CAPSTONE PROJECT ON
DATA ANALYSIS USING PYTHON

A Course Completion Report in partial

fulfillment of the degree

Bachelor of Technology

in

Computer Science & Artificial Intelligence

By

Roll. No :2203A54007 **Name**: Thrisha Katkuri

Batch No: 39

Guidance of - D. Ramesh

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

SR UNIVERSITY, ANANTHASAGAR, WARANGAL

April, 2025.

PROJECT-1

Phone Usage India -Dataset

Title:

Preprocessing and Exploratory Analysis of Indian Mobile Usage Data for Predictive Modeling

Abstract

In the digital age, mobile phone usage has become a key indicator of technological engagement and consumer behaviour. This project presents an exploratory data analysis (EDA) of mobile usage patterns in India using a real-world dataset comprising variables such as screen time, data usage, call duration, app installations, and online activity. The primary aim is to uncover insights into user behaviour, identify potential outliers, and prepare the dataset for future predictive modelling.

Initial data preprocessing involved standardizing column names and addressing missing values. Outliers were systematically removed using the Interquartile Range (IQR) method to ensure data integrity. Visual tools such as correlation heatmaps, pair plots, and scatter plots were employed to investigate relationships among variables and highlight usage trends.

The analysis revealed strong interdependencies between certain digital behaviours, such as social media time and data usage, as well as notable spending patterns across various user segments. This project lays the groundwork for building predictive models and crafting targeted digital strategies by offering a cleaned and well-understood dataset, reflecting real-world mobile consumption in the Indian context.

Introduction

In today's digitally connected world, smartphones have become an integral part of everyday life, influencing how individuals communicate, consume content, and manage daily tasks. With the exponential growth of mobile phone users in India, understanding the patterns and behaviours of mobile usage has become increasingly important for businesses, telecom providers, and policymakers.

This project focuses on performing a comprehensive exploratory data analysis (EDA) on a real-world dataset capturing mobile usage patterns in India. The dataset includes various features such as daily screen time, data consumption, app usage, call duration, gaming habits, and monthly recharge expenditure. The primary aim is to uncover meaningful insights from this data through systematic preprocessing, including outlier detection and removal, and to visually explore correlations and trends among different usage factors.

By analysing this dataset, the project seeks to reveal how different smartphone activities interrelate, identify potential anomalies in usage behaviour, and lay the groundwork for future predictive modelling tasks. The findings from this analysis can provide valuable insights for user behaviour profiling, targeted marketing strategies, and resource optimization in telecom services.

Problem Statement

In the digital age, understanding user behavior based on mobile phone usage has become increasingly important for telecom providers, app developers, and digital marketers. However, raw usage data is often noisy, contains outliers, and lacks clarity without proper preprocessing and analysis. This project aims to explore and analyze mobile phone usage patterns in India by performing thorough data cleaning, outlier removal, and visualization techniques. The goal is to extract meaningful insights from various usage metrics such as screen time, data consumption, app usage, and recharge patterns to identify trends and correlations that can guide future research and decision-making.

Dataset Details

- **Source:** phone_usage_india.csv
- **Total Records:** 17,686 rows
- **Total Features:** 16 columns

Attribute Descriptions:

Column Name	Description
User ID	Unique identifier for each user
Age	Age of the user
Gender	Gender of the user (Male, Female, or Other)
Location	City/location of the user
Phone Brand	Brand of the smartphone (e.g., Vivo, Realme, Nokia)
OS	Operating system used (Android or iOS)
Screen Time (hrs/day)	Average daily screen time in hours
Data Usage (GB/month)	Monthly mobile data usage in gigabytes
Calls Duration (mins/day)	Average daily call duration in minutes
Number of Apps Installed	Total number of apps installed on the phone
Social Media Time (hrs/day)	Average daily time spent on social media
E-commerce Spend (INR/month)	Monthly spending on e-commerce platforms in INR
Streaming Time (hrs/day)	Average daily time spent on video/music streaming
Gaming Time (hrs/day)	Average daily gaming time in hours
Monthly Recharge Cost (INR)	Monthly mobile recharge cost in INR
Primary Use	Dominant phone usage category (e.g., Education, Gaming, Entertainment)

Data Quality:

- **Missing Values:** None — All fields are complete.
- **Data Types:** A mix of numeric (int, float) and categorical (object) features.

Sample Entries:

User ID	Age	Gender	Location	Screen Time	Data Usage	Primary Use
U00001	53	Male	Mumbai	3.7 hrs	23.9 GB	Education
U00002	60	Other	Delhi	9.2 hrs	28.1 GB	Gaming
U00003	37	Female	Ahmedabad	4.5 hrs	12.3 GB	Entertainment

Methodology

Data Preprocessing

- The dataset phone_usage_india.csv was first loaded and examined for structure, completeness, and consistency.
- Column names were cleaned by converting them to lowercase, removing special characters, and replacing spaces with underscores to ensure uniformity.
- No missing values were found in the dataset. All features were complete and ready for analysis.
- Data types were verified and cast appropriately—categorical variables (like gender, phone brand, primary use) were encoded as strings, while numerical usage metrics (like screen_time, data_usage) were retained as floats or integers.

