Department of Computer Engineering
BE Computer-B (2025-26 Sem I)
LP-III Machine Learning
Practical Assignment 1: Regression

[CO1-CO2, BT: L3 (Apply)] [Max Marks: 10]

Problem Definition: Predict the price of the Uber ride from a given pickup point to the agreed drop-off location.

Perform following tasks:

1. Pre-process the dataset.

2. Identify outliers.

3. Check the correlation.

4. Implement linear regression and random forest regression models.

5. Evaluate the models and compare their respective scores like R2, RMSE, etc.

Dataset link: https://www.kaggle.com/datasets/yasserh/uber-fares-dataset

Learning Outcomes:

By completing this practical, students will be able to:

Learning Outcome	**Bloom's Taxonomy Level**
Understand the importance of data preprocessing in ML	Understand
Apply Linear Regression and Random Forest Regressor models	Apply
Analyze the performance of different ML models	Analyze
Evaluate regression models using performance metrics	Evaluate

Software / Hardware Requirements

Software:

- Python 3.8+
- Jupyter Notebook / VS Code
- Libraries: pandas, numpy, matplotlib, seaborn, sklearn, geopy

Hardware:

- Minimum 4 GB RAM

- Dual-core processor
- Stable internet connection (for data access)

Theory:

Uber is a ride-hailing service where customers' book rides via an app, and fares are calculated based on several factors like distance, time, and traffic. In this lab, we use historical ride data to build a model that can predict the fare amount given the ride's input features. This is a classic regression problem in machine learning.

1. Dataset Description:

The project is about on world's largest taxi company Uber inc. In this project, we're looking to predict the fare for their future transactional cases. Uber delivers service to lakhs of customers daily. Now it becomes really important to manage their data properly to come up with new business ideas to get best results. Eventually, it becomes really important to estimate the fare prices accurately.

The dataset contains the following fields:

- key - a unique identifier for each trip
- fare_amount - the cost of each trip in usd
- pickup_datetime - date and time when the meter was engaged
- passenger_count - the number of passengers in the vehicle (driver entered value)
- pickup_longitude - the longitude where the meter was engaged
- pickup_latitude - the latitude where the meter was engaged
- dropoff_longitude - the longitude where the meter was disengaged
- dropoff_latitude - the latitude where the meter was disengaged

Acknowledgement:

The dataset is referred from Kaggle: https://www.kaggle.com/datasets/yasserh/uber-fares-dataset

2. Data Preprocessing:

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

Key steps involve cleaning data, removing outliers, and handling missing values, feature engineering, and correlation analysis.

3. Supervised Learning Overview:

Supervised learning is a machine learning approach where the model is trained on a labeled dataset (i.e., input-output pairs). The goal is to learn a mapping from input variables (**features**) to an output variable (**target**).

- **Input Features**: pickup/drop-off coordinates, time, passenger count
- **Target**: fare amount (a continuous numeric value)

Since we are predicting a **continuous value**, we use **regression algorithms**.

3.1 Linear Regression:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc. It models the relationship between one or more independent variables (features) and a dependent variable (target) by fitting a **linear equation** to the observed data.

Mathematical Model:

For **Simple Linear Regression** (with one feature):

$y=\beta0+\beta1x+\epsilon$

For **Multiple Linear Regression** (multiple features):

$y=\beta0+\beta1x1+\beta2x2+\cdots+\beta nxn+\epsilon$

Where:

- y: Predicted fare amount
- x1,x2,...,xnx_1, x_2, ..., x_nx1,x2,...,xn: Input features (e.g., distance, passenger count, time of day)
- β0: Intercept (constant term)
- β1,β2,...,βn: Coefficients (weights)
- ϵ: Error term (difference between actual and predicted values)

Advantages:

- Simple to implement and fast to compute
- Easily interpretable
- Good baseline model for regression problems

Limitations:

- Fails to capture **non-linear relationships**
- Sensitive to **outliers**
- Assumes **linearity**, which may not always be true in real-world datasets like Uber fares

3.2 Random Forest Regressor: Random Forest is an **ensemble learning** technique that combines the predictions of multiple **decision trees** to improve accuracy and robustness. When used for regression tasks (like fare prediction), the model outputs the **average** of the predictions made by individual trees.

