Sprint 5

Day 10 Task 2:

Sprints for AI
Use Case For Demand Forecasting Using POS Transaction Data

Team Members:

Tufail Irshad
Amjad Ali Malik
Zameer Ahmad Mir
Asif Ahmad Najar
Irfan Ahmad Mutoo

Sprint 5 Day 10 Task 2

AI Regression Diagrams Documentation

Visualization is a powerful tool in data analysis and model evaluation, allowing us to gain insights into patterns, relationships, and the performance of machine learning models. Through visualizations, we can effectively communicate complex information in a clear and intuitive manner.

In the context of machine learning and regression analysis, visualization plays a crucial role in several aspects:

Data Exploration:

Before building models, it's essential to understand the data. Visualizations such as histograms, box plots, and scatter plots help to explore the distribution of variables, identify outliers, and detect relationships between features and the target variable.

Model Evaluation:

After training regression models, visualizations help assess their performance. Plots comparing actual vs. predicted values provide a quick overview of how well the model captures the underlying patterns in the data. Additionally, visualizations of evaluation metrics such as R-squared, MSE, and RMSE allow for easy comparison of model performance across different algorithms.

Feature Importance:

Visualizations such as bar plots or heatmaps of feature importance help identify which features have the most significant impact on the target variable. This information can guide feature selection and feature engineering efforts to improve model performance.

Model Interpretability:

Certain visualization techniques, such as partial dependence plots or SHAP (SHapley Additive exPlanations) plots, help explain the predictions of complex models. These plots illustrate how changes in input features affect the model's output, providing valuable insights into its decision-making process.

Hyperparameter Tuning:

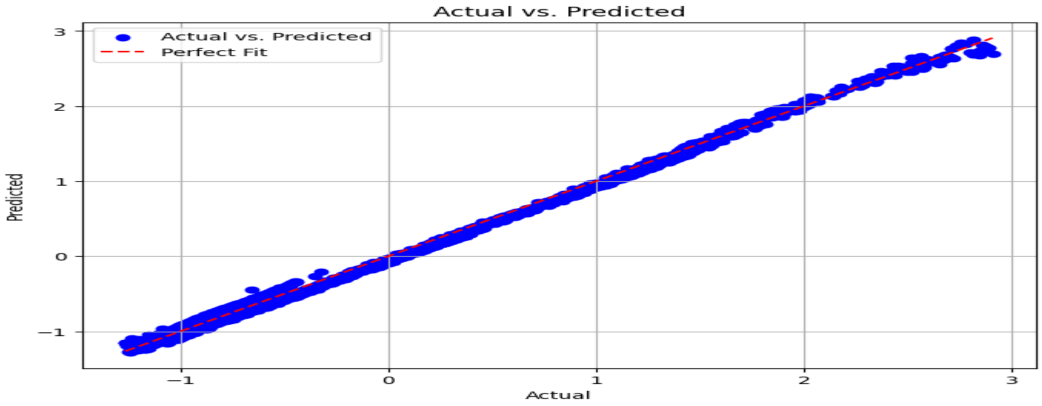
Visualizations can aid in hyperparameter tuning by plotting model performance metrics across different parameter values. This helps identify the optimal hyperparameters for the best model performance.

By leveraging visualizations at various stages of the machine learning pipeline, from data exploration to model evaluation and interpretation, we can effectively analyze and communicate the results of regression analysis, leading to more informed decision-making and improved model performance.

Scatter plot of Actual vs. Predicted values:

- This plot compares the actual target values (y_test) with the predicted values (y_pred) obtained from a regression model.
- Each point in the plot represents an observation in the dataset.
- Ideally, all the points should fall along the diagonal line (the red dashed line), which represents a perfect fit where actual and predicted values are equal.
- The blue dots represent how close the predicted values are to the actual values. If the model is good, the points should be clustered around the red line.