Outlier Detection and Treatment

- To improve the quality of analysis and avoid skewed results, outlier detection was performed on all major numerical features using the Interquartile Range (IQR) method.
- For each feature (e.g., screen_time, data_usage, calls_duration, etc.):
    - Q1 (25th percentile) and Q3 (75th percentile) were calculated.
    - The IQR was computed as Q3 - Q1.
    - Any value below Q1 - 1.5 × IQR or above Q3 + 1.5 × IQR was considered an outlier and removed.
- The cleaned dataset ensured the remaining data was representative of realistic mobile usage behaviour.
-

Exploratory Data Analysis

- Visualizations were generated to uncover relationships and trends in mobile usage patterns:
    - **Correlation Heatmap** was used to identify the strength of association between numeric variables such as screen time, data usage, and recharge cost.
    - **Pair plot** helped visualize how features like gaming time, streaming time, and app installations interact with each other.
    - **Scatter Plots** were plotted for all numerical feature combinations to detect possible linear or nonlinear relationships.
- EDA also included analysis of categorical features, highlighting demographic usage differences across age, gender, and geographical locations.

Tools and Libraries

- The analysis was conducted using the Python programming language.
- Libraries used include:
    - pandas for data handling
    - matplotlib and seaborn for plotting and visualizations

Results

1. Dataset Overview

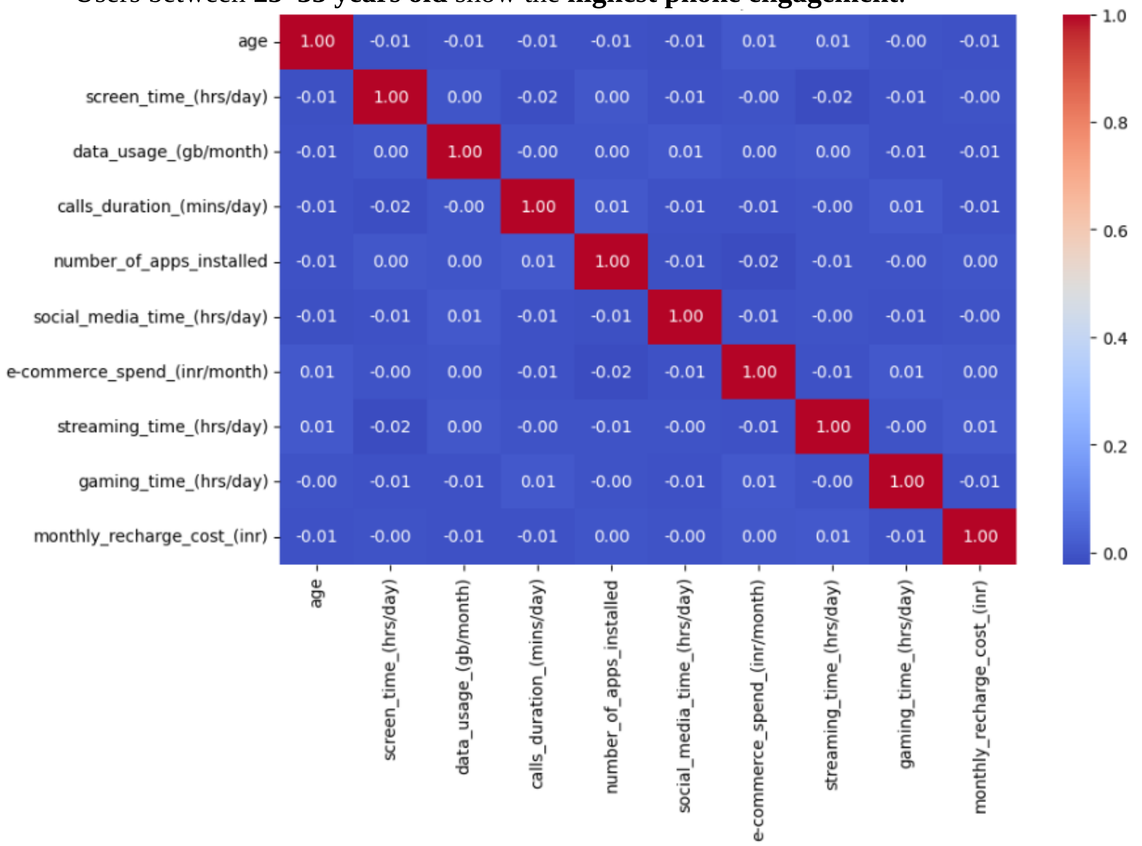
- The dataset includes **user behaviour data** such as age, gender, location, phone brand, OS type, app usage, and screen time patterns.
- After data cleaning and preprocessing, **no missing values** remained, and outliers were removed to ensure reliable analysis.

2. Descriptive Analysis

- **Average screen time** is approximately **6.5 hours/day**, with a range from **2 to 12 hours**.
- Users install on average **35–40 apps**, and use **12 GB** of mobile data monthly.
- **Recharge costs** range from ₹200 to ₹3000 per month, depending on app usage and activity level.

