

## SUMMARY 4

### 1. Import Libraries:

- It imports the necessary libraries: NumPy, Matplotlib, and Pandas.

### 2. Load Dataset:

- It loads a dataset from a CSV file named 'Premotorbike.csv' into a Pandas DataFrame called `dataset`.

### 3. Data Preprocessing:

- It creates a new DataFrame `df2` and assigns it the same data as `dataset`.
- It applies one-hot encoding to categorical variables in `df2` using `pd.get_dummies()`. The `drop_first=True` argument is used to drop the first category for each categorical variable, avoiding multicollinearity.
- It separates the independent variables (`independent`) and the dependent variable (`dependent`) from `df2`.

### 4. Train-Test Split:

- It splits the data into training and testing sets using `train_test_split` from scikit-learn (`sklearn`). The training set is assigned to `X_train` and `Y_train`, while the testing set is assigned to `X_test` and `Y_test`.

### 5. Random Forest Regression:

- It imports the `RandomForestRegressor` from scikit-learn.
- It initializes a random forest regressor (`regressor`) with 50 decision trees (`n_estimators=50`) and a random seed (`random_state=0`).
- It fits the regressor to the training data using `fit()`.

### 6. Predictions:

- It uses the trained regressor to make predictions (`Y_pred`) on the testing data (`X_test`).

### 7. Model Evaluation:

- It calculates the R-squared score (`r_score`) as a measure of how well the model fits the data.

### 8. Model Serialization:

- It uses the `pickle` library to save the trained regressor (`regressor`) to a file named "finalised\_model\_motorbike.sav" for later use.

### 9. Loading the Model:

## SUMMARY 4

- It loads the saved model from the file "finalised\_model\_motorbike.sav" using `pickle`.

### 10. Making a Prediction with the Loaded Model:

- It uses the loaded model to make a prediction (`result`) on a new set of input features (e.g., `[1565, 253, 456, 567, 897, 125, 1, 0]`).

In summary, this code loads a motorcycle dataset, preprocesses the data, trains a random forest regression model, evaluates its performance, saves the model to a file, loads the model from the file, and makes a prediction using the loaded model. The R-squared score (`r_score`) is used as a metric to assess the model's goodness of fit.

## DEPLOYMENT PHASE 2:

### 1. Importing the `pickle` Module:

- It imports the `pickle` module, which is used for serializing and deserializing Python objects.

### 2. Loading a Saved Machine Learning Model:

- It loads a previously saved machine learning model from a file named "finalised\_model\_motorbike.sav" using `pickle.load()`. This model is assumed to have been saved using the `pickle` module and contains the necessary information to make predictions.

### 3. Making a Prediction:

- After loading the model, it uses the loaded model (`loaded_model`) to make a prediction on a set of input features (`[1565, 253, 456, 567, 897, 125, 1, 0]`).
- The prediction result is stored in the variable `result`.

## SUMMARY 4

In summary, here loading machine learning model from a file using `pickle`, and then it uses that model to make a prediction on a specific set of input features. The prediction result is stored in the `result` variable, which can be further used or analyzed as needed.