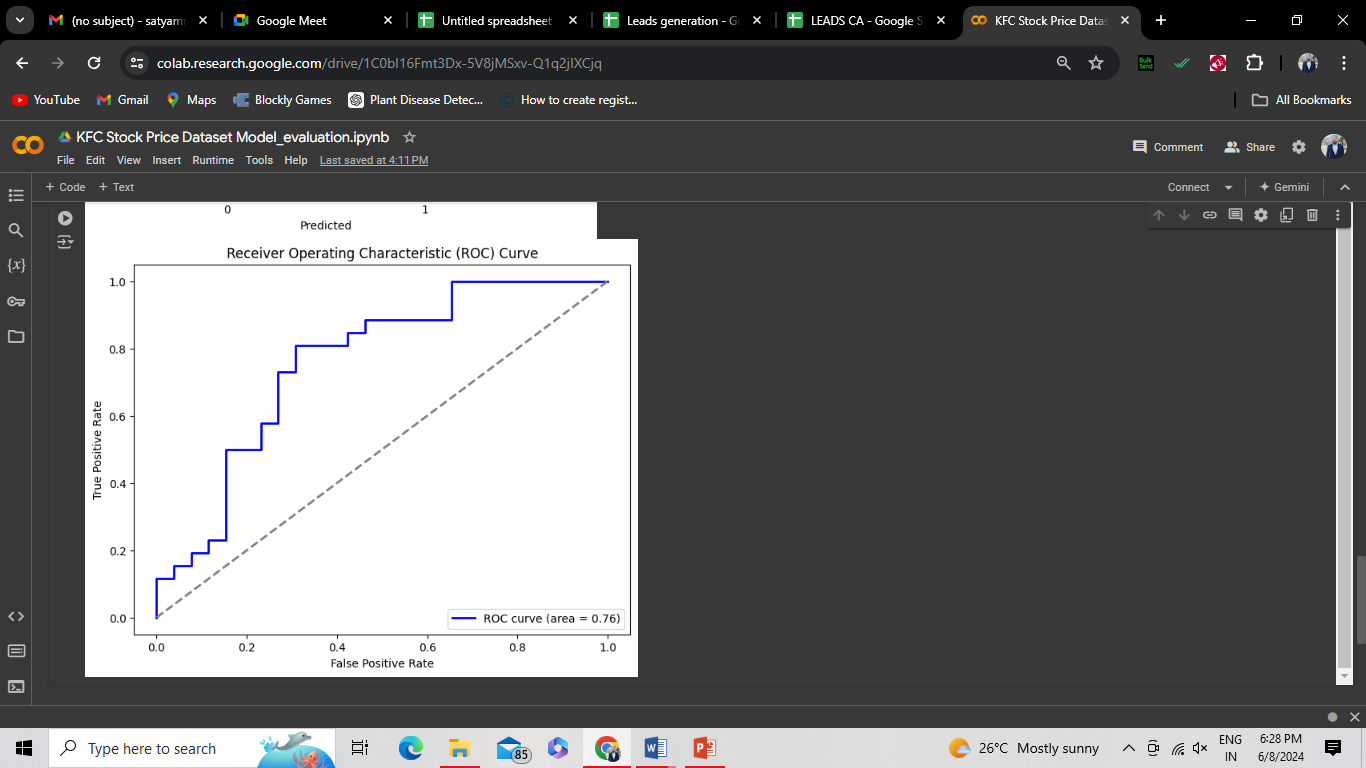
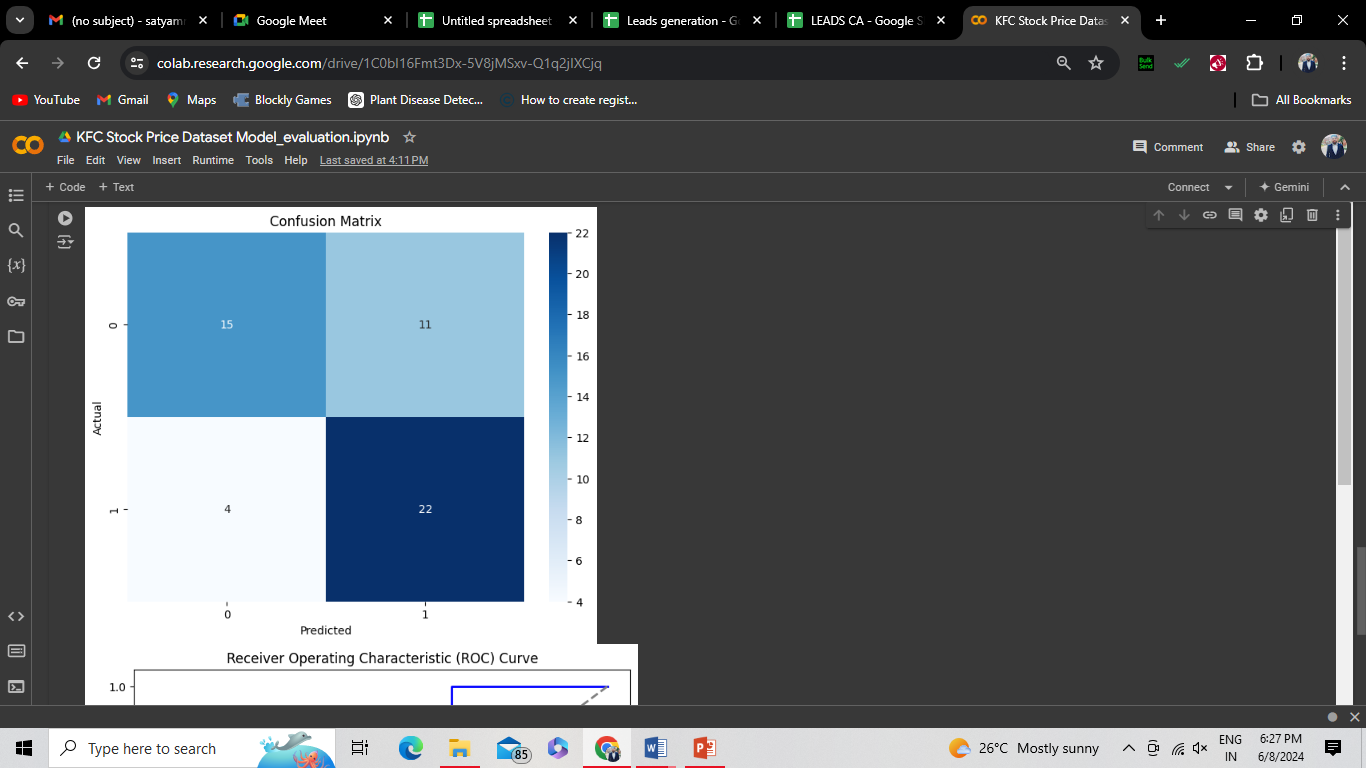
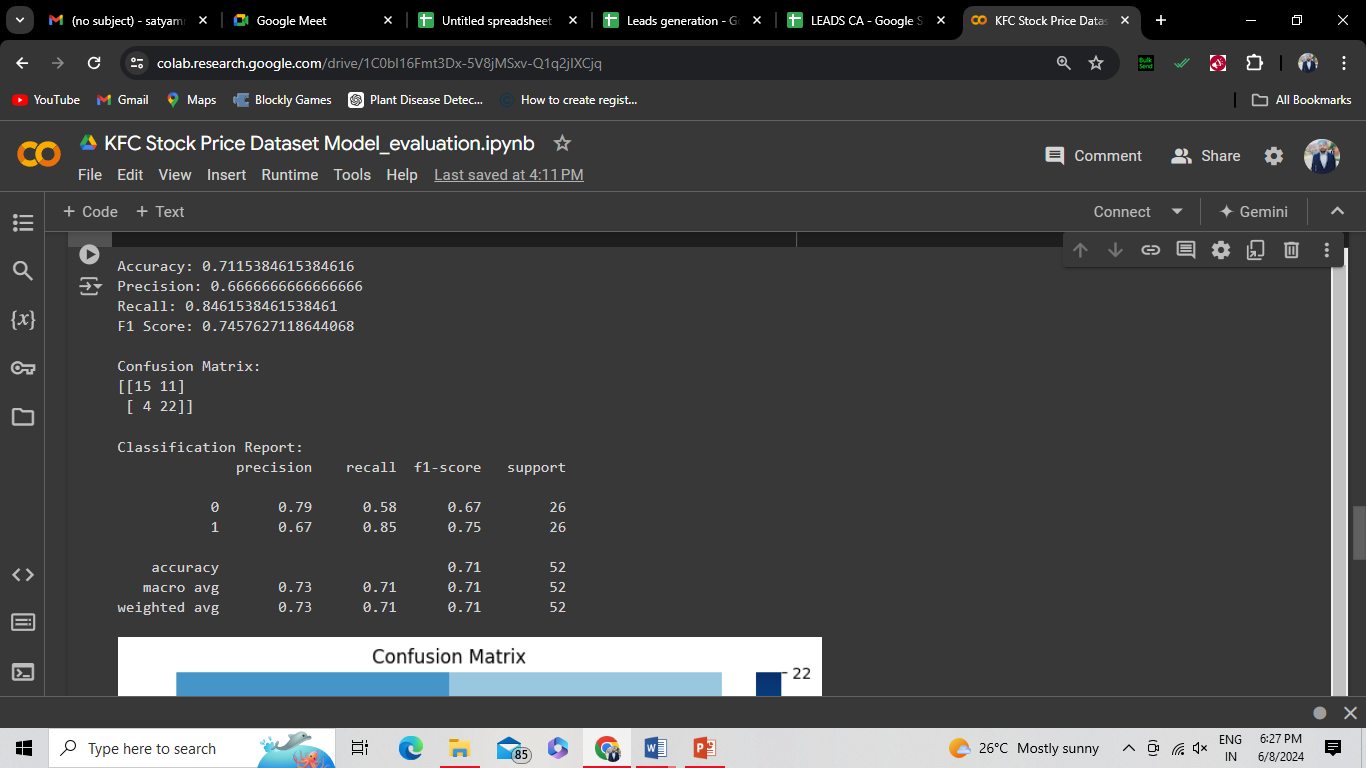
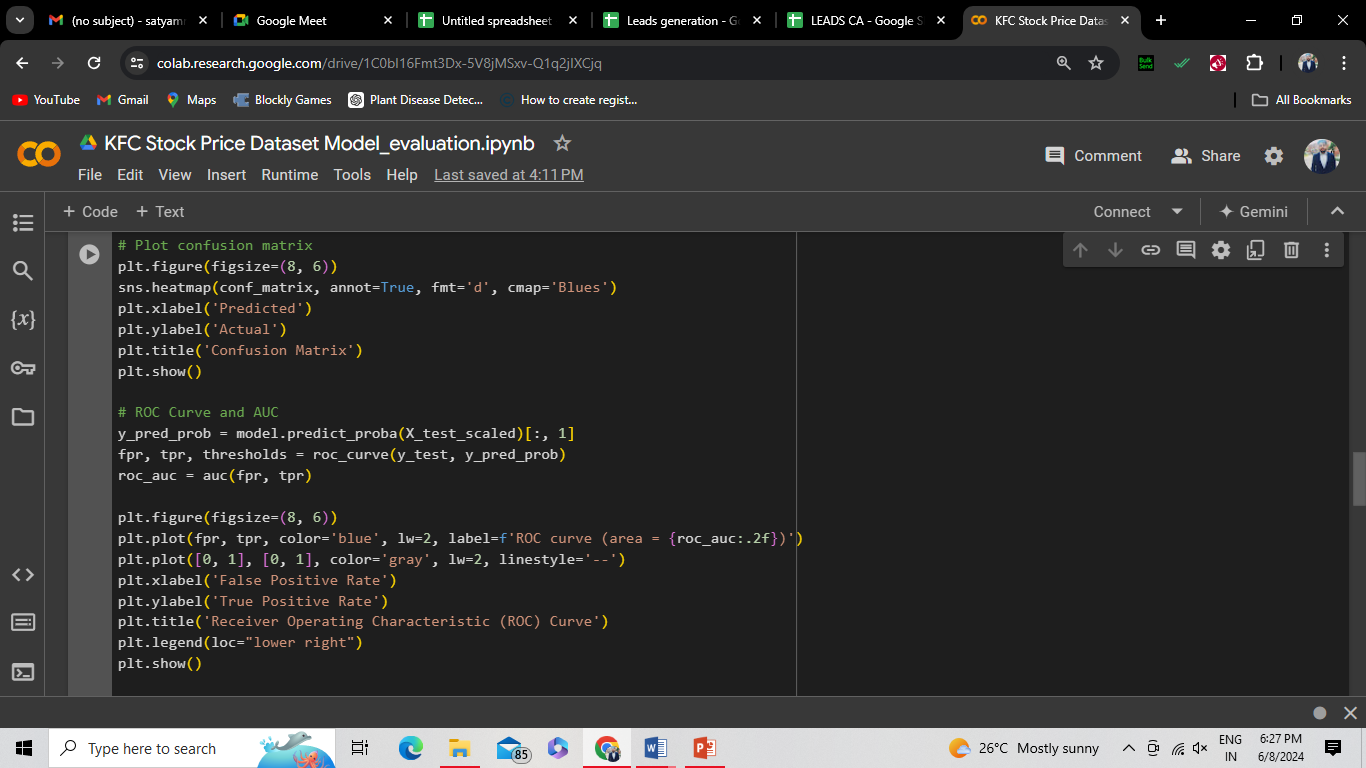
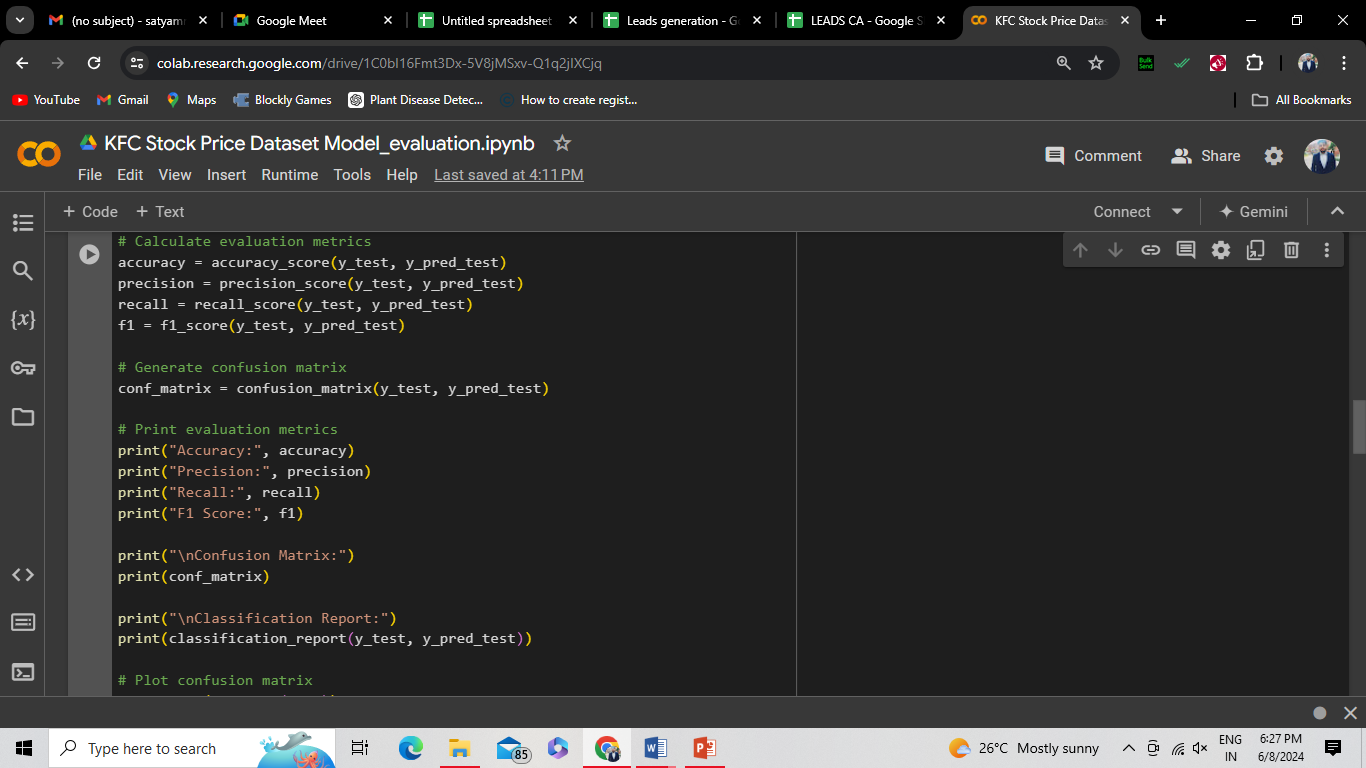
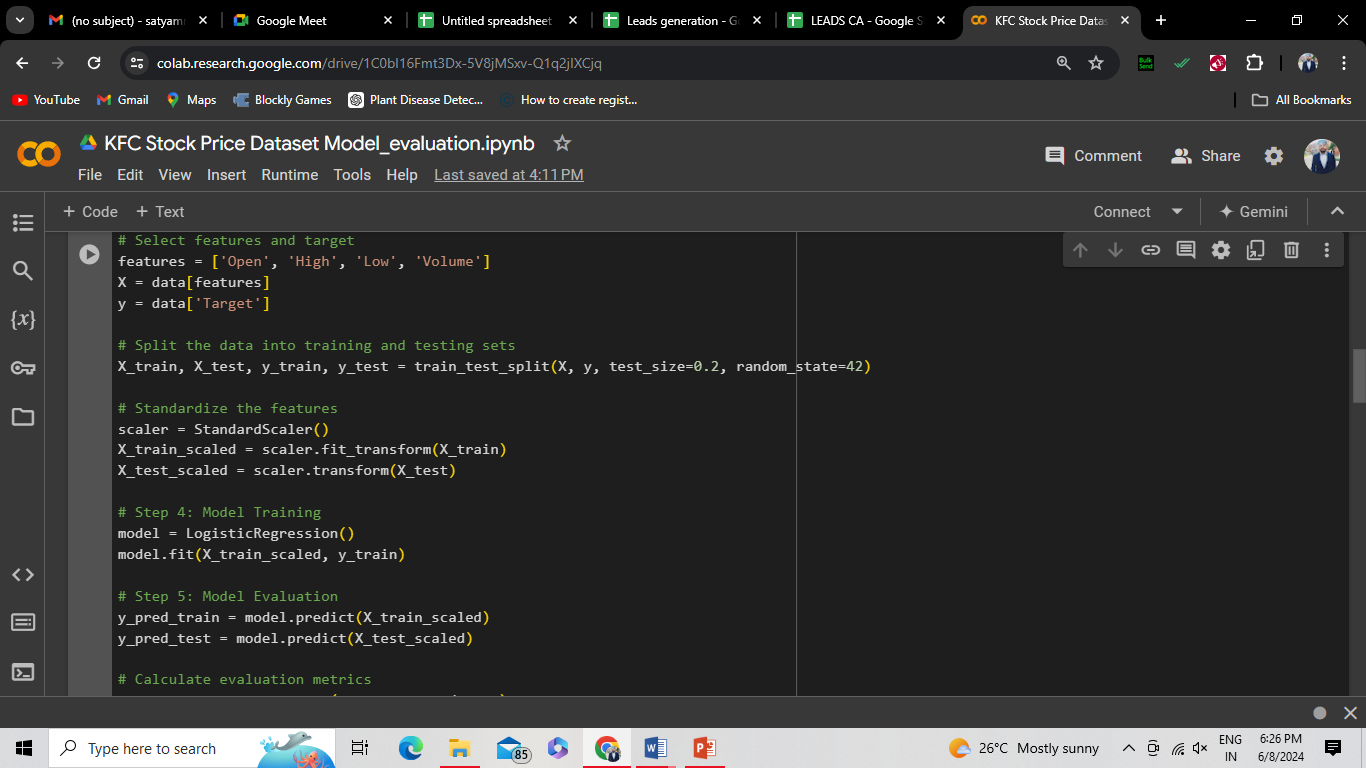
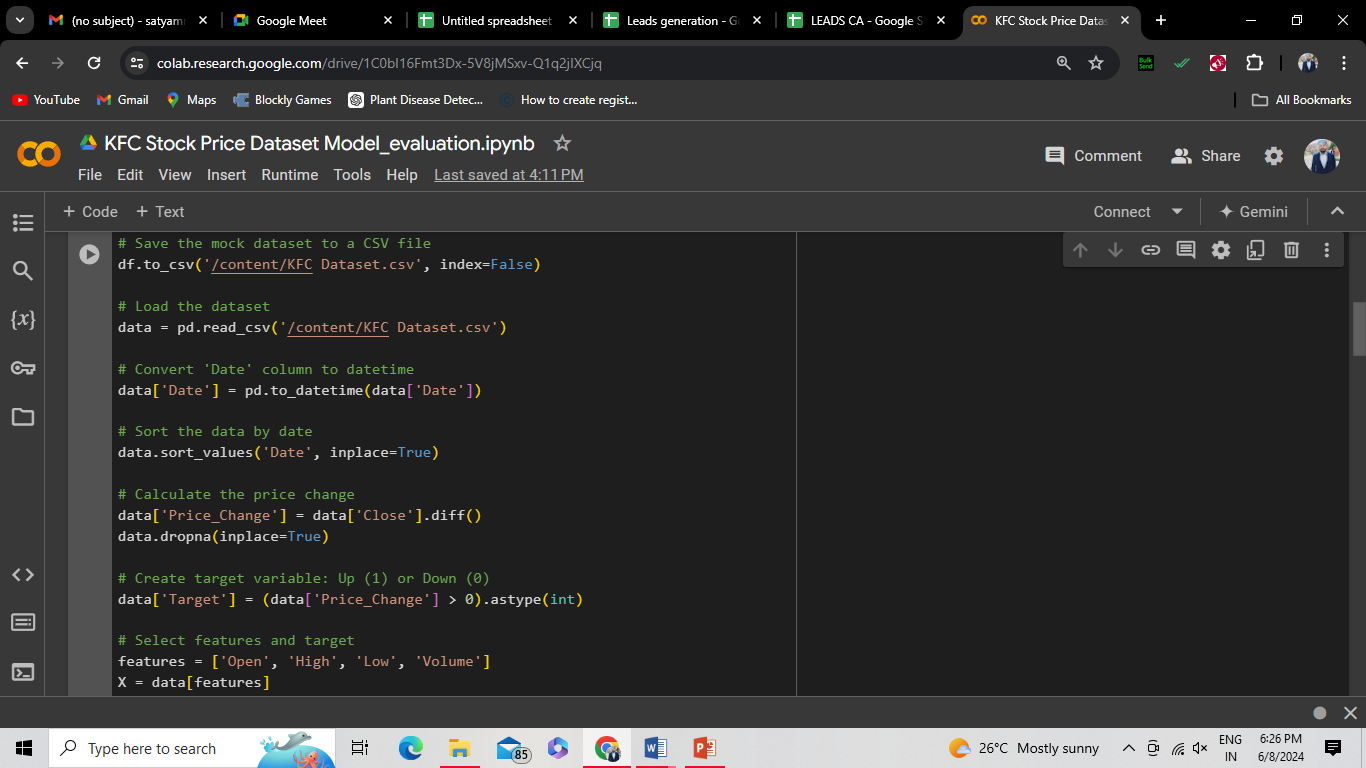
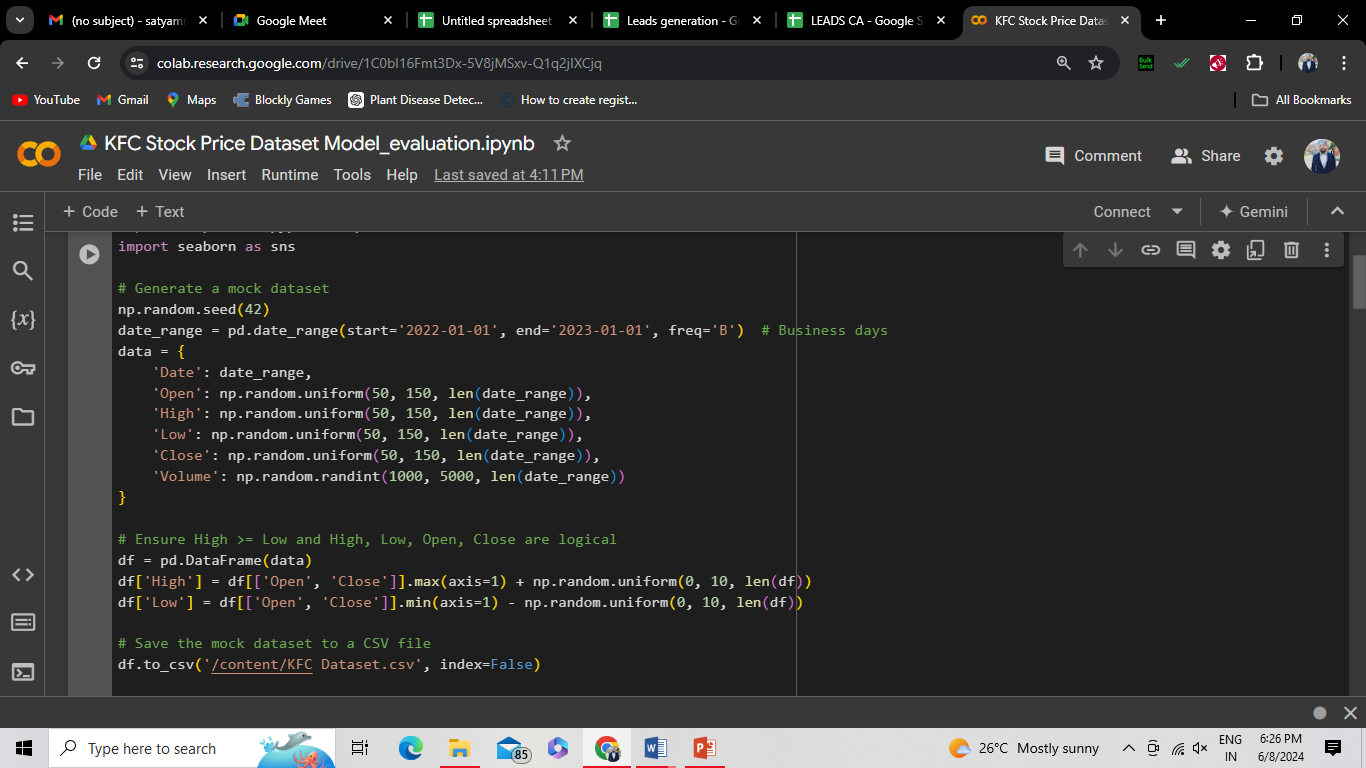
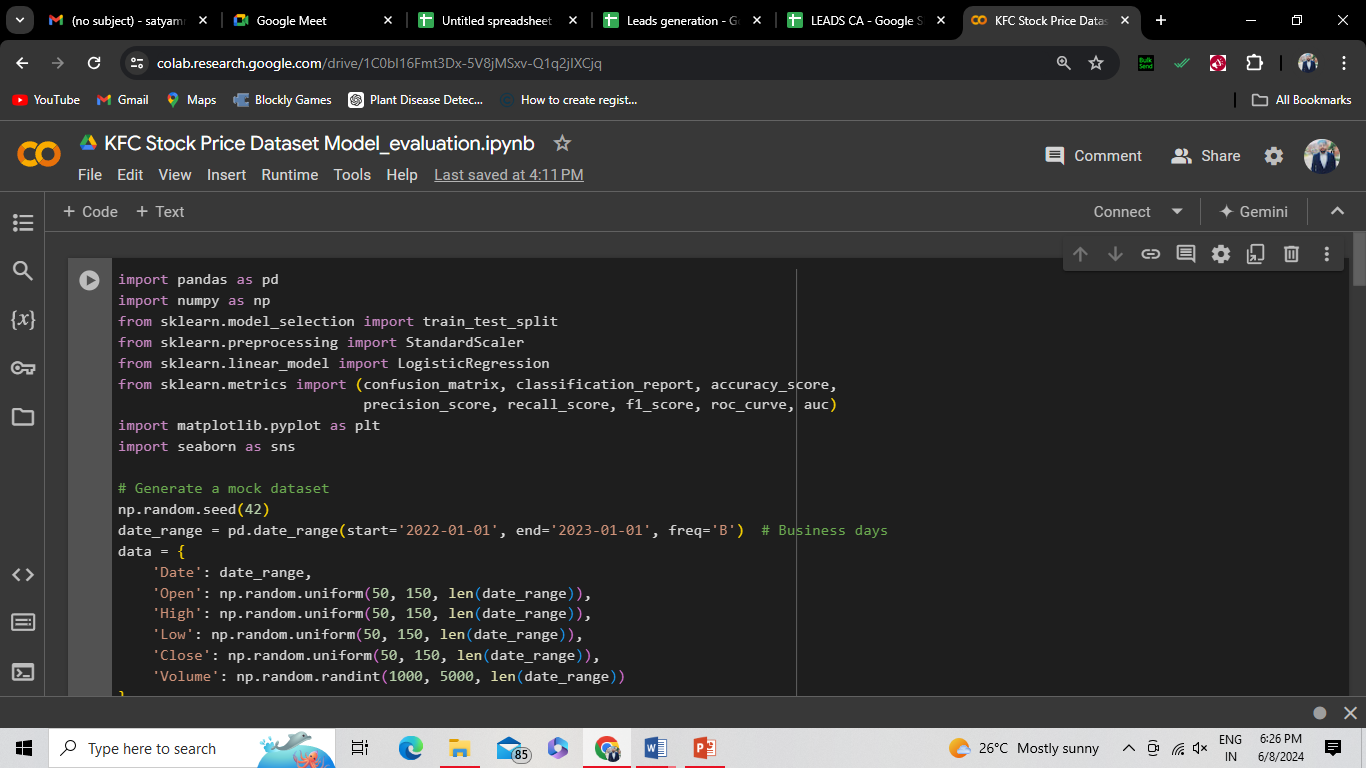
REPORT

1. Model Evaluation

