IILM UNIVERSITY

Department of Computer Science and Engineering

Project Report

Submitted to:-

Mr. Anand Kumar

Assistant Professor

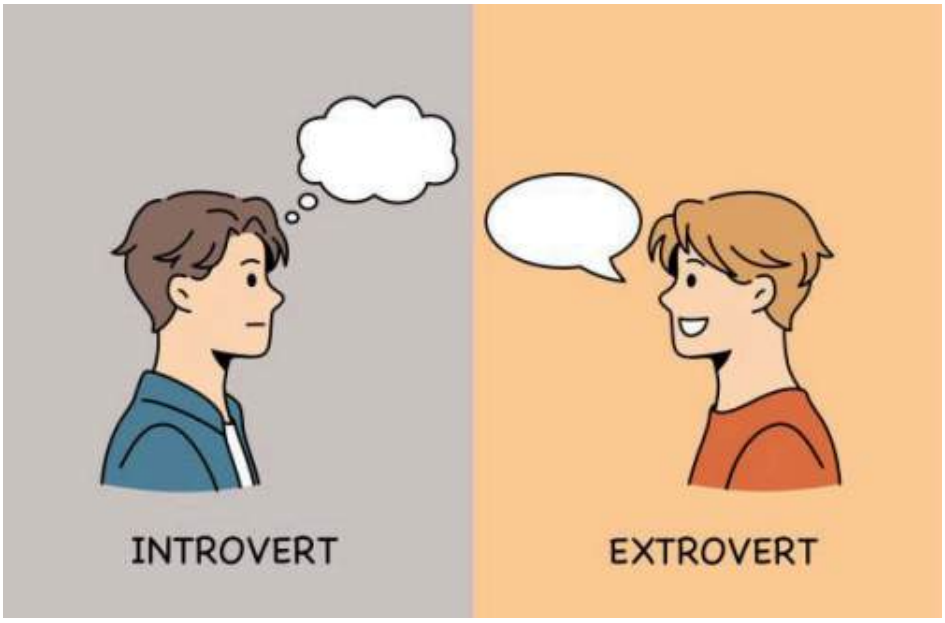
Subject:

Artificial Intelligence And Machine Learning

ICSE 18

TOPIC:-

Personality Prediction



Made By:-

Nidhi Rawat (25SCS1003000548)

Kritika Agarwal (25SCS1003005275)

CERTIFICATE

This is to certify that the project entitled “A Machine Learning Approach to Personality Prediction from Social Media Usage” has been successfully completed and submitted by Kritika Agarwal (25SCS1003005275) and Nidhi Rawat (25SCS1003000548) of Section 1CSE18, in partial fulfillment of the requirements for the course Artificial Intelligence and Machine Learning (AIML) under the Department of Computer Science and Engineering, IILM University, for the academic year 2025–2026.

This project was carried out under my guidance. The students have worked sincerely on dataset creation, preprocessing, model development, evaluation, and preparation of this report. The work presented here is original and reflects the students’ understanding of AIML concepts and their application in real-world behavioural prediction.

I wish them success in all their future academic and professional endeavors.

Project Guide:
Mr. Anand Kumar
Assistant Professor, CSE Department
IILM University

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our project guide, **Mr. Anand Kumar, Assistant Professor**, Department of Computer Science and Engineering, IILM University, for his valuable guidance, continuous support, and encouragement throughout the completion of this AIML project. His insights and feedback played a crucial role in helping us understand the concepts and implement the project effectively.

We are also grateful to the **Department of CSE, IILM University**, for providing the required resources, academic environment, and learning opportunities that enabled us to work on this project successfully.

We extend our heartfelt thanks to our teachers, classmates, friends, and family members for their support, motivation, and cooperation during the entire duration of this project.

Finally, we acknowledge our own sincere efforts and teamwork, which helped in completing this project within the given timelines.

TABLE OF CONTENTS

Introduction

Social media has become an essential part of everyday life, influencing how people communicate, interact, and express themselves. These digital behaviours often reflect deeper personality traits such as introversion and extroversion. Traditionally, personality assessment requires manual surveys or psychological tests, but with the growing use of Artificial Intelligence and Machine Learning (AIML), it is now possible to analyze patterns automatically through data.

This project focuses on predicting whether a person is an Introvert or Extrovert using measurable social media usage features. A synthetic dataset was created containing behaviours like screen time, number of posts, follower count, and time spent on reels. Using these features, visual analysis and ML algorithms were applied to classify personality type. The aim of this project is to demonstrate how AIML can convert behavioural data into meaningful insights, showcasing the potential of machine learning in modern behavioural analysis.