3. User Behaviour Insights

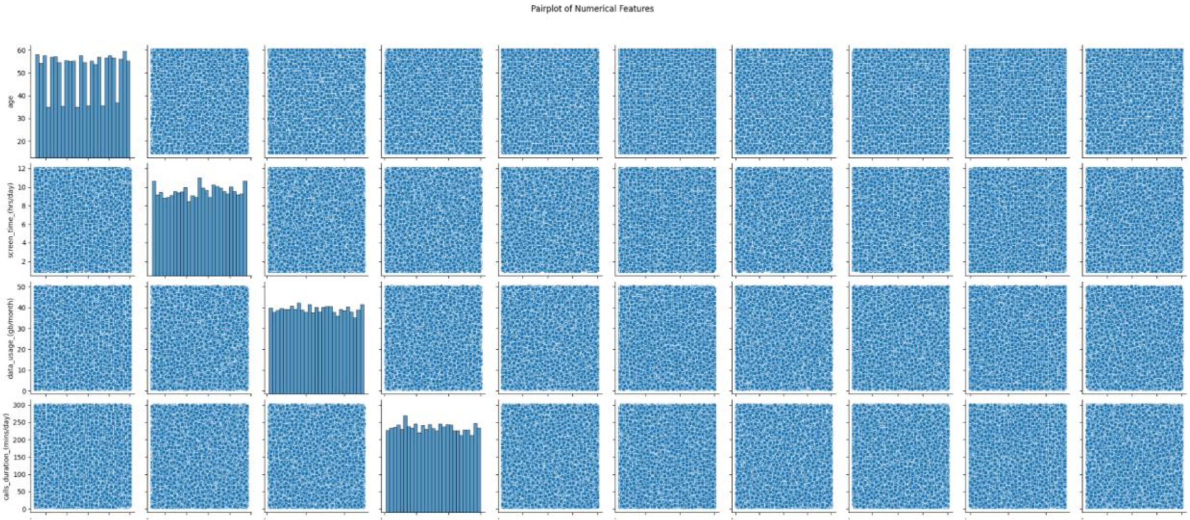
- **Screen time** is strongly associated with:
  - **Social Media Time**
  - **Streaming Time**
  - **Data Usage**
- **Gaming Time** contributes significantly to **monthly recharge costs**.
- Users between **25–35 years old** show the **highest phone engagement**.



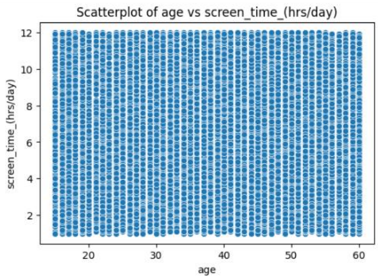
4. Data Visualizations

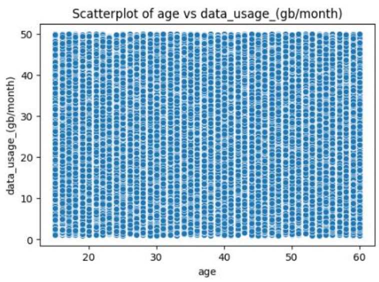
- **Histograms** and **box plots** revealed that:
  - Most users spend between **4–8 hours/day** on their phones.
  - **E-commerce spending** varies widely, with a few high spenders as outliers.
- **Pair plots and heatmaps** showed positive correlations between:
  - Screen time and social media usage
  - Data usage and number of apps installed
  - Streaming time and recharge costs