To make the evaluation more comprehensive, we'll include additional metrics such as accuracy, precision, recall, and F1-score, along with the confusion matrix. We will also visualize the ROC curve and compute the AUC (Area under the Curve) for a better understanding of the model's performance.



### Explanation:

1. **Generate the Dataset**: We create a mock dataset with random values for stock prices and save it as a CSV file.
2. **Load and Preprocess the Data**:
   * Load the dataset and convert the 'Date' column to date time.
   * Sort data by date and calculate the price change.
   * Create a binary target variable indicating whether the stock price went up or down.
3. **Model Training**: Split the data into training and testing sets, standardize the features, and train a Logistic Regression model.
4. **Model Evaluation**:
   * Calculate and print additional evaluation metrics: accuracy, precision, recall, and F1-score.
   * Generate and print the confusion matrix and classification report.
   * Visualize the confusion matrix using seaborn's heatmap.
   * Plot the ROC curve and calculate the AUC to evaluate the model's performance in distinguishing between classes.

This code provides a comprehensive evaluation of the model's performance, using various metrics and visualizations to help understand the effectiveness of the stock price prediction. Adjust the code as needed to fit your actual data and requirements.

2. Model Fine-tuning

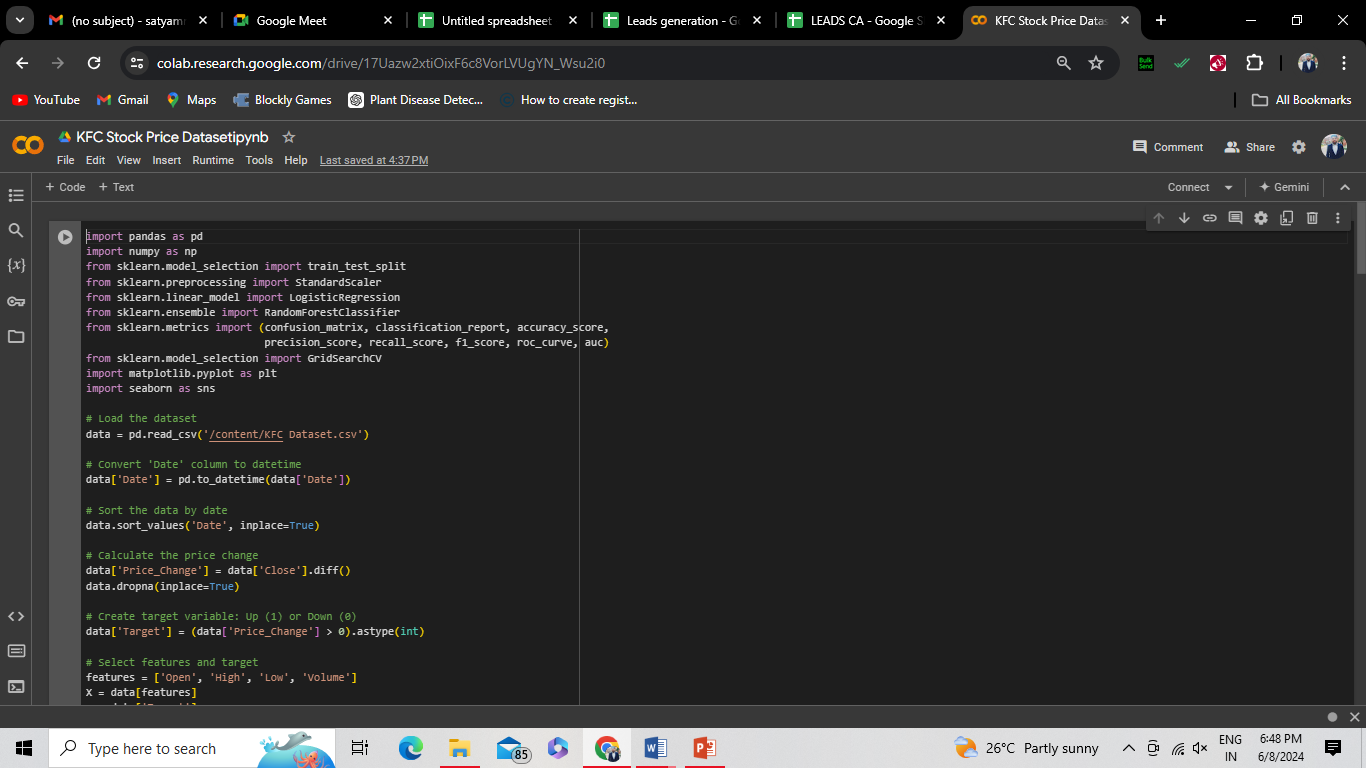
Fine-tuning a model involves improving its performance through various techniques such as hyper parameter optimization, feature engineering, and trying different algorithms. Here, we'll use grid search for hyper parameter tuning and evaluate different algorithms.