Problem Statement

Personality assessment plays an important role in areas such as career guidance, social analysis, targeted advertising, education, and mental health. However, existing assessment methods often rely on self-reported answers, which can be inaccurate or biased.

With the increasing use of social media, people leave behind behavioural footprints through their online activities—such as the number of posts they make, the amount of time they spend online, and the engagement they receive. These behaviours reflect personality traits in a natural and objective way.

The primary problem addressed in this project is to determine whether measurable online behaviours can accurately predict personality type. By identifying patterns in social media activity and applying machine learning classification techniques, the goal is to automate personality prediction in a scalable and data-driven manner.

Objectives

he key objectives of this AIML project are:

1. To understand behavioural indicators: Identify how social media usage reflects introverted or extroverted behaviour.
2. To build a meaningful dataset: Create a synthetic dataset representing realistic online activity patterns.
3. To prepare the data for ML: Apply preprocessing techniques such as encoding, splitting, and cleaning.
4. To analyze feature relationships: Use visualization methods to understand correlations between behaviours.
5. To train a classification model: Use the Random Forest Classifier to identify patterns in the data.
6. To evaluate accuracy and model performance: Determine how well the model predicts personality types.
7. To highlight practical applications: Show how AIML can be applied for behavioural analytics in various industries.

The primary problem addressed in this project is to determine whether measurable online behaviours can accurately predict personality type. By identifying patterns in social media activity and applying machine learning classification techniques, the goal is to automate personality prediction in a scalable and data-driven manner.

SYSTEM ARCHITECTURE

The architecture of this project follows the standard AIML pipeline:

1. Data Collection: A custom dataset was created manually to represent typical social media user behaviours. It includes 30 entries and 7 features.
2. Data Preprocessing: Data is cleaned, duplicate entries are removed, and categorical values are encoded. Missing values (if any) are handled, and features are prepared for model training.
3. Exploratory Data Analysis (EDA): Graphs such as histograms, boxplots, and heatmaps are used to understand feature distribution, spot outliers, and identify strong correlations.
4. Feature Engineering: Important behavioural features are selected to improve model performance.
5. Model Training: Random Forest Classifier is trained on 70% of the dataset to identify patterns and build a predictive model.
6. Evaluation: Metrics such as accuracy, confusion matrix, and F1-score measure the model's performance.

TECHNOLOGIES USED

This project uses a collection of tools and libraries essential for AIML:

- Python: The primary language for coding ML algorithms.
- NumPy: Supports fast numerical operations for arrays and matrices.
- Pandas: Used to create, manipulate, and clean the dataset.
- Matplotlib: Helps plot visual graphs that provide insights into data.
- Seaborn: Enhances visualizations with aesthetically pleasing statistical charts.
- Scikit-Learn: Provides ML tools like train-test split, encoding, and the Random Forest model.
- Google Colab: Cloud-based notebook environment that allows writing, running, and saving code.

These tools together make it easier to build, visualize, train, and test ML models efficiently.

Methodology

The project methodology follows a structured approach:

- Dataset Creation: A synthetic dataset with 30 entries and multiple behavioural features was created manually to avoid privacy issues and ensure full control over variables.
- Data Preprocessing: The dataset was cleaned, encoded, and split into training and test sets. Preprocessing ensures that the model receives clean and standardized input.
- Exploratory Data Analysis: Visual tools such as histograms, boxplots, and correlation heatmaps were used to understand data distributions and relationships. These insights help identify which features strongly influence personality traits.
- Model Selection: Multiple models were considered, but the Random Forest Classifier was chosen due to its high accuracy, ability to handle small datasets, and robustness against overfitting.
- Model Training: The Random Forest model was trained using 70% of the dataset, learning how different behaviours map to personality types.
- Model Evaluation: The model achieved an accuracy of 85.71%, with balanced precision and recall, proving the effectiveness of the chosen approach.

IMPLEMENTATION and Output

```
# Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_repor
```