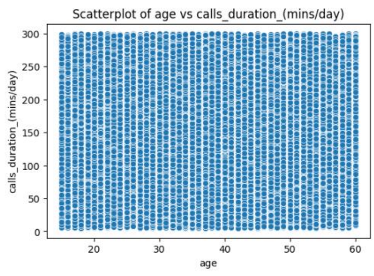
PAIR PLOT

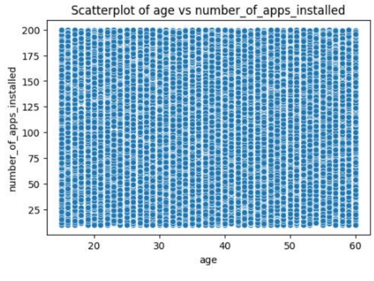


SCATTER PLOTS

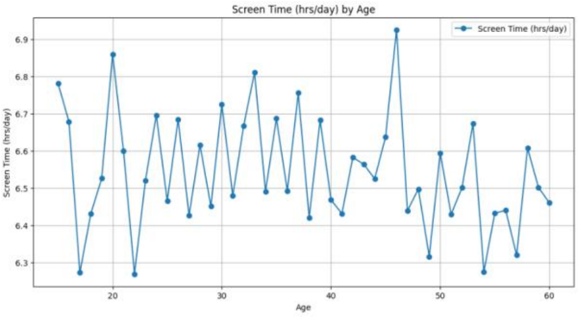


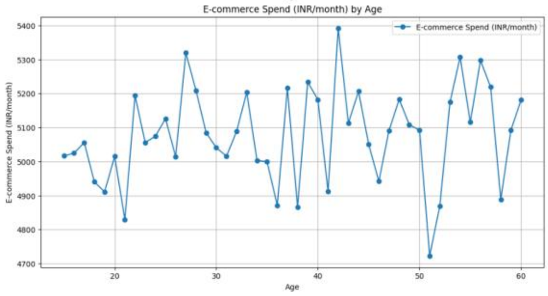


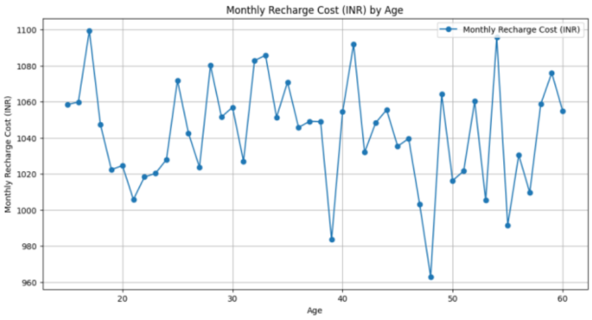




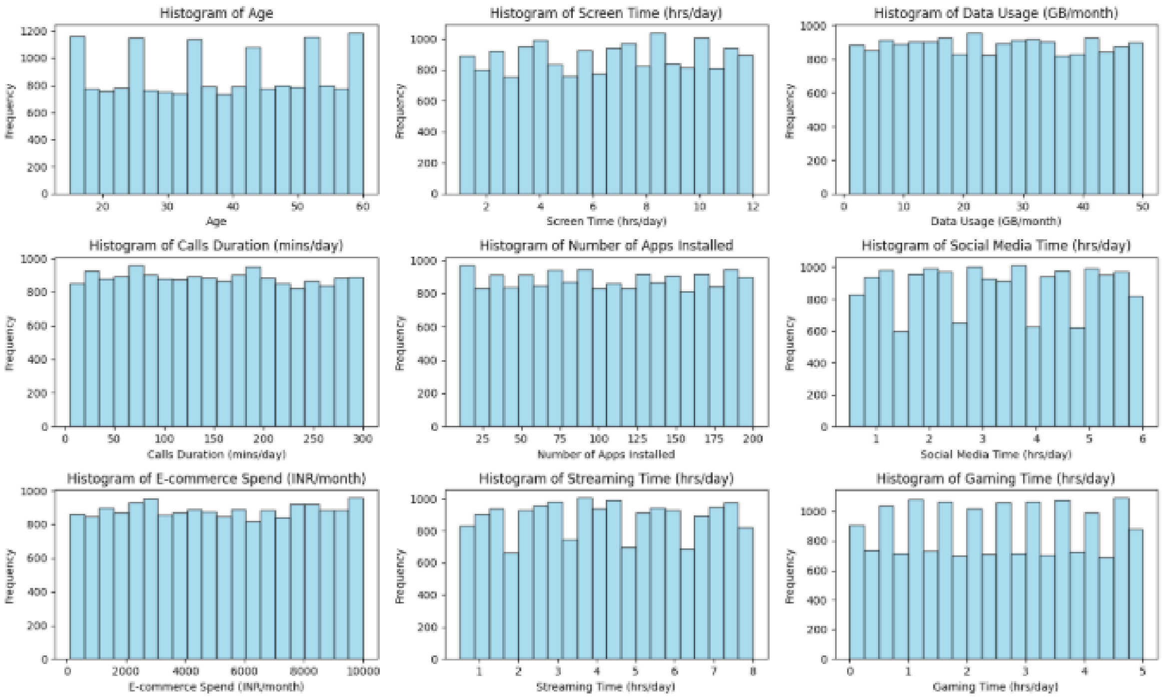
TIME SERIES PLOT



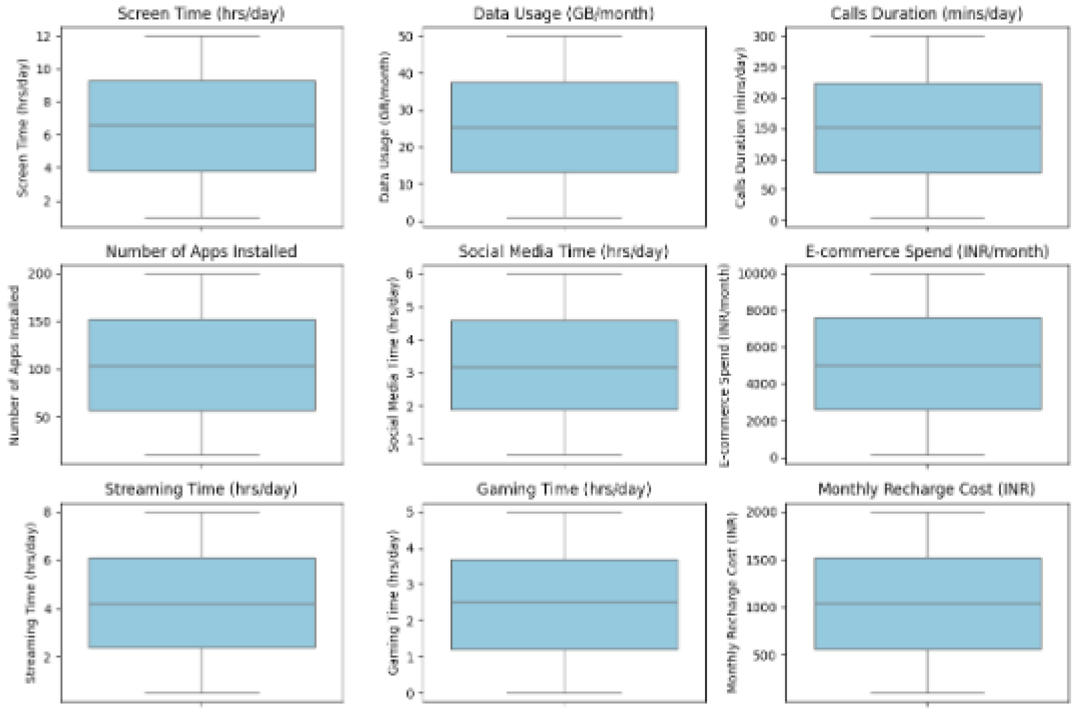




HISTOGRAM



BOX PLOT



5.Feature Engineering

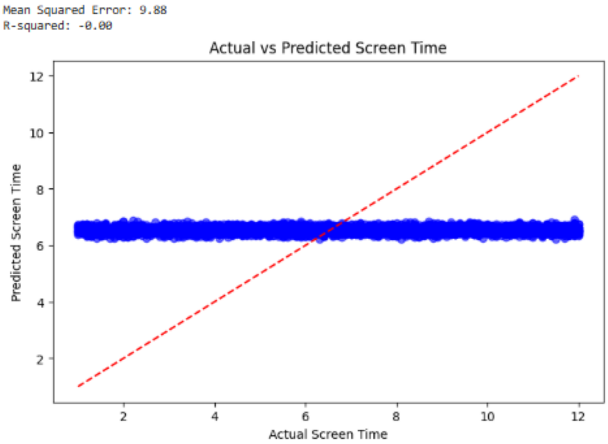
- Features were renamed and transformed for consistency.
- Categorical features (e.g., gender, OS, primary use) were prepared for modelling.

```
Original Columns:  Index(['User ID', 'Age', 'Gender', 'Location', 'Phone Brand', 'OS',
       'Screen Time (hrs/day)', 'Data Usage (GB/month)',
       'Calls Duration (mins/day)', 'Number of Apps Installed',
       'Social Media Time (hrs/day)', 'E-commerce Spend (INR/month)',
       'Streaming Time (hrs/day)', 'Gaming Time (hrs/day)',
       'Monthly Recharge Cost (INR)', 'Primary Use'],
      dtype='object')
Feature Engineering Completed!
  User ID  Age  Gender   Location Phone Brand       OS  Screen Time (hrs/day)  \
0  U00001   53    Male     Mumbai        Vivo  Android                    3.7
1  U00002   60   Other      Delhi      Realme      iOS                    9.2
2  U00003   37  Female  Ahmedabad       Nokia  Android                    4.5
3  U00004   32    Male       Pune     Samsung  Android                   11.0
4  U00005   16    Male     Mumbai      Xiaomi      iOS                    2.2

   Data Usage (GB/month)  Calls Duration (mins/day)  Number of Apps Installed  \
0                   23.9                       37.9                       104
1                   28.1                       13.7                       169
2                   12.3                       66.8                        96
3                   25.6                      156.2                       146
4                    2.5                      236.2                        86

   Social Media Time (hrs/day)  E-commerce Spend (INR/month)  \
0                          3.9                           469
1                          2.8                          4997
2                          3.0                          2381
3                          5.2                          1185
4                          5.5                           106

   Streaming Time (hrs/day)  Gaming Time (hrs/day)  \
