|  |  |
| --- | --- |
| Name Of The Student | Himanshu |
| Internship Project Topic | TCS iON RIO-210: Build a Classification Model for Drug Trials Dataset |
| Name of the Organization | TCS iON |
| Name of the Industry Mentor | Himdweep Walia |
| Name of the Institute | Amity University |

|  |  |  |
| --- | --- | --- |
| Date | Day # | Hours Spent |
| 04-06-2024 | Day-42 | 6 Hours |
| Activities done during the day:  **Project Hands-on – Feature Selection**  **Link of the google drive google Colab file :-**  <https://colab.research.google.com/drive/1VQRq0l6oc9Uj4cOOqiuhkfS1JmpKr3fU?usp=sharing>  **Feature Selection:**  Feature selection is a critical step in machine learning where you choose the most relevant features (variables) to include in your model from the original set of features. This process helps to improve model performance, reduce overfitting, and enhance interpretability.  Feature selection using a Random Forest classifier involves utilizing the built-in feature importance attribute of the Random Forest algorithm to rank the importance of each feature in predicting the target variable. Here's a general outline of how you can perform feature selection using Random Forest:   * Train a Random Forest Model * Retrieve Feature Importance * Select Features * Rebuild Model with Selected Features * Evaluate Model Performance   **Code:**  from sklearn.feature\_selection import SelectFromModel  from sklearn.ensemble import RandomForestClassifier  # Create a Random Forest classifier  model = RandomForestClassifier(random\_state=42)  # Fit the model to the training data  model.fit(X\_train, y\_train)  # Perform feature selection  feature\_selector = SelectFromModel(model, threshold='median')  feature\_selector.fit(X\_train, y\_train)  # Get the selected feature indices  selected\_feature\_indices = feature\_selector.get\_support(indices=True)  # Get the selected feature names  selected\_feature\_names = X\_train.columns[selected\_feature\_indices]  # Select the features in the training and testing data  X\_train\_selected = feature\_selector.transform(X\_train)  X\_test\_selected = feature\_selector.transform(X\_test)  **Check what feature is selected by the model**  print("Selected Features:")  for feature\_name in selected\_feature\_names:      print(feature\_name) | | |
|  | | |