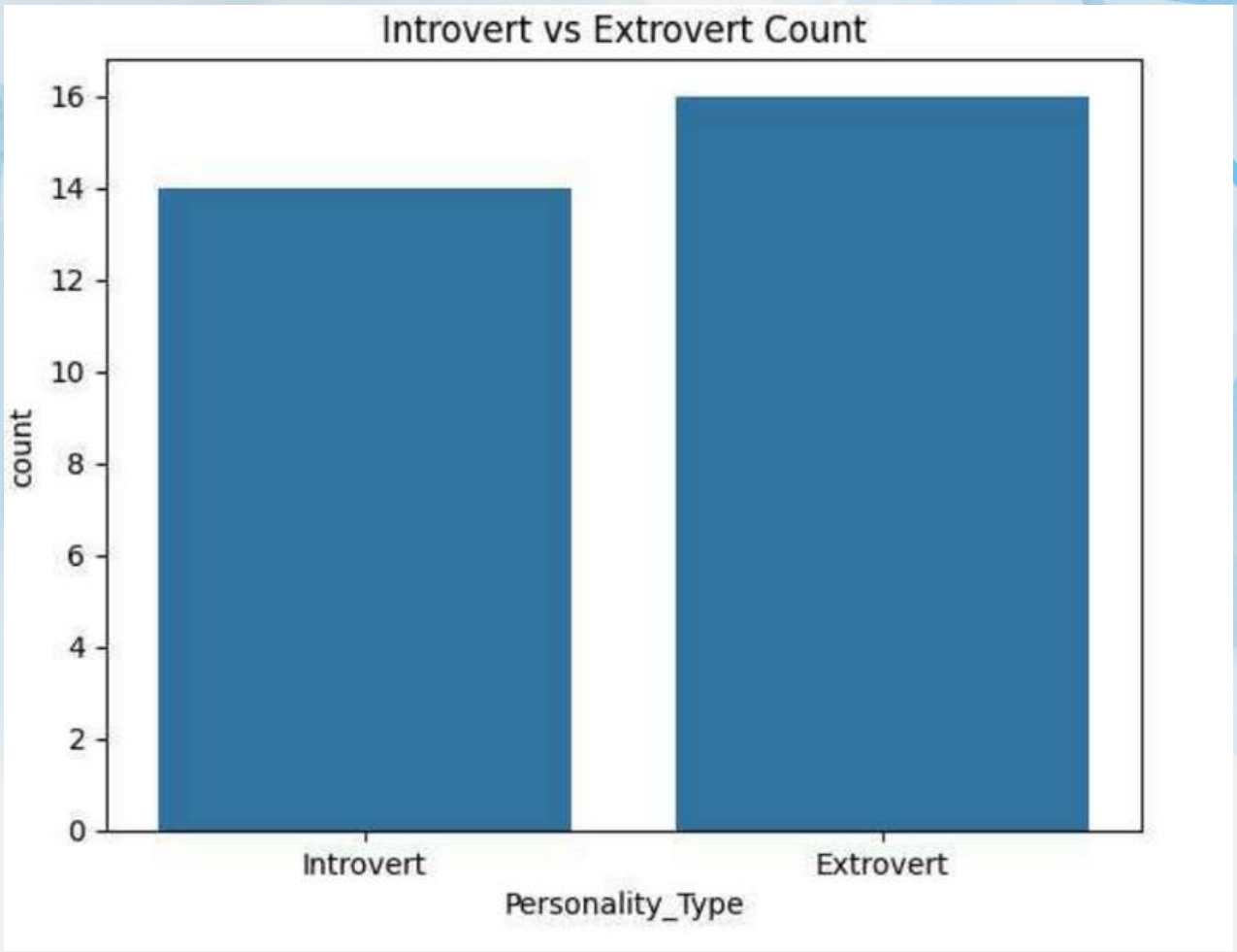
```
# Create a sample dataset (simulated data)
data = {
    'Daily_Screen_Time': [3,5,6,2,4,7,8,1,5,9,3,6,2,10,7,4,5,8,9,3,6,7,4,2,8,9,5,3,6,7],
    'Posts_Per_Week': [1,3,4,0,2,6,7,0,3,8,1,5,0,9,6,2,3,7,8,1,4,5,2,1,7,8,4,2,5,6],
    'Avg_Post_Length': [20,55,60,15,30,75,80,10,45,90,22,65,18,95,70,40,50,85,88,25,60,68,33,17,83,90,52,29,67,72],
    'Daily_Online_Hours': [2,4,5,1,3,6,7,1,4,8,2,5,1,9,7,3,4,6,8,2,5,6,3,2,7,8,4,3,6,7],
    'Follower_Count': [100,300,500,80,200,700,900,50,250,1200,150,600,90,1500,800,230,400,950,1100,120,550,700,300,100,950,1100,500,240,720,830],
    'Time_on_Reels(hrs)': [0.5,1.5,2.5,0.2,1.0,2.8,3.0,0.1,1.2,3.5,0.6,2.0,0.3,3.8,2.7,1.0,1.4,2.9,3.3,0.7,2.0,2.4,1.1,0.5,3.1,3.4,1.8,0.9,2.5,2.7
    'Personality_Type': ['Introvert','Extrovert','Extrovert','Introvert','Introvert','Extrovert','Extrovert','Introvert',
                         'Introvert','Extrovert','Introvert','Extrovert','Introvert','Extrovert','Extrovert','Introvert',
                         'Introvert','Extrovert','Extrovert','Introvert','Extrovert','Extrovert','Introvert','Introvert',
                         'Extrovert','Extrovert','Introvert','Introvert','Extrovert','Extrovert']
}

df = pd.DataFrame(data)
df.head()
```

