ADMISSION PREDICTION USING LINEAR REGRESSION

Abstract

This project aims to develop a predictive model for university admissions using linear regression, focusing on key factors such as GRE scores, TOEFL scores, Statement of Purpose (SOP), and Letters of Recommendation (LOR). By analyzing these features, we seek to determine their impact on the likelihood of admission. The linear regression model is trained and evaluated using a dataset that includes these variables. The results indicate that the model can effectively predict admission outcomes. This study offers a data-driven approach to understanding the admissions process, benefiting both prospective students and educational institutions.

IMPLEMENTATION

Introduction:

In this project, we develop a linear regression model to predict university admission chances based on key factors such as GRE scores, TOEFL scores, Statement of Purpose (SOP), Letters of Recommendation (LOR), CGPA, and research experience. The implementation involves several steps, which are outlined below:

Dataset Description:

The dataset used in this project is sourced from Kaggle's Admission Prediction dataset. It contains information on various factors that influence the admission chances of students applying to graduate programs. The dataset includes the following features:

Features:

- GRE Score: Graduate Record Examination score (out of 340).
- TOEFL Score: Test of English as a Foreign Language score (out of 120).
- University Rating: Rating of the university (on a scale of 1 to 5).
- SOP: Strength of Statement of Purpose (on a scale of 1 to 5).
- LOR: Strength of Letters of Recommendation (on a scale of 1 to 5).
- CGPA: Undergraduate GPA (on a scale of 10).
- Research: Research experience (0 or 1).

Target:

- Chance of Admit: Probability of admission (ranging from 0 to 1).

The dataset consists of 400 observations, providing a comprehensive overview of the factors that impact admission decisions. The data is preprocessed to handle any missing values and ensure it is suitable for training the linear regression model.

Example:

S. No	GRE Score	TOEFL Score	University Ranking	SOP	LOR	CGPA	Research	Chance of Admit
1	337	118	4	4.5	4.5	9.65	1	0.92
2	324	107	4	4	4.5	8.87	1	0.76
3	316	104	3	3	3.5	8	1	0.72

Loading the Dataset: We use the ‘pandas’ library to load the dataset containing the relevant features and target variable.

```
import pandas as pd

data = pd.read_csv('/content/Admission_Predict.csv')
print(data.head()) #sample data display
print(data.isna().sum()) #check for null values
```

Data Preparation: The dataset is preprocessed to select the necessary features (GRE Score, TOEFL Score, University Rating, SOP, LOR, CGPA, Research) and the target variable (Chance of Admit).

```
  x = data.iloc[:,1:8] #index locator[no.of.rows,no.of cols]- all rows and 8 cols
  y= data.iloc[:,8:9]
```

Data Splitting: The dataset is split into training and testing sets using an 80-20 split to ensure that the model is trained on a subset of the data and tested on unseen data.

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2)
```

Model Training: We employ the ‘LinearRegression’ class from the ‘sklearn’ library to train the linear regression model on the training data.

```
from sklearn.linear_model import LinearRegression
L = LinearRegression()
L.fit(xtrain, ytrain)
```

Making Predictions: The trained model is used to make predictions on the test set, allowing us to evaluate its performance.

```
ypred = L.predict(xtest)
```

Model Evaluation: The model’s accuracy is assessed using the R-squared score and Mean Squared Error (MSE), providing insights into its predictive power.

```
from sklearn.metrics import r2_score, mean_squared_error
print("Accuracy:", r2_score(ytest, ypred))
print("MSE:", mean_squared_error(ytest, ypred))
```

Model Coefficients: The coefficients of the linear regression model are extracted to understand the impact of each feature on the admission chances.

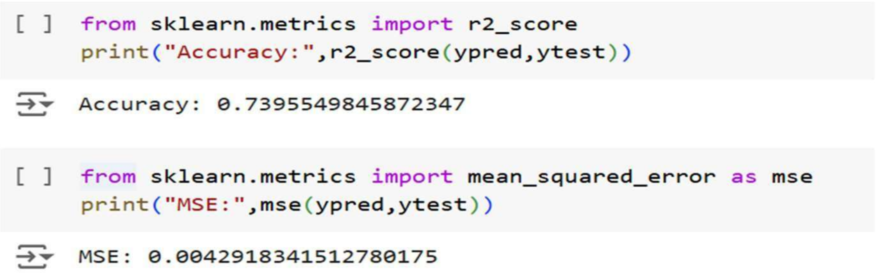
```
print("Coefficients:", L.coef_)
```

Predicting New Data: Finally, the model is used to predict the admission chance for a new set of input values, demonstrating its practical application.

