Forest Calssifier':rf_final.score(x_train,y_train),'Bagging Classifier':bc_final.best_score_}
models=list(train_score.keys())
scores=list(train_score.values())
plt.bar(models,scores,
    width=0.4)

plt.xlabel("models")
plt.ylabel("Scores of Models")
plt.xticks(rotation=45)
plt.title("comparitive performance of Models")
plt.show()
```




```
print(train_score)
print(test_score)
```

```
{'Decision tree': 0.8862329853310427, 'Random Forest Calssifier': 0.9086108541473944, 'Bagging Classifier': np.float64(0.33401061627241085)}
{'Decision Tree': 0.8198537509360821, 'Random Forest Classifier': 0.8208669221620193, 'Bagging Classifier': 0.33062420157702305}
```

```
import pickle
```

```
from sklearn.tree import DecisionTreeClassifier

pickle.dump(bc_final,open("model.pkl","wb"))
```

```
pickle.dump(scaler,open('scaler.pkl',"wb"))
```

```
import joblib
joblib.dump(dt_encoder,"DayTimeEncoder")
joblib.dump(wd_encoder,"weekdayEncoder")
joblib.dump(wknd_encoder,"wkndEncoder")
joblib.dump(hw_encoder,"HWEncoder")
joblib.dump(w_encoder,"WeatherEncoder")
joblib.dump(c_encoder,"CostEncoder")
joblib.dump(n_encoder,"NameEncoder")
```

```
['NameEncoder']
```

```python
import os
os.getcwd()
```

```
'c:\\Users\\mandh\\OneDrive\\Documents\\MobileActivity_MiniProject - Copy'
```

```python
dt_encoder
```

```
▾ LabelEncoder
LabelEncoder()
```