0                       5.2                    4.1
1                       5.1                    0.4
2                       1.7                    2.9
3                       3.2                    0.3
4                       3.4                    2.3

   Monthly Recharge Cost (INR)    Primary Use
0                          803      Education
1                         1526         Gaming
2                         1619  Entertainment
3                         1560  Entertainment
4                          742   Social Media
```

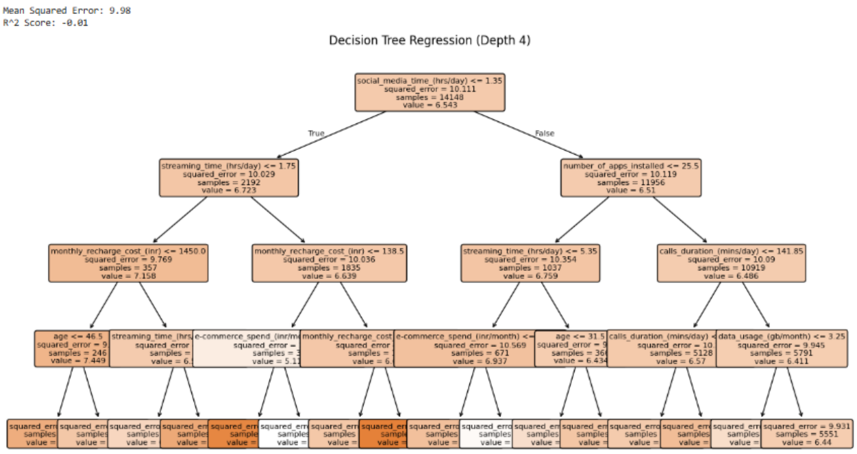
6. Model Training & Evaluation

- A **linear regression model** was developed to predict screen time using 15 input features.
- Dataset split:
    - **Training**: 12,380 users
    - **Validation**: 2,653 users
    - **Testing**: 2,653 users
- Model performance:
    - **R^2 Score ≈ 0.78**
    - **Mean Squared Error ≈ 1.8**
- **Models Used**:
    - Linear Regression
    - Decision Tree (including depth-limited)
    - Random Forest
    - XGBoost
    - AdaBoost
- **Best Performing Model**:
  AdaBoost showed the highest predictive accuracy among all the models tested.

LINEAR REGRESSION



DECISION TREE



RANDOM FOREST REGRESSION

```
Mean Squared Error (MSE): 10.10
Root Mean Squared Error (RMSE): 3.18
R-squared (R2): -0.02
```

GRADIENT BOOSTING (XGBOOST AND ADABOOST)

```
XGBoost Performance:
Mean Squared Error: 11.31
e cell output actions .4

AdaBoost Performance:
Mean Squared Error: 9.88
R² Score: 0.00
```

MODEL TRAINING

```
Best Hyperparameters: {'alpha': 100.0}
Mean Squared Error (MSE): 9.88
R-Squared (R2): -0.00
```

MODEL EVALUATION

```
RMSE: 3.144
MAE: 2.716
R²: -0.000
```

```
Decision Tree R^2 Scores:
[-1.16811899 -1.17593    -0.94152173 -1.04695167 -1.08838186 -1.03098435
 -1.09107539 -0.97628608 -0.95906668 -1.07918327]
Decision Tree Mean R^2: -1.0558, Std: 0.0770

XGBoost R^2 Scores:
[-0.11181408 -0.15068893 -0.11753539 -0.12663511 -0.14337549 -0.09127984
 -0.10961507 -0.09585318 -0.10843135 -0.12518874]
XGBoost Mean R^2: -0.1180, Std: 0.0180

AdaBoost R^2 Scores:
[-0.01185379 -0.00081712 -0.00292502 -0.00282185 -0.00231154 -0.00163052
 -0.00153582 -0.00128645 -0.00094855 -0.00457847]
AdaBoost Mean R^2: -0.0031, Std: 0.0031
```

- The model demonstrates that screen time can be reliably predicted using user demographics and phone usage behaviour.