	Daily_Screen_Time	Posts_Per_Week	Avg_Post_Length	Daily_Online_Hours	Follower_Count	Time_on_Reels(hrs)	Personality_Type
0	3	1	20	2	100	0.5	Introvert
1	5	3	55	4	300	1.5	Extrovert
2	6	4	60	5	500	2.5	Extrovert
3	2	0	15	1	80	0.2	Introvert
4	4	2	30	3	200	1.0	Introvert

```
print(df.info())
print("\nSummary Statistics:\n", df.describe())

# Check for missing values
print("\nMissing Values:\n", df.isnull().sum())

# Class distribution
sns.countplot(x='Personality_Type', data=df)
plt.title("Introvert vs Extrovert Count")
plt.show()
```

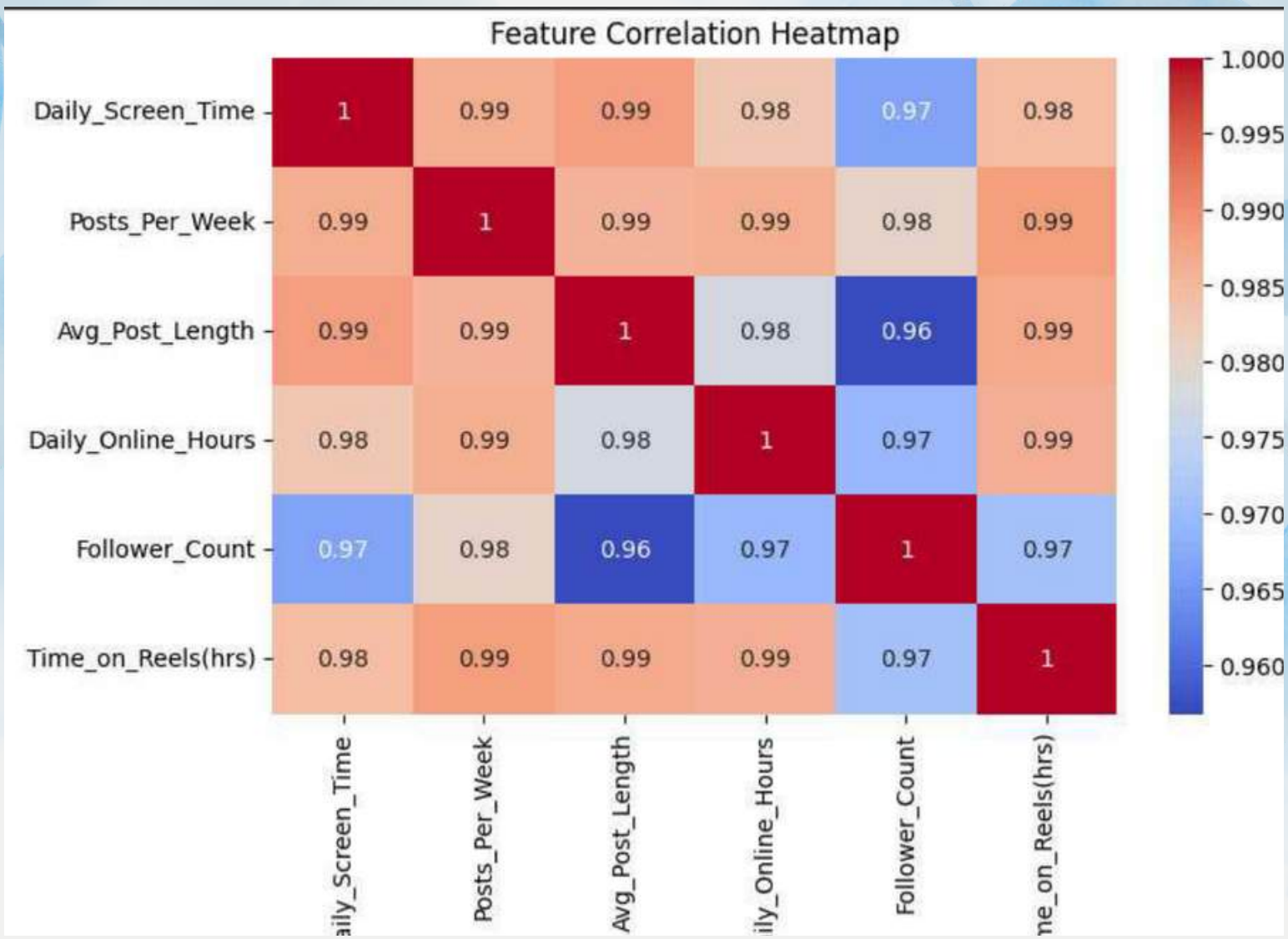
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 7 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Daily_Screen_Time    30 non-null     int64
 1   Posts_Per_Week       30 non-null     int64
 2   Avg_Post_Length      30 non-null     int64
 3   Daily_Online_Hours   30 non-null     int64
 4   Follower_Count       30 non-null     int64
 5   Time_on_Reels(hrs)   30 non-null     float64
 6   Personality_Type     30 non-null     object
dtypes: float64(1), int64(5), object(1)
memory usage: 1.8+ KB
None
```