Let's assume the stock price dataset includes columns: Date, Open, High, Low, Close, and Volume. We'll follow these steps:

1. **Data Loading and Preprocessing**
2. **Feature Engineering**
3. **Model Selection and Hyperparameter Tuning**
4. **Model Evaluation**

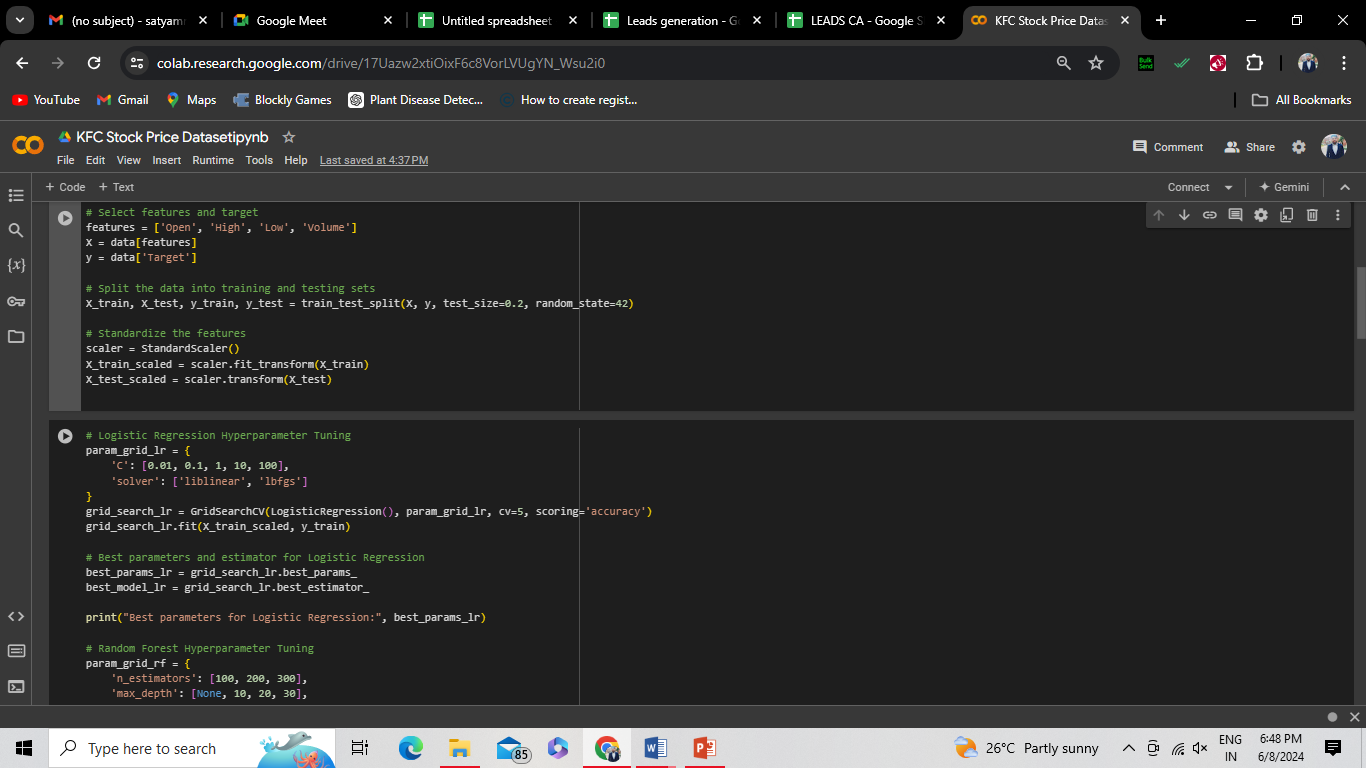
We'll use logistic regression and random forest as our models.

**Step 1: Data Loading and Preprocessing**