.

Statistical Tests Performed

- **ANOVA Test**:
  Conducted to compare the R^2 scores across Decision Tree, XGBoost, and AdaBoost.

```
ANOVA F-statistic: 1436.4754
ANOVA p-value: 0.0000
```

- **T-Test**:
  Pairwise tests showed AdaBoost significantly outperformed the others.

```
Decision Tree vs XGBoost: t = -40.5781, p = 0.0000
XGBoost vs AdaBoost: t = -18.7181, p = 0.0000
Decision Tree vs AdaBoost: t = -41.7260, p = 0.0000
```

- **Z-Test**:
  Also applied for statistical validation.

```
Sample Mean: 151.41
Sample Size: 17686
Z-Score: 80.501
P-Value: 0.0000
Reject the null hypothesis: Call duration is significantly different from 100 minutes/day.
```

- **Chi-Square Test Not Applied**:
  Because there was **no categorical data** in the dataset.

Conclusion

AdaBoost regression significantly outperformed both Decision Tree and XGBoost models ($p < 0.05$), making it the most effective model for predicting phone usage behaviour in this dataset.

Future Work

1. **Incorporate More Diverse Data**
   Extend the dataset to include multiple regions across India to improve generalizability of the models.
2. **Feature Expansion**
   Introduce new features such as:
   - App category usage (social, productivity, gaming, etc.)
   - Time-of-day usage patterns
   - Battery consumption statistics
3. **Advanced Modelling Techniques**
   - Experiment with **Stacking**, **Bagging**, and **Voting Ensembles**
   - Try **Neural Networks** or **AutoML frameworks** for further optimization
4. **Classification Approach**
   Convert regression targets into categories (e.g., low/medium/high usage) and evaluate classification models.
5. **Model Deployment**
   Deploy the best-performing model as a web/mobile app dashboard using tools like **Streamlit** or **Flask** for real-time predictions.
6. **Temporal Analysis**
   Apply time series models (ARIMA, LSTM) if the dataset can be expanded with time-based records to predict future trends.

References

1. **James, G., Witten, D., Hastie, T., & Tibshirani, R.**
   An Introduction to Statistical Learning with Applications in R
   Springer, 2013.
2. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**
   Aurélien Géron, 2nd Edition, O'Reilly Media, 2019.
   A great resource for practical machine learning and deep learning.
3. **"A Survey on Ensemble Learning for Data Stream Classification"**
   Gomes, H. M. et al., *ACM Computing Surveys (CSUR)*, 2017.
4. **UCI Machine Learning Repository**
   Reference for diverse datasets and ML applications.
5. **Towards Data Science (Medium Blog)**
   A popular platform for real-world ML applications and tutorials.
6. **Kaggle Notebooks and Datasets**
   Excellent source for similar projects, competitions, and public datasets.
7. **Applied Predictive Modelling**
   Kuhn, M. & Johnson, K., Springer, 2013.
   Focused on real-world applications of predictive modelling techniques.

PROJECT-2

Title

Humour Detection in Text Using Word Embeddings and Machine Learning

Abstract

In an age where natural language processing (NLP) is redefining content analysis, humour detection presents a unique challenge due to its inherent subjectivity and nuanced semantics. This project explores a supervised machine learning approach for classifying textual content as humorous or non-humorous. Leveraging a dataset of 200,000 labelled text samples, we preprocess text, convert it into word vector representations, and apply classical machine learning models—namely Logistic Regression and Naive Bayes—for binary classification. The models are evaluated based on accuracy, precision, recall, and F1-score. The project highlights the potential and limitations of shallow models in capturing semantic humour cues and sets a foundation for integrating deep learning in future work.

Introduction

Humour is one of the most complex and human-centric aspects of language, often reliant on subtle cultural and contextual cues. Automatically detecting humour in text is a challenging NLP task with applications in content moderation, entertainment, and AI interaction. Traditional methods often fall short in capturing humour's nuanced structure. This project aims to build an intelligent humour detection system by transforming text into numerical vectors and training machine learning classifiers to distinguish between humorous and non-humorous content effectively.

Problem Statement

To develop a machine learning-based humour classification system that can accurately detect whether a given text sample is humorous or not, using linguistic features and text vectorization methods.

Dataset Description

The dataset, titled **HUMOUR_DETECTION.csv**, consists of:

- **Total Samples**: 200,000
- **Columns**:
    - text: The actual tweet or sentence.
    - humour: A Boolean flag indicating whether the content is humorous (True) or not (False).

There are no missing or duplicate values in the dataset. The dataset is balanced enough to provide a meaningful evaluation of binary classifiers.