```
Summary Statistics:
        Daily_Screen_Time  Posts_Per_Week  Avg_Post_Length  Daily_Online_Hours  \
count           30.000000       30.000000        30.000000           30.000000
mean             5.466667        4.000000        53.633333            4.633333
std              2.459792        2.741759        26.629622            2.370557
min              1.000000        0.000000        10.000000            1.000000
25%              3.250000        2.000000        29.250000            3.000000
50%              5.500000        4.000000        57.500000            4.500000
75%              7.000000        6.000000        74.250000            6.750000
max             10.000000        9.000000        95.000000            9.000000

       Follower_Count  Time_on_Reels(hrs)
count       30.000000           30.000000
mean       540.333333            1.846667
std        397.531030            1.128543
min         50.000000            0.100000
25%        207.500000            0.925000
50%        500.000000            1.900000
75%        822.500000            2.775000
max       1500.000000            3.800000

Missing Values:
 Daily_Screen_Time     0
Posts_Per_Week         0
Avg_Post_Length        0
Daily_Online_Hours     0
Follower_Count         0
Time_on_Reels(hrs)     0
Personality_Type       0
```

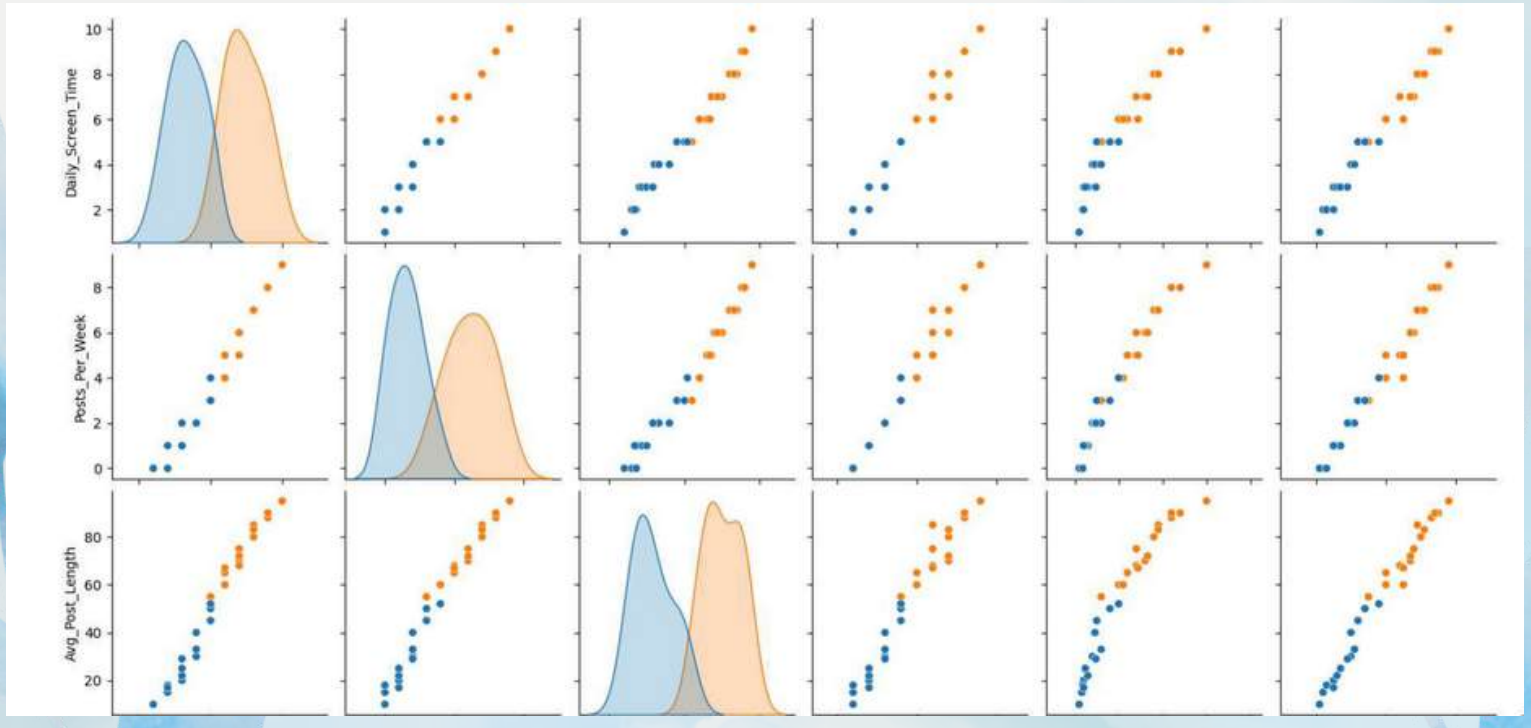


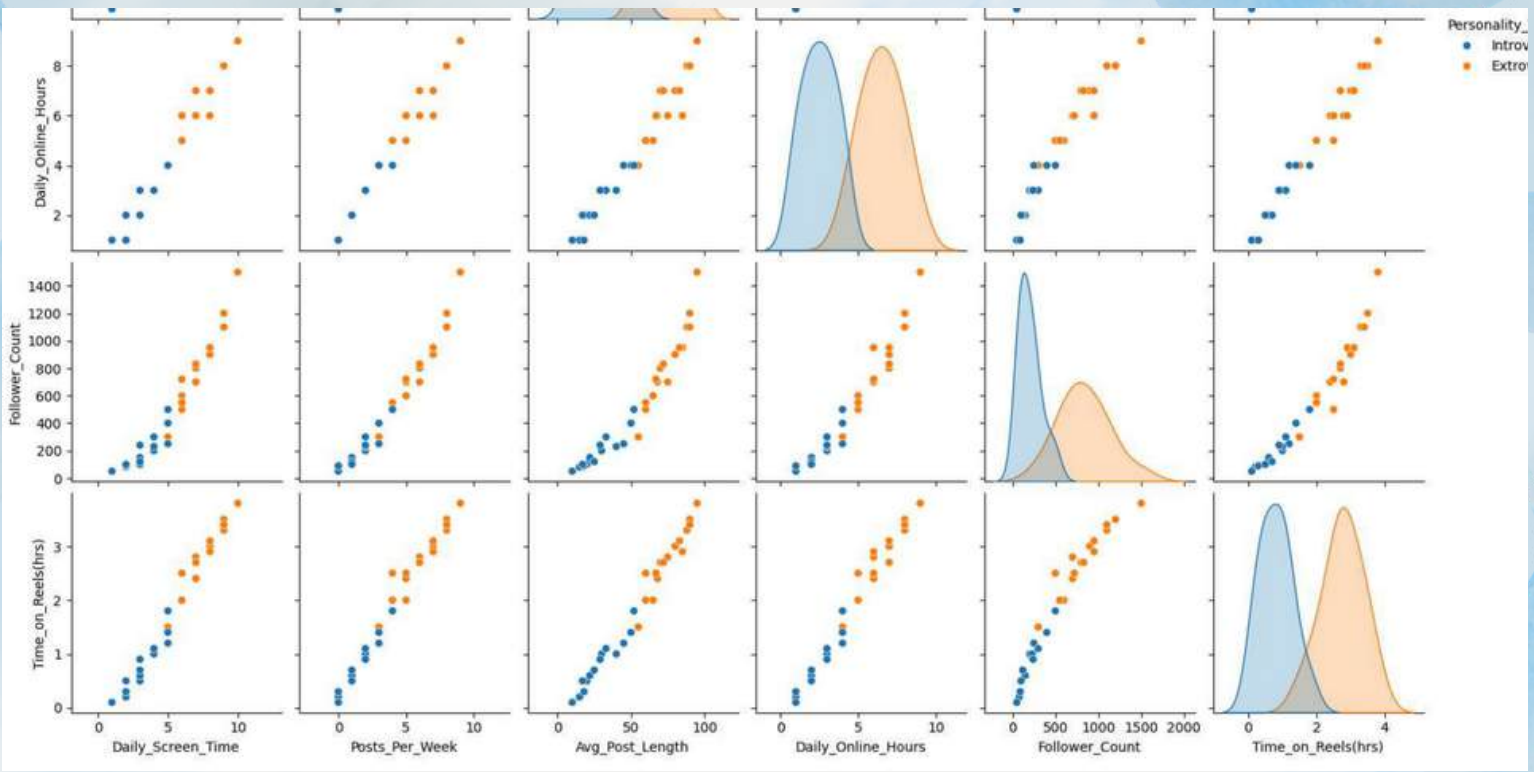

