PHASE 4 ASSIGNMENT

PROBLEM STATEMENT:

AI-Driven Exploration and Prediction of Company Registration Trends with Registrar of Companies (RoC).

PROJECT TITLE: Feature selection, Model training, Evaluation of an dataset

PROBLEM DEFINITION:

The problem is to perform an AI-driven exploration and predictive analysis on the master details of companies registered with the Registrar of Companies (RoC). The objective is to uncover hidden patterns, gain insights into the company landscape, and forecast future registration trends. This project aims to develop predictive models using advanced Artificial Intelligence techniques to anticipate future company registrations and support informed decision-making for businesses, investors, and policymakers.

GITHUB LINK:https://github.com/sheshapriya/RoC.git

https://github.com/sheshapriya/innovation.git

DATASET LINK: **https://tn.data.gov.in/resource/company-master-data-tamil-nadu-upto-28th-february-2019**

DOCUMENT: Building the project by Feature selection, Model training, Evaluation of an dataset

Exploratory Data Analysis (EDA):

EDA is the process of analyzing and visualizing your dataset to gain insights and understand its characteristics. Here's how you can perform EDA:

- ✓ Data Loading: Load your dataset into a suitable data analysis environment, such as Python using libraries like Pandas.
- ✓ Data Summary: Start by getting a high-level overview of your data. Use functions like head(), info(), and describe() to understand the dataset's structure, missing values, and basic statistics.
- ✓ Data Visualization: Create various plots and charts to visualize your data, such as histograms, scatter plots, box plots, and correlation matrices. Visualization libraries like Matplotlib and Seaborn are helpful.
- ✓ Data Cleaning: Address missing values, outliers, and any data inconsistencies you identified during EDA. You may need to impute missing values, remove outliers, or perform data transformations.
- ✓ Feature Analysis: Examine the relationships between features and the target variable. Identify features that may have a strong influence on the prediction task.

Feature Engineering:

Feature engineering involves creating new features or modifying existing ones to improve the predictive power of your model. Here are some common feature engineering techniques:

- ✓ Feature Creation: Generate new features that may be more informative for your prediction task. For example, you can create date-related features from timestamp data or combine multiple features to create interaction terms.
- ✓ Encoding Categorical Data: Convert categorical variables into numerical representations using techniques like one-hot encoding or label encoding.
- ✓ Scaling and Normalization: Standardize or normalize numerical features to ensure they have similar scales, which can be essential for certain machine learning algorithms.

- ✓ Feature Selection: Use techniques like correlation analysis, feature importance scores, and recursive feature elimination to select the most relevant features.

Predictive Modeling:

After EDA and feature engineering, you can proceed with building and training predictive models. Here are the steps for this phase:

- ✓ Data Splitting: Split your dataset into training and testing sets to evaluate the model's performance. Common splits are 70-30 or 80-20, but you can adjust based on the dataset size.
- ✓ Model Selection: Choose a machine learning algorithm appropriate for your prediction task. Common choices include linear regression, decision trees, random forests, gradient boosting, support vector machines, and neural networks.
- ✓ Model Training: Train the selected model on the training dataset using appropriate libraries like scikit-learn or TensorFlow/Keras.
- ✓ Model Evaluation: Assess the model's performance using relevant evaluation metrics, such as mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), R-squared, or custom metrics specific to your task.
- ✓ Hyperparameter Tuning: Optimize the model's hyperparameters to improve its performance. You can use techniques like grid search or random search.
- ✓ Model Validation: Validate the model's performance on the testing dataset to ensure it generalizes well to unseen data.
- ✓ Visualization: Create visualizations to interpret model results and predictions, such as feature importance plots, prediction vs. actual plots, and error distribution histograms.
- ✓ Deployment: Once satisfied with your model, deploy it to make predictions on new data.

Deployment:

```
# Import necessary 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.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Load the dataset
data = pd.read_csv('house_prices.csv')

# Exploratory Data Analysis (EDA)
# Basic data summary
print(data.head())
print(data.info())

# Data Visualization
sns.pairplot(data, x_vars=['sqft', 'bedrooms', 'bathrooms'], y_vars='price', height=4, aspect=1)
plt.show()

# Feature Engineering
# Example: Creating a feature 'age' based on 'year_built'
data['age'] = 2023 - data['year_built']

# Encoding categorical features (if any)
```

```
data = pd.get_dummies(data, columns=['location'], drop_first=True)

# Splitting data into features (X) and target (y)
X = data.drop(['price'], axis=1)
y = data['price']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Predictive Modeling
# Example: Using a Random Forest Regressor
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Model Evaluation
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

In this code:

- We load a sample dataset named 'house_prices.csv.'
- We perform exploratory data analysis (EDA) by displaying the first few rows, getting information about the dataset, and creating scatterplots to visualize the relationships between features and the target variable.
- We conduct feature engineering by creating a new 'age' feature based on 'year_built' and encoding categorical features.
- We split the data into training and testing sets.
- We build a predictive model using the Random Forest Regressor.
- We make predictions on the test set and evaluate the model's performance using mean squared error (MSE) and R-squared.

Monitoring and Maintenance:

- ✓ Continuously monitor the model's performance and update it as necessary to ensure it remains accurate and relevant.

SUBMITTED BY,

STUDENT REG NO: 711221104051

NAAN MUDHALVAN: au711221104051