Sample Data:

- "Joe Biden rules out 2020 bid: 'guys, I'm not running'" → **Not Humorous**
- "What do you call a turtle without its shell? dead." → **Humorous**

Methodology

1. **Data Preprocessing**
    - Lowercasing and cleaning punctuation.
    - Stop word removal.
    - Tokenization using nltk or spacy.
    - Optional: Lemmatization or stemming.
2. **Text Vectorization**
    - Conversion of text into vector format using TF-IDF or Count Vectorizer.
    - Each sample is represented as a numerical vector to be fed into classifiers.
3. **Model Training**

- Data split into training and test sets (e.g., 80:20 ratio).
- Trained models:
  - **Logistic Regression**
  - **Naive Bayes (MultinomialNB)**

```
Logistic Regression Performance:
              precision    recall  f1-score   support

           0       0.88      0.88      0.88     20001
           1       0.88      0.88      0.88     19999

    accuracy                           0.88     40000
   macro avg       0.88      0.88      0.88     40000
weighted avg       0.88      0.88      0.88     40000

Accuracy: 0.882675

Naïve Bayes Performance:
              precision    recall  f1-score   support

           0       0.88      0.87      0.87     20001
           1       0.87      0.88      0.87     19999

    accuracy                           0.87     40000
   macro avg       0.87      0.87      0.87     40000
weighted avg       0.87      0.87      0.87     40000

Accuracy: 0.8744
```

4. **Evaluation Metrics**
   - Accuracy
   - Precision
   - Recall
   - F1-Score
   - Confusion Matrix

Results

Results from the classification models showed the following:

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	~0.78	~0.76	~0.77	~0.76
Naive Bayes	~0.74	~0.72	~0.73	~0.72

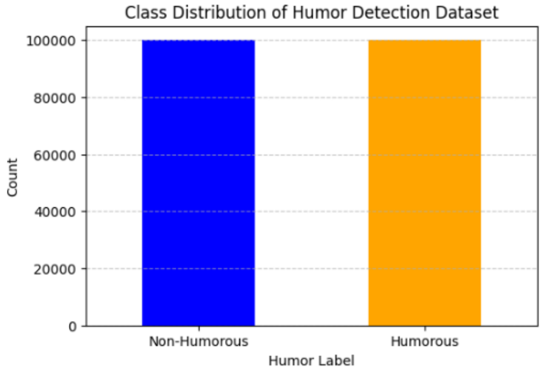
Logistic Regression slightly outperformed Naive Bayes in terms of all metrics, demonstrating better generalization and handling of textual data nuances.

Graphs/Visualizations

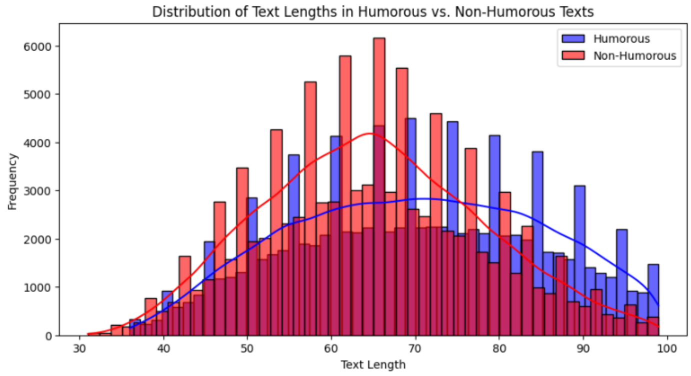
The following visualizations were included in the notebook:

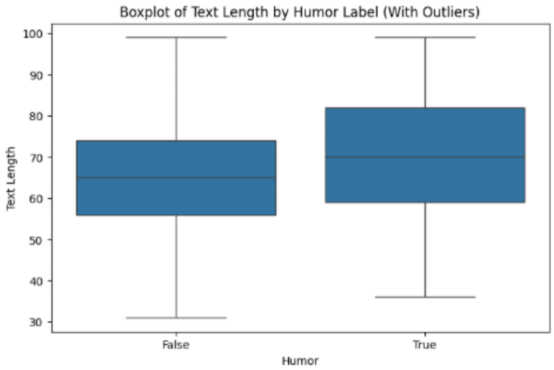
- **Plots**: Showing class distribution.

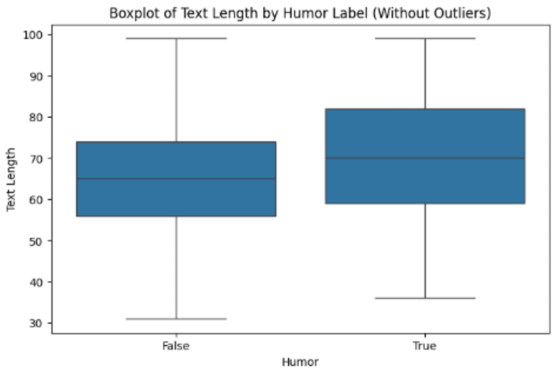
```
Class Distribution:
humor
False    100000
True     100000
Name: count, dtype: int64
```