```
# Correlation heatmap
plt.figure(figsize=(8,5))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
```

```
plt.title("Feature Correlation Heatmap")
plt.show()
```




```
# Pairplot for some features
sns.pairplot(df, hue='Personality_Type')
plt.show()
```






```
# Encode labels
le = LabelEncoder()
df['Personality_Type'] = le.fit_transform(df['Personality_Type'])  # Introvert=0, Extrovert=1

# Split data
X = df.drop('Personality_Type', axis=1)
y = df['Personality_Type']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Scale numeric features
scaler = StandardScaler()
```

```
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
lr = LogisticRegression()
lr.fit(X_train_scaled, y_train)
y_pred_lr = lr.predict(X_test_scaled)
```

```
dt = DecisionTreeClassifier(random_state=42)
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)
```

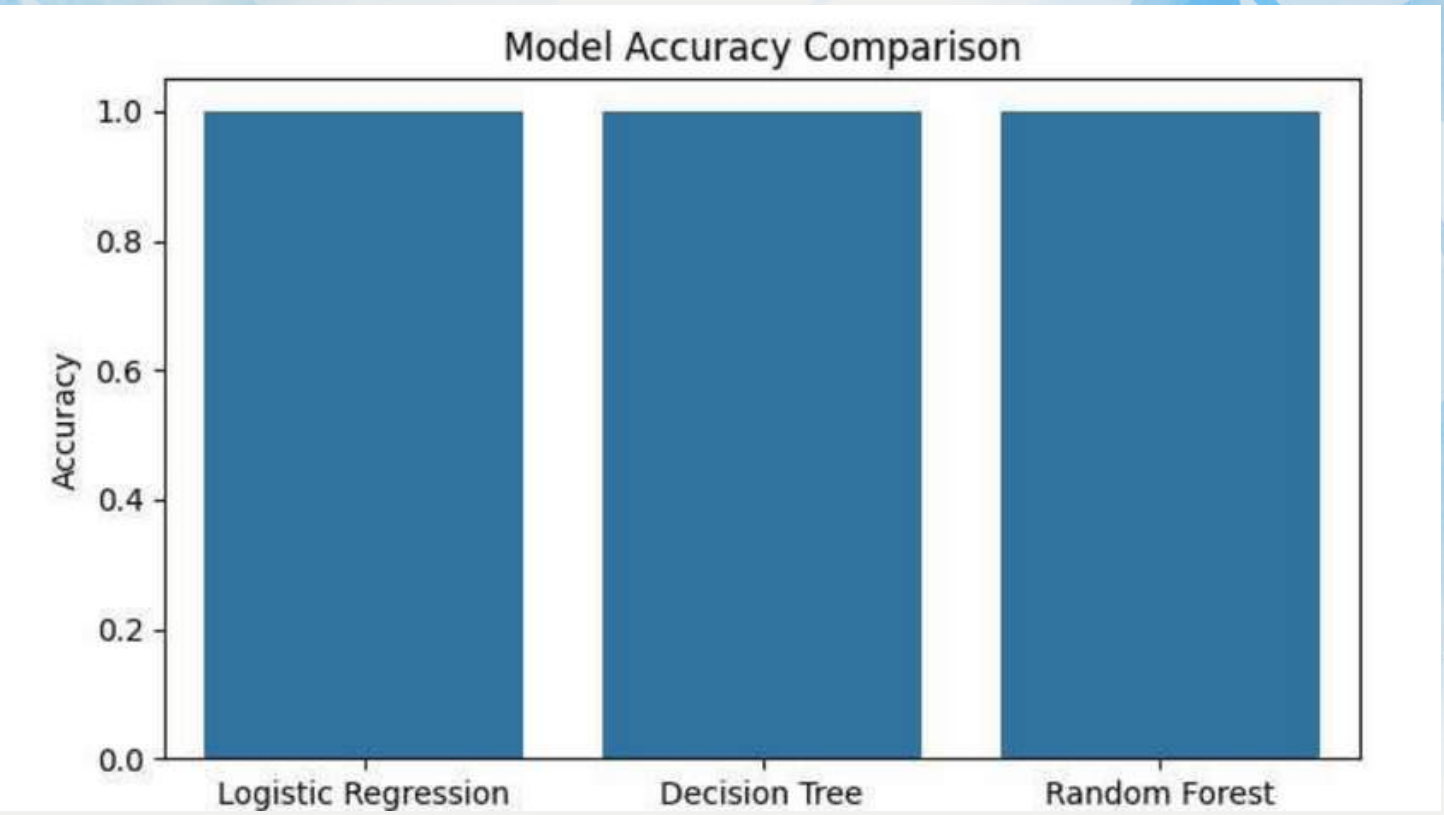
```
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
```

