Machine Learning Model Report

Objective:

To build a machine learning model to predict the class of cars based on a set of features in the “cars_price.csv” dataset.

Code Flow:

1.Import Required Libraries:

Required libraries like pandas, sklearn were imported.

2.Data Exploration and Preprocessing:

Data Loading: The dataset was loaded into a Pandas DataFrame named “cars_price”. Feature and Target Variable Identification: Features (X) and the target variable (y) were identified. The target variable is 'Class,' and the features are all other columns.

Handling Missing Data: Handle missing values by replacing '?' with NaN and dropping rows with NaN.

Data Type Conversion: Selected columns ('normalized-losses', 'bore', 'stroke', 'horsepower', 'peak-rpm', 'price') were converted to appropriate numerical data types.

3.Data Transformation:

One Hot Encoding: Categorical variables were One-Hot encoded to convert them into a format suitable for Machine Learning Models.

4.Data Splitting:

Separate Features and Target Variables: The dataset was split into features (X) and the target variable (y) where 'price' is the target variable.

Target Train-Test Split: The data was divided into training and testing sets with a test size of 20% for model evaluation.

5.Data Standardization:

Feature Scaling: StandardScaler from scikit-learn was applied to standardize the feature values. Standardization ensures that all features have a mean of 0 and a standard deviation of 1.

6.Model Selection and Training:

Model Initialization: A Random Forest Regressor was chosen for its ability to handle non-linear relationships and capture feature importance.

Model Training: The Random Forest Regressor was trained on the scaled training data

7.Model Evaluation:

Model Prediction: The trained model was used to predict car prices on the test set.

Regression Metrics: Model performance was assessed using mean squared error (MSE), mean absolute error (MAE), and R-squared (R^2).

These metrics provide insights into how well the model predicts numerical values.

OUTPUT:

```
Mean Squared Error: 2951426.2754267366
Mean Absolute Error: 1294.6430208333336
R-squared: 0.8341394978343453
```

Conclusion:

In conclusion, the evaluation metrics, including Mean Squared Error, Mean Absolute Error, and R-squared, highlight the model's effectiveness in predicting car prices. These results provide a great insight for further refinement of the model.