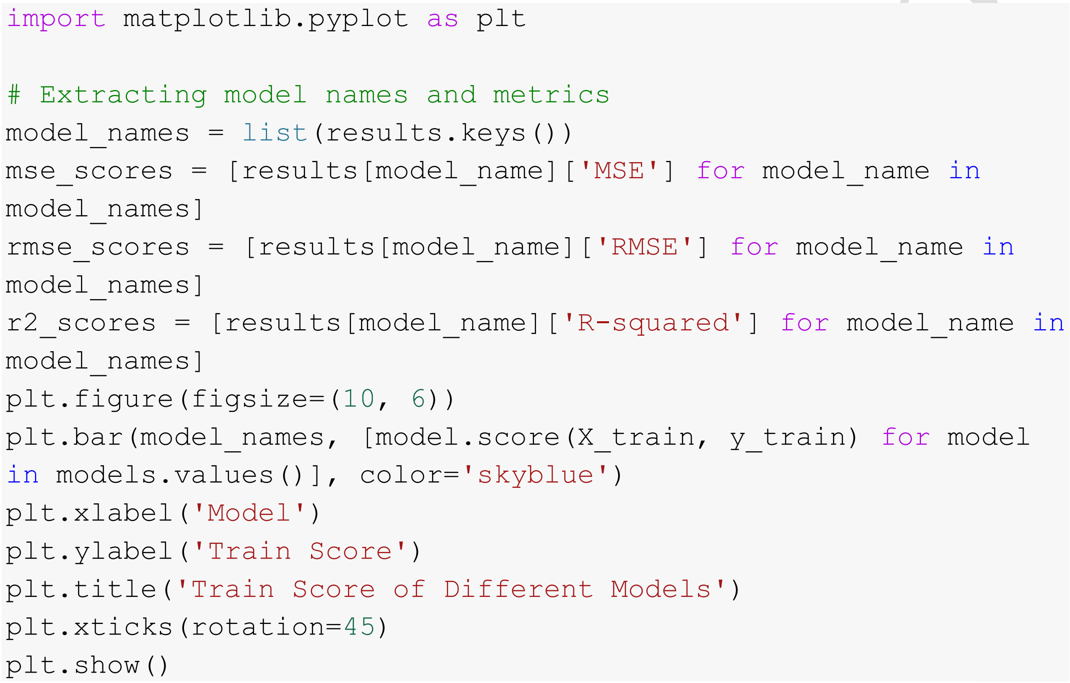
```
import matplotlib.pyplot as plt

# Assuming y_pred and y_test are the predicted and actual
variables, respectively
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, color='blue', label='Actual vs.
Predicted')
plt.plot([min(y_test), max(y_test)], [min(y_test),
max(y_test)], color='red', linestyle='--', label='Perfect Fit')
plt.title('Actual vs. Predicted')
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.legend()
plt.grid(True)
pl).show()
```



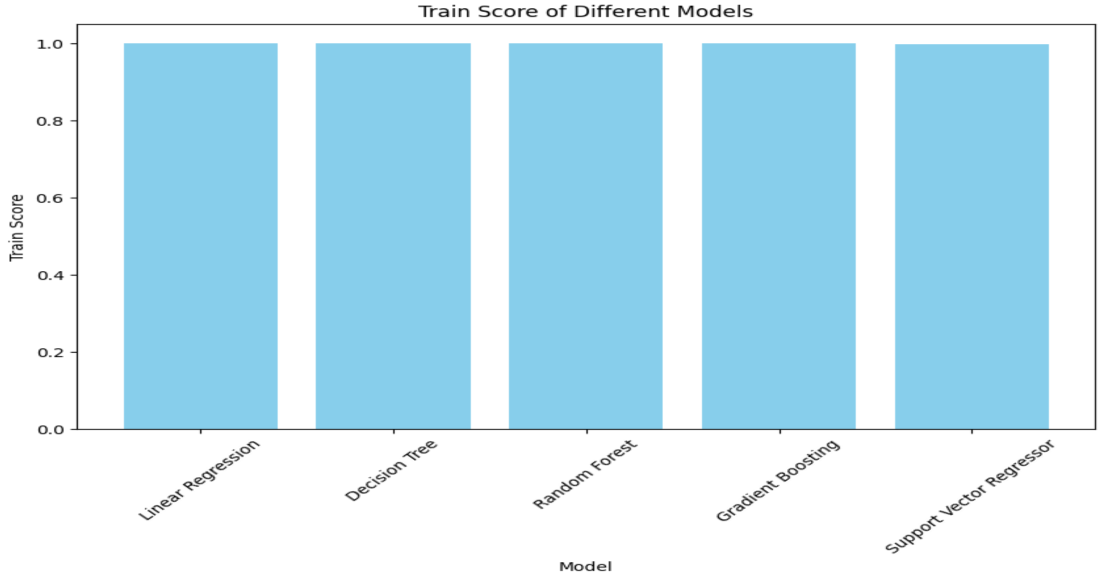
Bar plot of Train Score of Different Models:

- This plot shows the training scores of different machine learning models.