It belongs to the family of **Bagging (Bootstrap Aggregating)** methods and is particularly effective for datasets with **non-linear relationships** and **interactions among variables**.

How It Works:

1. **Bootstrap Sampling**:
   - From the original dataset, **multiple random samples (with replacement)** are generated.
   - Each sample is used to train a different decision tree (this is called **bagging**).
2. **Random Feature Selection**:
   - At each split in the decision tree, a **random subset of features** is considered rather than all features.
   - This introduces more diversity among trees, reducing overfitting.
3. **Tree Construction**:
   - Each tree is grown **to its maximum depth** (or limited depth, based on parameters).
   - No pruning is applied.
4. **Prediction Aggregation**:
   - For regression, each tree outputs a numeric prediction.
   - The final prediction is the **average** of all tree outputs.

$$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} y_i^{(tree)}$$

Advantages:

- **Handles non-linearity:** Unlike linear models, it does not assume a linear relationship.
- **Robust to outliers and noise:** Since predictions are averaged, outliers have less influence.
- **Automatically detects feature importance:** Helps in feature selection.
- **Reduces overfitting:** Due to randomization and ensemble averaging.

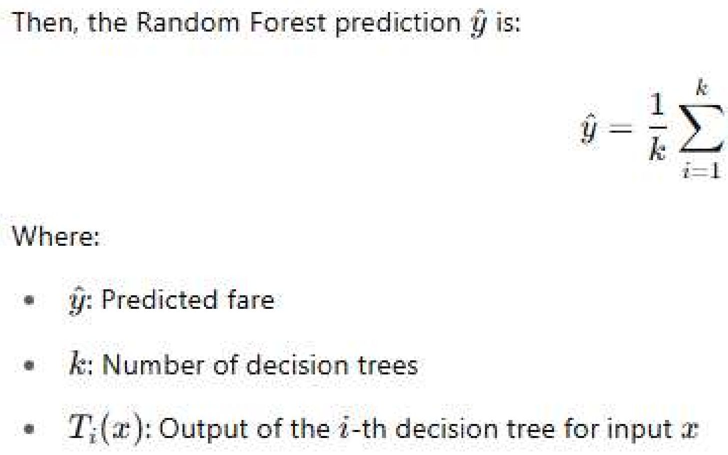
Limitations:

- **Less interpretable** than linear models (it's a black-box).
- **Computationally intensive**, especially with large datasets and many trees.
- May **overfit** if the number of trees or depth is too high without proper tuning.

Mathematical Model

Let:

- $T_1, T_2, \ldots, T_k$ be k regression trees in the forest.
- Each tree $T_i$makes a prediction $T_i(x)$ for input vector x.



Then, the Random Forest prediction $\hat{y}$ is:

$$\hat{y} = \frac{1}{k} \sum_{i=1}^{k} T_i(x)$$

Where:

- $\hat{y}$: Predicted fare
- k: Number of decision trees
- $T_i(x)$: Output of the i-th decision tree for input x

This is a **bagged estimate** (average) over multiple trees trained on different data and feature subsets.

4. Model Evaluation Metrics:

After building models, we evaluate how well they predict fares using:

Metric	**Description**
MAE (Mean Absolute Error)	Average of the absolute differences between predicted and actual fares.
RMSE (Root Mean Squared Error)	Square root of the average squared differences. Penalizes large errors.
R² Score (Coefficient of Determination)	Indicates how much of the variance in the target variable is explained by the model (ranges from 0 to 1). Higher is better.

Distance Calculation – Haversine Formula:

To estimate ride distance based on latitude and longitude, we use:

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

Where:

- ϕ= latitude, λ = longitude (in radians)
- r = Earth's radius (~6371 km)

This distance becomes a major feature in predicting fare.