```
new_input = [320, 108, 3, 3.5, 4, 8.44, 1]
predicted_value = L.predict([new_input])
print("Predicted Chance of Admit:", predicted_value)
```

OUTPUT

- R2 and Mean Square Values:



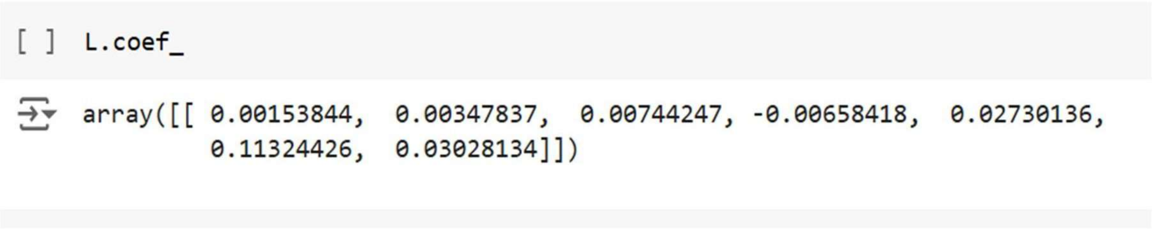
```
[ ]  from sklearn.metrics import r2_score
     print("Accuracy:",r2_score(ypred,ytest))

     Accuracy: 0.7395549845872347

[ ]  from sklearn.metrics import mean_squared_error as mse
     print("MSE:",mse(ypred,ytest))

     MSE: 0.0042918341512780175
```

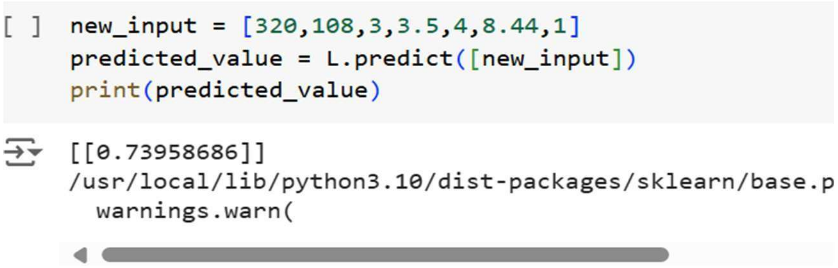
- Coefficients of Linear Regression:



```
[ ]  L.coef_

     array([[ 0.00153844,  0.00347837,  0.00744247, -0.00658418,  0.02730136,
              0.11324426,  0.03028134]])
```

- Predicted Output from Test Set:



```
[ ]  new_input = [320,108,3,3.5,4,8.44,1]
     predicted_value = L.predict([new_input])
     print(predicted_value)