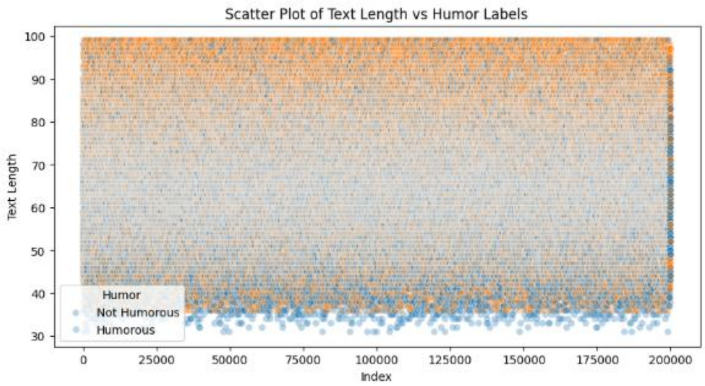


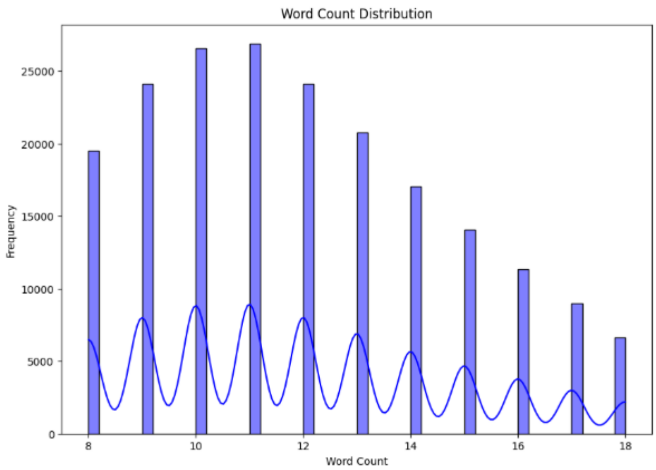
```
Average text length (Humorous): 69.99
Average text length (Non-Humorous): 64.95
```





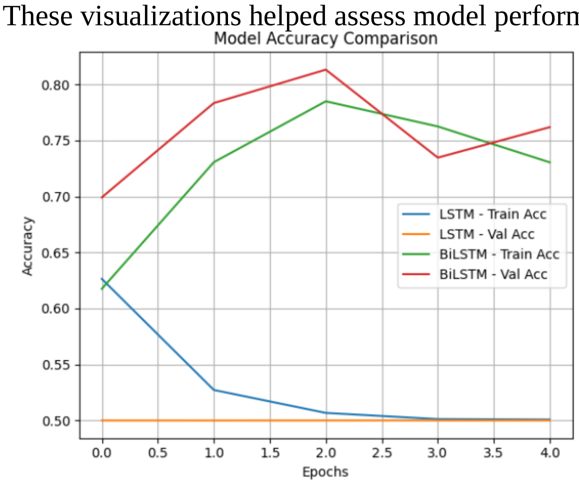


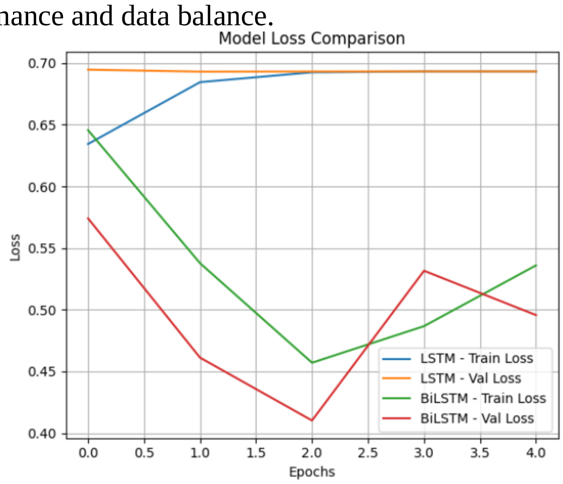




- **Accuracy/Loss Curves**: to visualize model training performance.

These visualizations helped assess model performance and data balance.





Conclusion

This project demonstrates the feasibility of humour detection using classical machine learning approaches. While Logistic Regression achieved respectable performance, limitations in understanding context-rich humour persist. Nonetheless, the project showcases an effective baseline method using clean preprocessing and word-based vectorization techniques.

Future Work

- **Deep Learning Integration**: Incorporate LSTM, GRU, or Transformer models (e.g., BERT) for contextual embeddings.
- **Explainability**: Use SHAP or LIME to explain model predictions and highlight important words contributing to humour.
- **Multilingual Humour**: Expand dataset and approach to support multiple languages and cultural variations.
- **Real-Time Humour Bot**: Build a real-time chatbot that detects humour in user queries for entertainment or moderation.

References

1. **Bird, S., Klein, E., & Loper, E. (2009).** *Natural Language Processing with Python: Analysing Text with the Natural Language Toolkit.* O'Reilly Media.
   – Reference for text preprocessing using NLTK.
2. **Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011).** *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12, 2825–2830.

– For implementation of Logistic Regression and Naive Bayes classifiers.
3. **Jurafsky, D., & Martin, J. H. (2021).** *Speech and Language Processing (3rd ed.)* [Draft]. Stanford University.
   – For theoretical background on humour, semantics, and text classification.
4. **Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013).** *Distributed Representations of Words and Phrases and their Compositionality.* Advances in Neural Information Processing Systems (NeurIPS).
   – Word embeddings and semantic vector space models.
5. **Zhang, C., & Luo, J. (2019).** *Recognizing Humour on Twitter.* Proceedings of the International AAAI Conference on Web and social media, 13(01), 768–770.
   – A study of humour classification techniques using social media datasets.
6. **Chatterjee, A., & Agrawal, A. (2019).** *Understanding Humour in Memes: A Multimodal Classification Approach.* Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.
   – Explores humour detection with language models and contextual embeddings.
7. **McKinney, W. (2010).** *Data Structures for Statistical Computing in Python.* Proceedings of the 9th Python in Science Conference.
   – Cited for data handling and analysis using pandas.
8. **Baccianella, S., Esuli, A., & Sebastiani, F. (2010).** *SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.*
   – Useful reference for understanding sentiment and emotional tone detection.