- The x-axis represents the names of the models, and the y-axis represents the training scores.
- The training score typically indicates how well each model fits the training data. A higher score suggests a better fit. However, a very high training score might indicate overfitting, where the model has memorized the training data and performs poorly on unseen data.



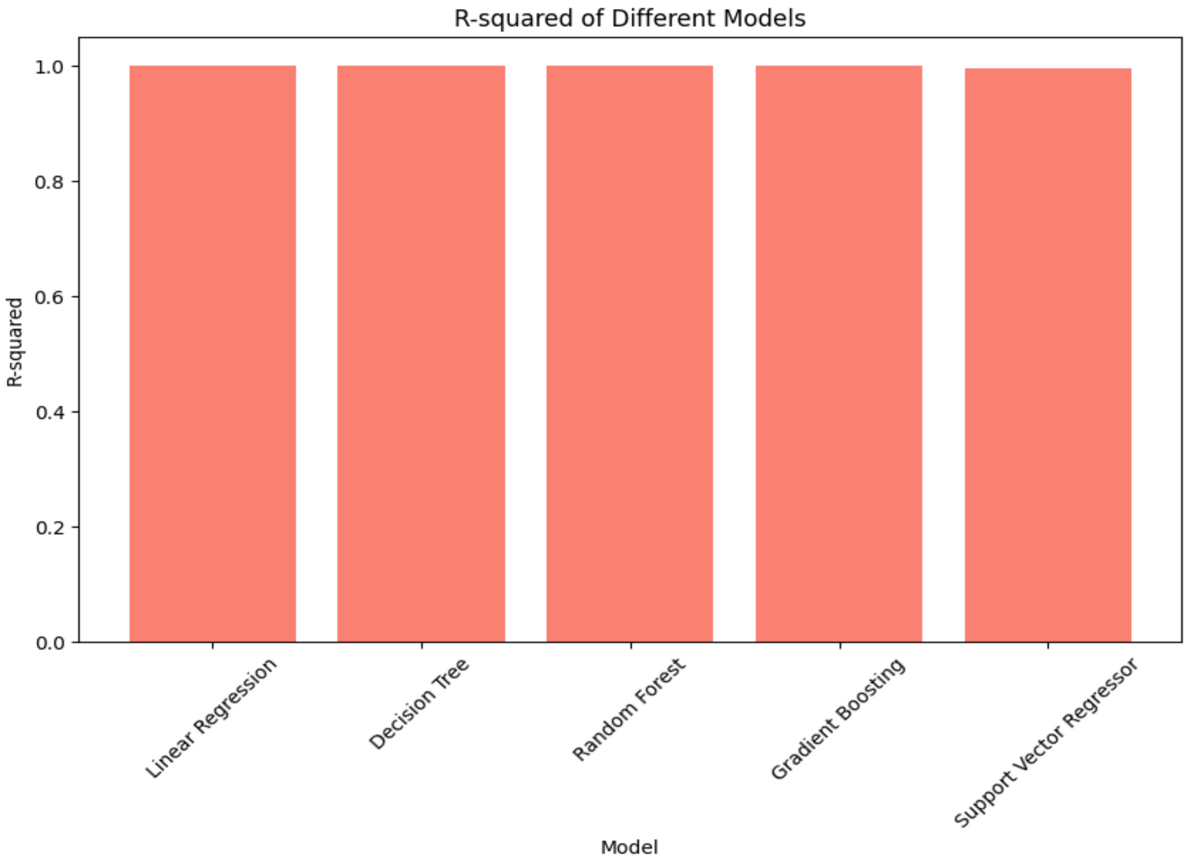
```
import matplotlib.pyplot as plt

# Extracting model names and metrics
model_names = list(results.keys())
mse_scores = [results[model_name]['MSE'] for model_name in
model_names]
rmse_scores = [results[model_name]['RMSE'] for model_name in
model_names]
r2_scores = [results[model_name]['R-squared'] for model_name in
model_names]
plt.figure(figsize=(10, 6))
plt.bar(model_names, [model.score(X_train, y_train) for model
in models.values()], color='skyblue')
plt.xlabel('Model')
plt.ylabel('Train Score')
plt.title('Train Score of Different Models')
plt.xticks(rotation=45)
plt.show()
```



Bar plot of R-squared of Different Models:

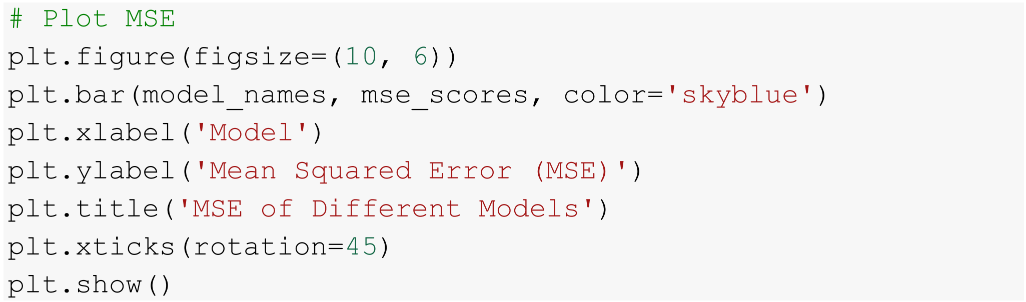
- R-squared (coefficient of determination) is a statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variables.
- This plot compares the R-squared values of different models.
- A higher R-squared value indicates that the model explains more of the variance in the target variable.

```
# Plot R-squared
plt.figure(figsize=(10, 6))
plt.bar(model_names, r2_scores, color='salmon')
plt.xlabel('Model')
plt.ylabel('R-squared')
plt.title('R-squared of Different Models')
plt.xticks(rotation=45)
plt.show()
```

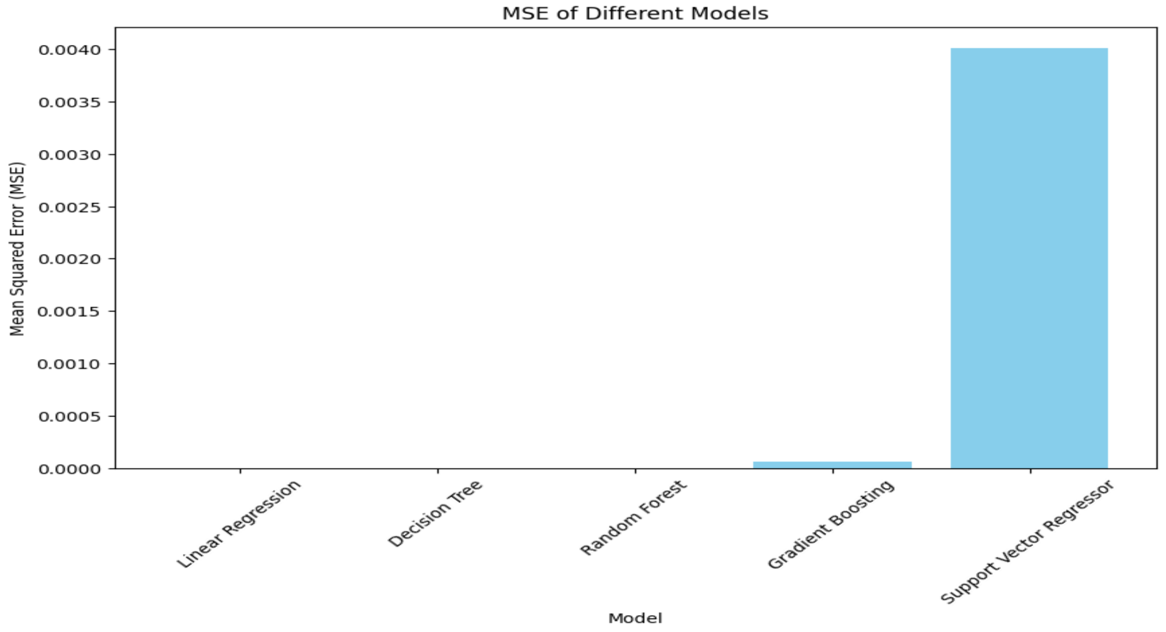


Bar plot of Mean Squared Error (MSE) of Different Models:

- MSE is a measure of the average squared difference between the actual and predicted values.
- This plot compares the MSE values of different models.
- Lower MSE values indicate better model performance, as they suggest that the model's predictions are closer to the actual values on average.