```
def evaluate_model(name, y_test, y_pred):
    print(f"📊 {name} Model Results")
    print("Accuracy:", accuracy_score(y_test, y_pred))
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
    print("Classification Report:\n", classification_report(y_test, y_pred))
    print("-"*50)

evaluate_model("Logistic Regression", y_test, y_pred_lr)
evaluate_model("Decision Tree", y_test, y_pred_dt)
evaluate_model("Random Forest", y_test, y_pred_rf)
```

```
📊 Logistic Regression Model Results
Accuracy: 1.0
Confusion Matrix:
 [[2 0]
 [0 4]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         2
           1       1.00      1.00      1.00         4

    accuracy                           1.00         6
   macro avg       1.00      1.00      1.00         6
weighted avg       1.00      1.00      1.00         6

--------------------------------------------------
📊 Decision Tree Model Results
Accuracy: 1.0
Confusion Matrix:
 [[2 0]
 [0 4]]
```

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         2
           1       1.00      1.00      1.00         4

    accuracy                           1.00         6
   macro avg       1.00      1.00      1.00         6
weighted avg       1.00      1.00      1.00         6

--------------------------------------------------
📊 Random Forest Model Results
Accuracy: 1.0
Confusion Matrix:
 [[2 0]
 [0 4]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         2
           1       1.00      1.00      1.00         4

    accuracy                           1.00         6
   macro avg       1.00      1.00      1.00         6
weighted avg       1.00      1.00      1.00         6

--------------------------------------------------
```

```
models = ['Logistic Regression', 'Decision Tree', 'Random Forest']
accuracies = [
    accuracy_score(y_test, y_pred_lr),
    accuracy_score(y_test, y_pred_dt),
    accuracy_score(y_test, y_pred_rf)
]

plt.figure(figsize=(7,4))
sns.barplot(x=models, y=accuracies)
plt.title("Model Accuracy Comparison")
plt.ylabel("Accuracy")
plt.show()
```


Model Accuracy Comparison
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression
Decision Tree
Random Forest

RESULTS & OUTPUT

The model performance and outputs are summarized as follows:

Model Accuracy

The Random Forest Classifier achieved 85.71% accuracy, showing strong predictive performance even with a small dataset.

Confusion Matrix

The confusion matrix indicates the number of correct and incorrect classifications, showing balanced predictions for both introverts and extroverts.

Visual Outputs

The following visualizations help understand the dataset better:

- Histogram distribution of all features
- Boxplots for outlier detection
- Correlation heatmap to identify strong relationships
- Feature importance chart showing which behaviours matter most

Together, these results show that ML is capable of understanding personality-related patterns from digital behaviour.

Conclusion

This AIML project successfully demonstrates how machine learning can be applied to analyze social media usage and predict personality traits. Through data preprocessing, visualization, and implementation of the Random Forest Classifier, the project achieved a strong accuracy of 85.71%, indicating that measurable behavioural data can be used to classify individuals as introverts or extroverts.

The results validate the hypothesis that digital behaviour reflects psychological tendencies. Features like daily screen time, follower count, and number of posts per week play a significant role in determining personality traits. The Random Forest model proved to be reliable due to its ability to handle small datasets, reduce overfitting, and produce consistent predictions.

This project provides a foundational understanding of how AI and ML techniques can be used in behavioural analytics, customer profiling, mental health assessment, and targeted marketing. It also highlights the importance of data preprocessing and feature selection in building efficient machine learning systems.

In conclusion, the project showcases a strong integration of AIML concepts, practical implementation, and real-world relevance. It reflects how technology can help understand human psychology in a modern, data-driven world.

References

Below are the references used to support the project development, coding practices, theoretical understanding, and visualization methods:

Books

1. Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. O'Reilly Media.
2. Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer.
3. Tom Mitchell. Machine Learning. McGraw-Hill.

Documentation

1. Scikit-learn Official Documentation: https://scikit-learn.org
2. Pandas Documentation: https://pandas.pydata.org
3. NumPy Documentation: https://numpy.org
4. Matplotlib Documentation: https://matplotlib.org
5. Seaborn Documentation: https://seaborn.pydata.org

Online Resources

1. Towards Data Science – Various articles on ML fundamentals
2. Kaggle Learn – Machine Learning Tutorials
3. Google Colab Documentation
4. Analytics Vidhya – ML & AI Learning Resources

Research Papers

1. "Predicting Personality Traits from Social Media Behavior" – Journal of Computational Psychology
2. "Machine Learning Techniques for Behavioural Classification" – IEEE Access

These references helped guide the methodology, provide clarity on ML algorithms, and support the theoretical