     [[0.73958686]]
     /usr/local/lib/python3.10/dist-packages/sklearn/base.p
       warnings.warn(
```

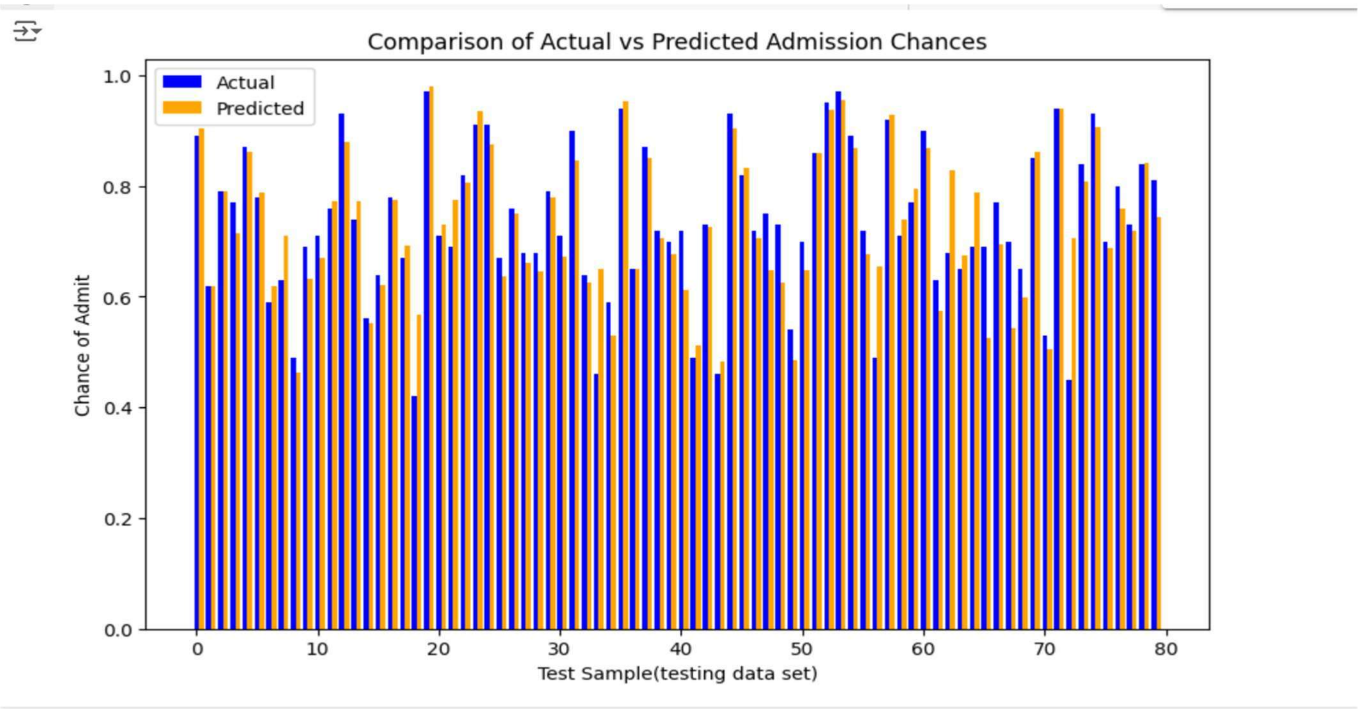
Actual Data:

GRE Score	TOEFL Score	University Ranking	SOP	LOR	CGPA	Research	Chance of Admit
320	108	3	3.5	4	8.44	1	0.76

ANALYSIS AND FLOW CHART

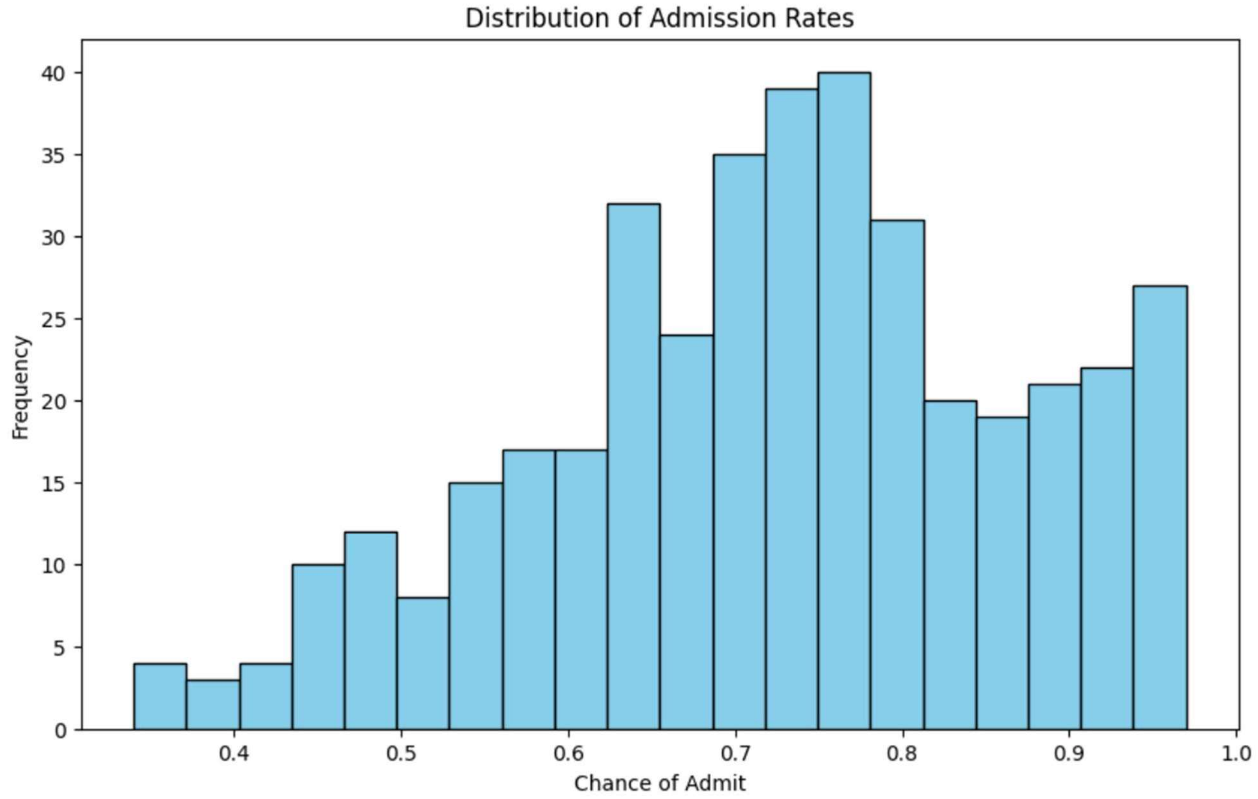
1. Prediction Analysis:

```
#Prediction Analysis
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
index = range(len(ypred))  # Number of predictions
plt.bar(index, ytest.values.flatten(), color='blue', width=0.4, label='Actual')
plt.bar([i + 0.4 for i in index], ypred.flatten(), color='orange', width=0.4,
label='Predicted')
plt.xlabel('Test Sample(testing data set)')
plt.ylabel('Chance of Admit')
plt.title('Comparison of Actual vs Predicted Admission Chances')
plt.legend()
plt.show()
```

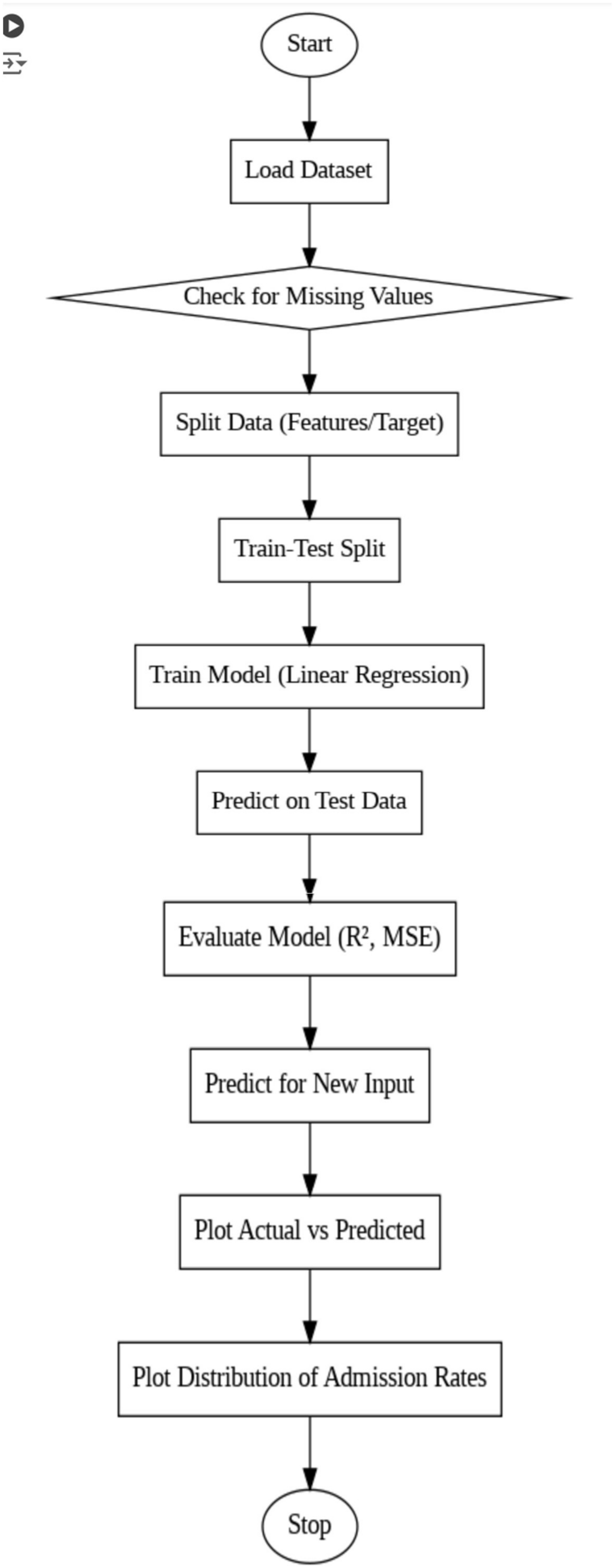


2. Admission Rate Analysis:

```
#Admission Analysis
import matplotlib.pyplot as plt
admission_rates = data['Chance of Admit']
plt.figure(figsize=(10, 6))
plt.hist(admission_rates, bins=20, color='skyblue', edgecolor='black')
plt.xlabel('Chance of Admit')
plt.ylabel('Frequency')
plt.title('Distribution of Admission Rates')
plt.show()
```



3. Program Flow Representation using digraph:


Start
Load Dataset
Check for Missing Values
Split Data (Features/Target)
Train-Test Split
Train Model (Linear Regression)
Predict on Test Data
Evaluate Model (R², MSE)
Predict for New Input
Plot Actual vs Predicted
Plot Distribution of Admission Rates
Stop

ALGORITHM

Predicting Admission Chances Using Linear Regression:

Step 1: Load the dataset into a Data Frame.
Step 2: Check for missing values and inspect the dataset.
Step 3: Select relevant features as input variables (e.g., GRE Score, TOEFL Score) and the target variable (Chance of Admit).
Step 4: Split the dataset into training and test sets (e.g., 80% training, 20% testing).
Step 5: Train a Linear Regression model using the training data.
Step 6: Use the trained model to make predictions on the test data.
Step 7: Evaluate the model’s performance using metrics like R^2 score and Mean Squared Error (MSE).
Step 8: Predict admission chances for new inputs using the trained model.
Step 9: Visualize predictions using a bar chart comparing actual vs predicted values.

CONCLUSION

In this project, we aimed to predict university admission chances using a linear regression model based on key factors such as GRE scores, TOEFL scores, Statement of Purpose (SOP), Letters of Recommendation (LOR), CGPA, and research experience. The dataset was sourced from Kaggle and included 400 observations with these features.

This study highlights the potential of using linear regression for admission prediction, providing valuable information for prospective students and educational institutions. The model can help students understand which factors are most critical for their admission chances and allow universities to make more informed decisions.

Future work could involve exploring more advanced machine learning techniques, such as decision trees or neural networks, to further improve the model's accuracy. Additionally, incorporating more features, such as extracurricular activities or personal statements, could provide a more comprehensive view of the admission process.

For a detailed implementation and code, please refer to my Colab notebook:
https://colab.research.google.com/drive/1jYUC1IOTYtA0oSZ6zxIVtjVzulyT6gPM#scrollTo=J5bsYjwz3J-c