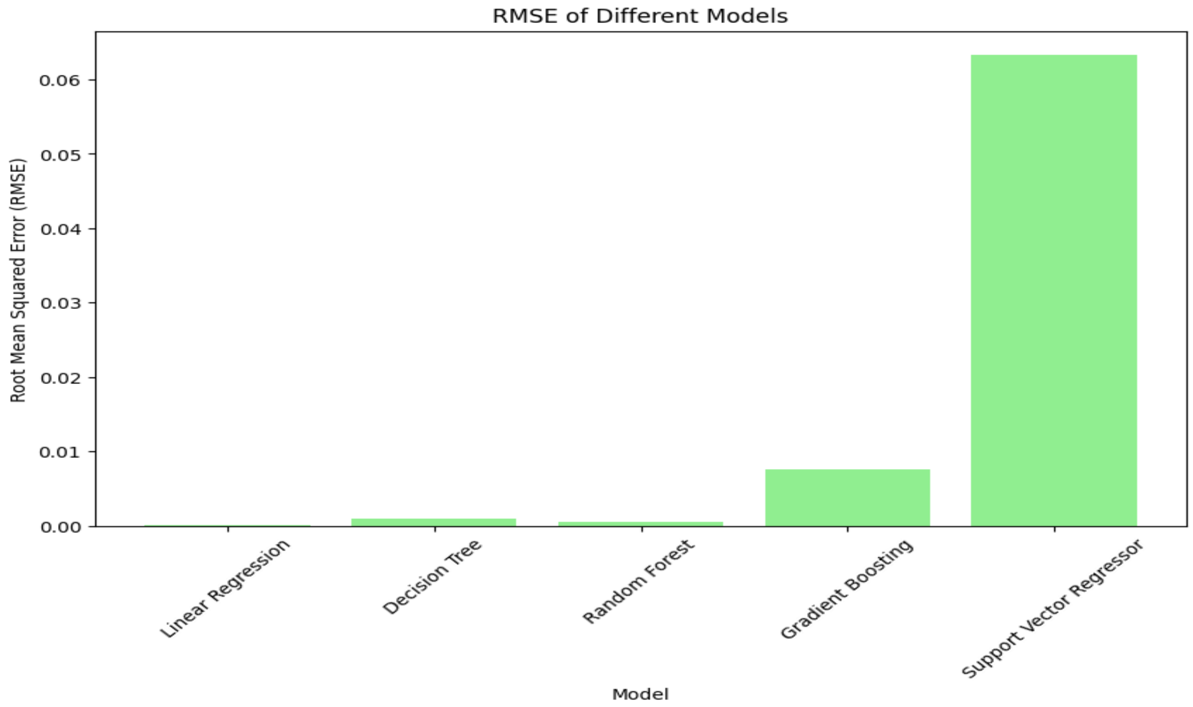
```
# Plot MSE
plt.figure(figsize=(10, 6))
plt.bar(model_names, mse_scores, color='skyblue')
plt.xlabel('Model')
plt.ylabel('Mean Squared Error (MSE)')
plt.title('MSE of Different Models')
plt.xticks(rotation=45)
plt.show()
```



Bar plot of Root Mean Squared Error (RMSE) of Different Models:

- RMSE is the square root of the MSE and represents the average distance between the predicted values and the actual values.
- This plot compares the RMSE values of different models.
- Like MSE, lower RMSE values indicate better model performance, as they suggest that the model's predictions are closer to the actual values on average.
- Each of these plots provides a different perspective on the performance of the regression models being evaluated, helping to assess their accuracy and effectiveness in predicting the target variable.

```
# Plot RMSE
plt.figure(figsize=(10, 6))
plt.bar(model_names, rmse_scores, color='lightgreen')
plt.xlabel('Model')
plt.ylabel('Root Mean Squared Error (RMSE)')
plt.title('RMSE of Different Models')
plt.xticks(rotation=45)
plt.show()
```



Conclusion

By leveraging visualizations at various stages of the machine learning pipeline, from data exploration to model evaluation and interpretation, we can effectively analyze and communicate the results of regression analysis, leading to more informed decision-making and improved model performance.

Gantt Chart

Sprints	Days Worked on Each Sprint	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Total Man-Days
Synthetic Data Generation	2	X	X													2
Data Cleaning	2			X	X											2
Feature Engineering	1					X										1
Model Training							X	X								2
Model Application and Iteration									X							1
Minimization of Bias and Overfitting										X						1
Test Models on Unseen Data or Validation Set											X	X				2
Final Model Selection and Conclusion													X			
Documentation														X	X	
Total Days		X	X	X	X	X	X	X	X	X	X	X				14