5. Algorithm

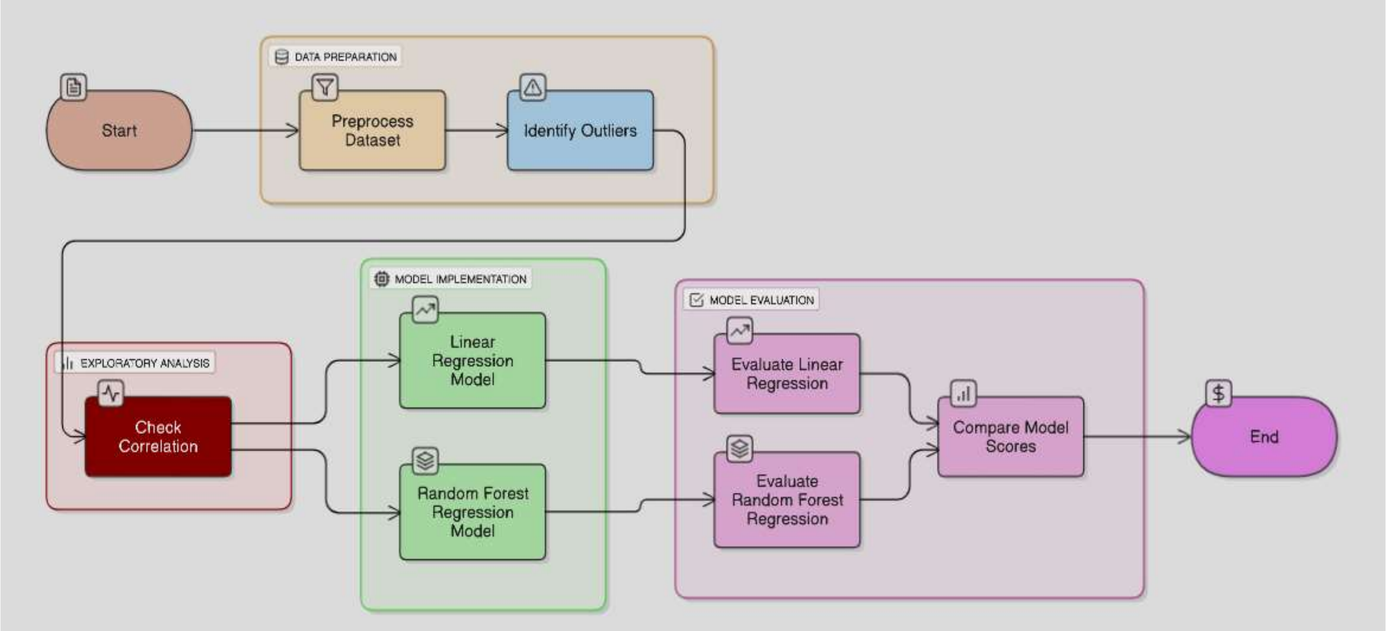
Linear Regression Algorithm

1. Load and clean the dataset.
2. Extract datetime features and compute ride distance.
3. Split dataset into training and test sets.
4. Fit a linear regression model on training data.
5. Predict fare amount and evaluate model.

Random Forest Algorithm

1. Load and preprocess the dataset.
2. Create distance and time-based features.
3. Split data into train/test.
4. Fit Random Forest Regressor on training data.
5. Predict fare and evaluate using metrics.

6. Flow Chart



Result:

(Give Comparative Performance analysis of linear regression and random forest regression models using various evaluation metrics in Tabular Form)

Conclusion

The Random Forest Regressor performed better than Linear Regression for Uber fare prediction due to its ability to model non-linear relationships and reduce overfitting. Regression metrics confirmed improved accuracy and reduced error.

Viva Questions

Question	**Bloom's Level**
What is regression in machine learning?	Remember
Why do we remove outliers from the dataset?	Understand
How do you calculate the distance between two coordinates?	Apply
Explain the difference between RMSE and MAE.	Understand
Which model performed better and why?	Analyze
How would you improve the prediction accuracy further?	Evaluate
Can you suggest an alternate feature that might help improve performance?	Create

Rubrics for Evaluation (Example)

Question No.	**Criteria**	**Best**	**Good**	**Average**	**Poor**
Q1	Correctness of Program / Demonstration of Program [4]	Perfect [4]	Can be better [3]	Satisfied [2]	Poor [0]
	File Submission/Documentation [2]	Perfect [2]	Can be better [1.5]	Satisfied [1]	Poor [0]
	Timely Submission [2]	On Time [2]	After 1 week [1.5]	After 1.5 week [1]	At the end of semester [0]
	Viva [2]	Perfect [2]	Can be better [1.5]	Satisfied [1]	Poor [0]

Dr. Nikita Singhal

(Subject Incharge)

DR SR Dhore

(HoD)