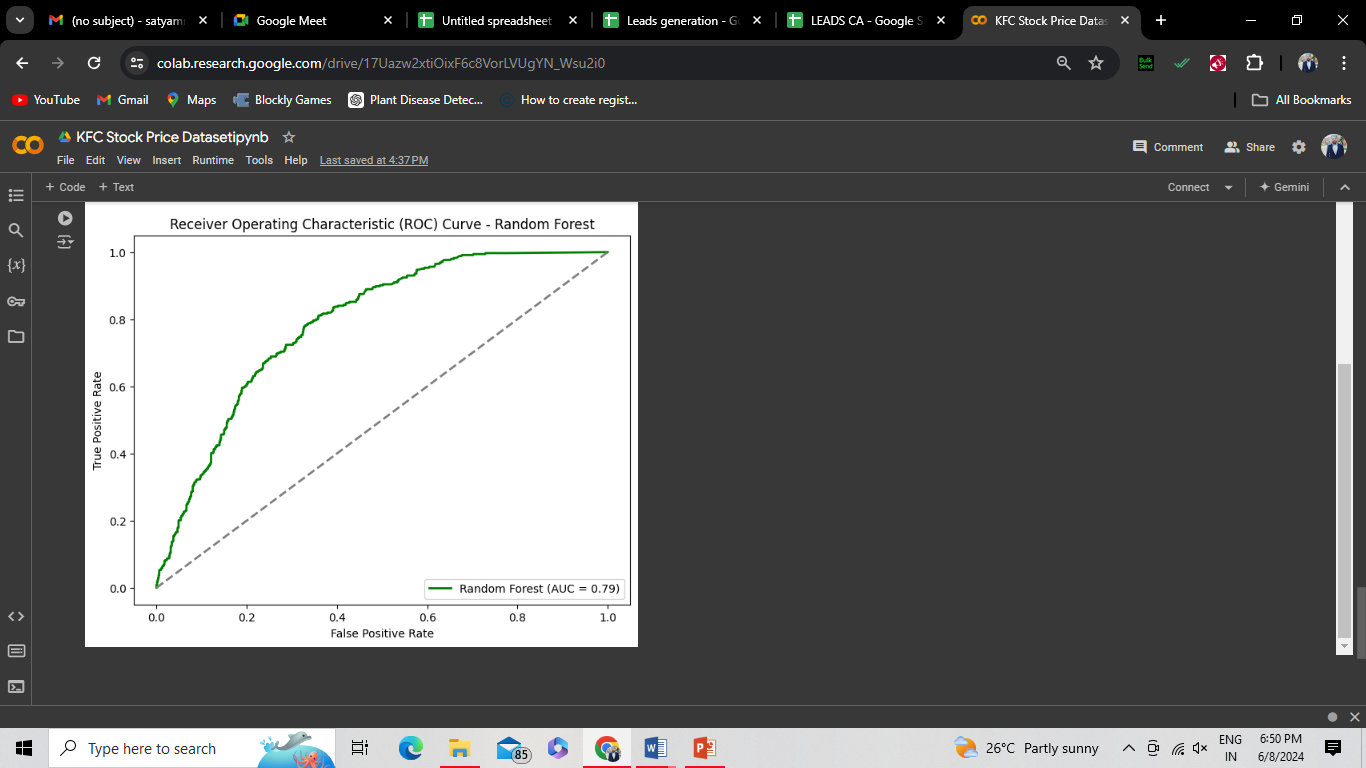
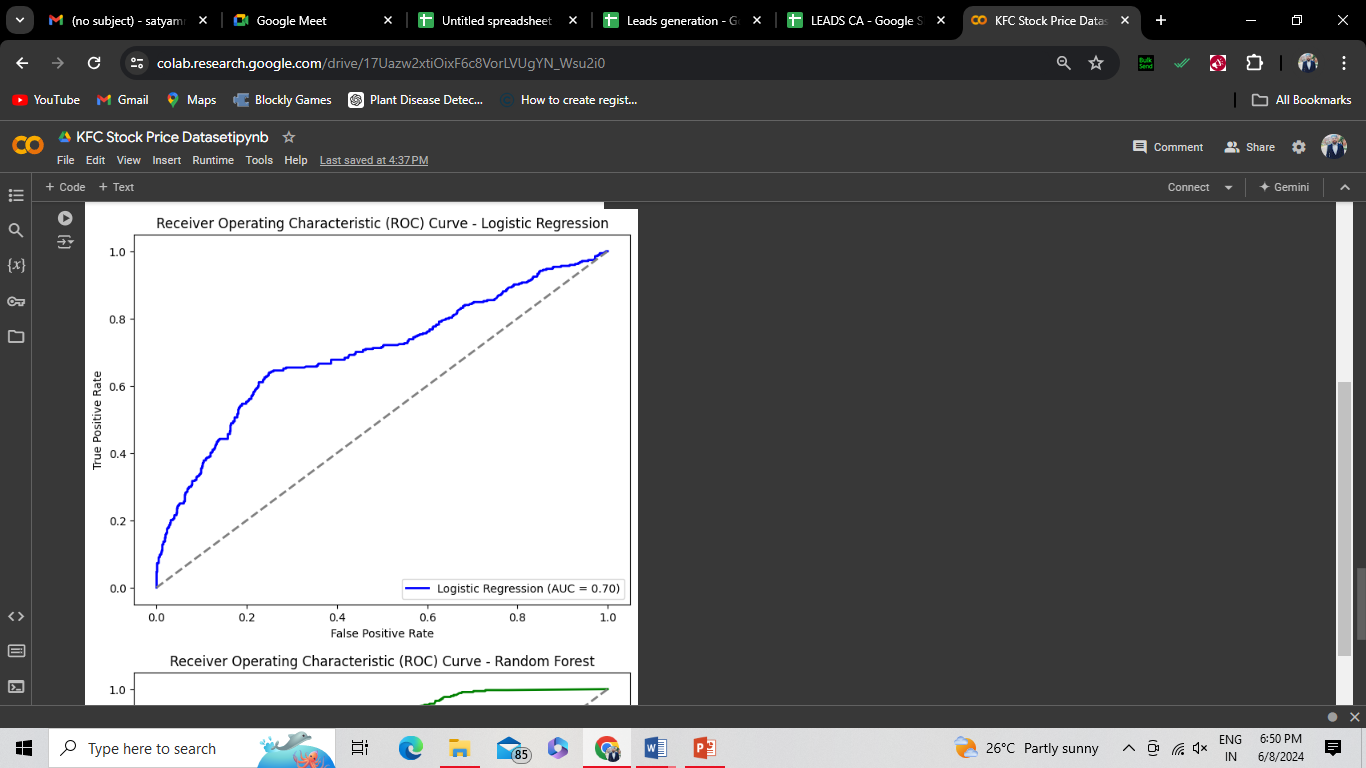
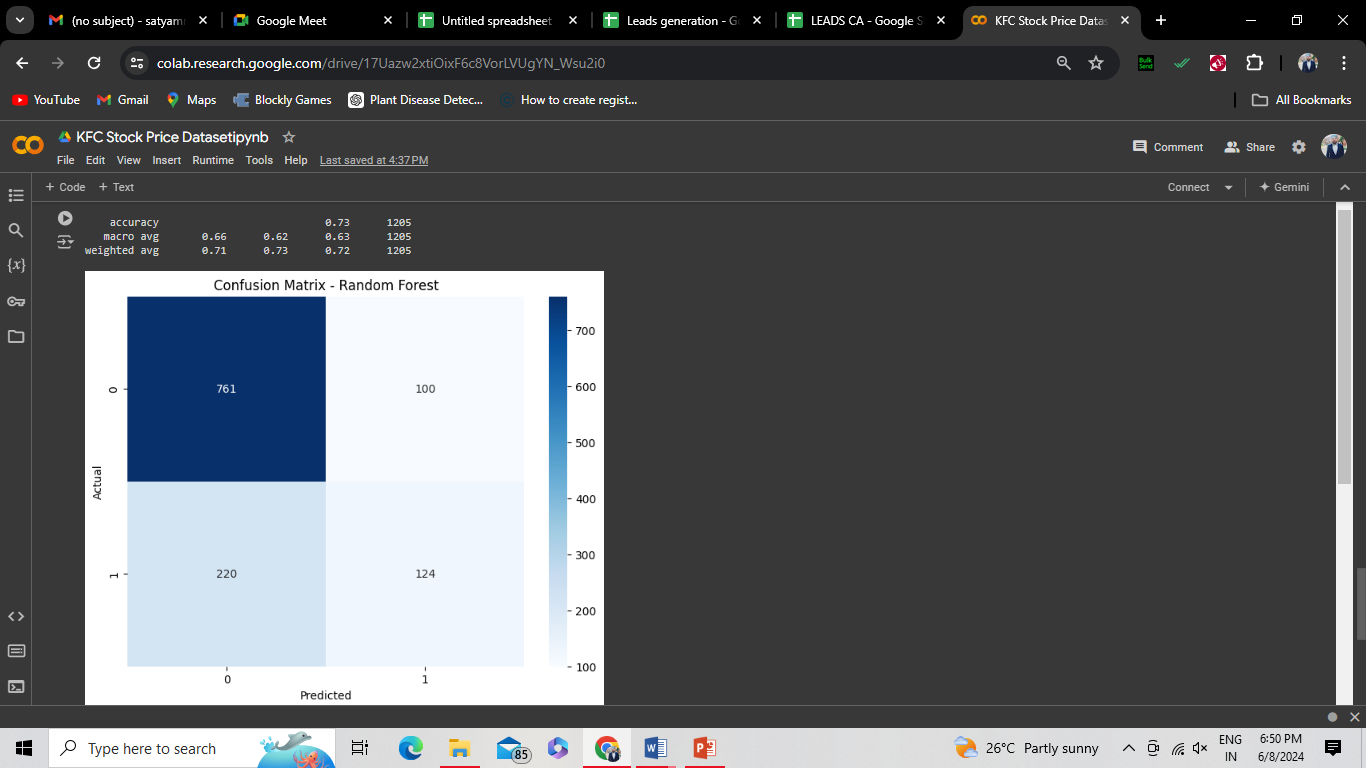
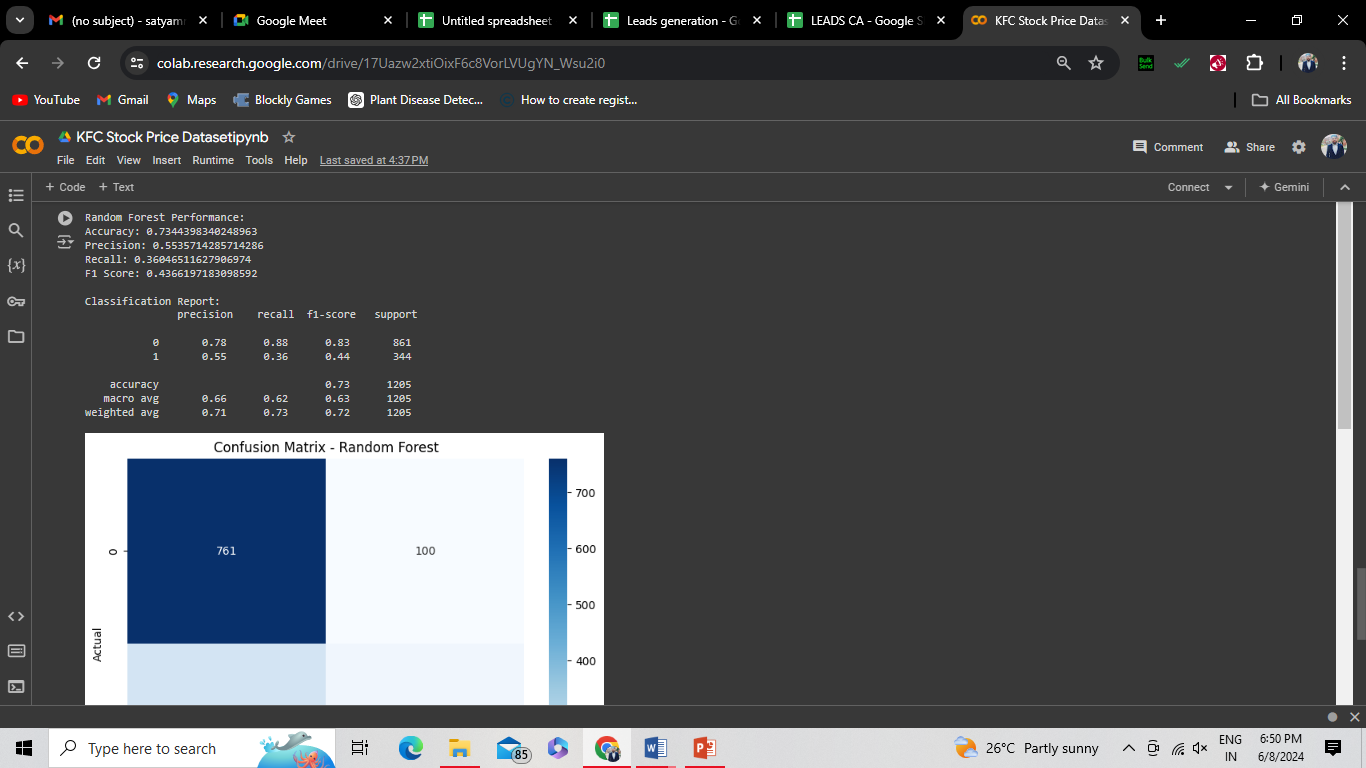
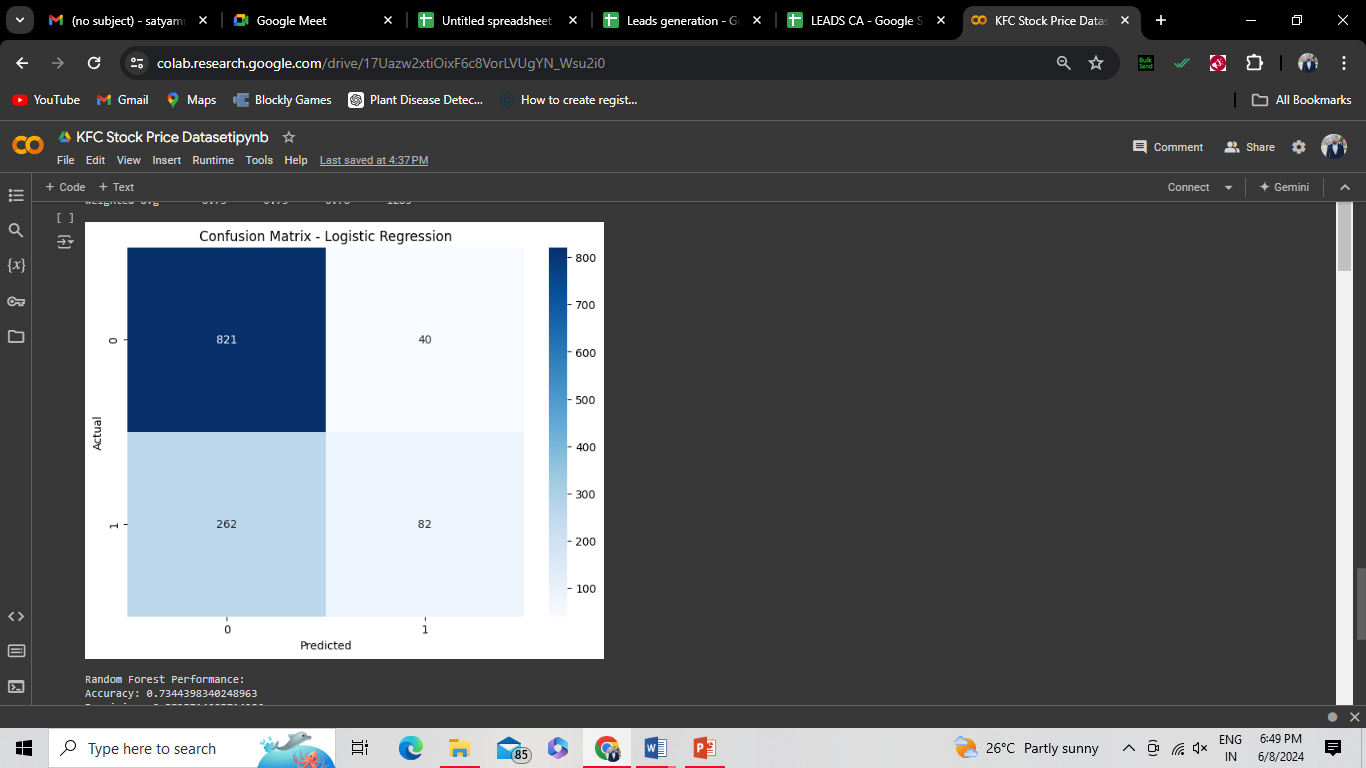
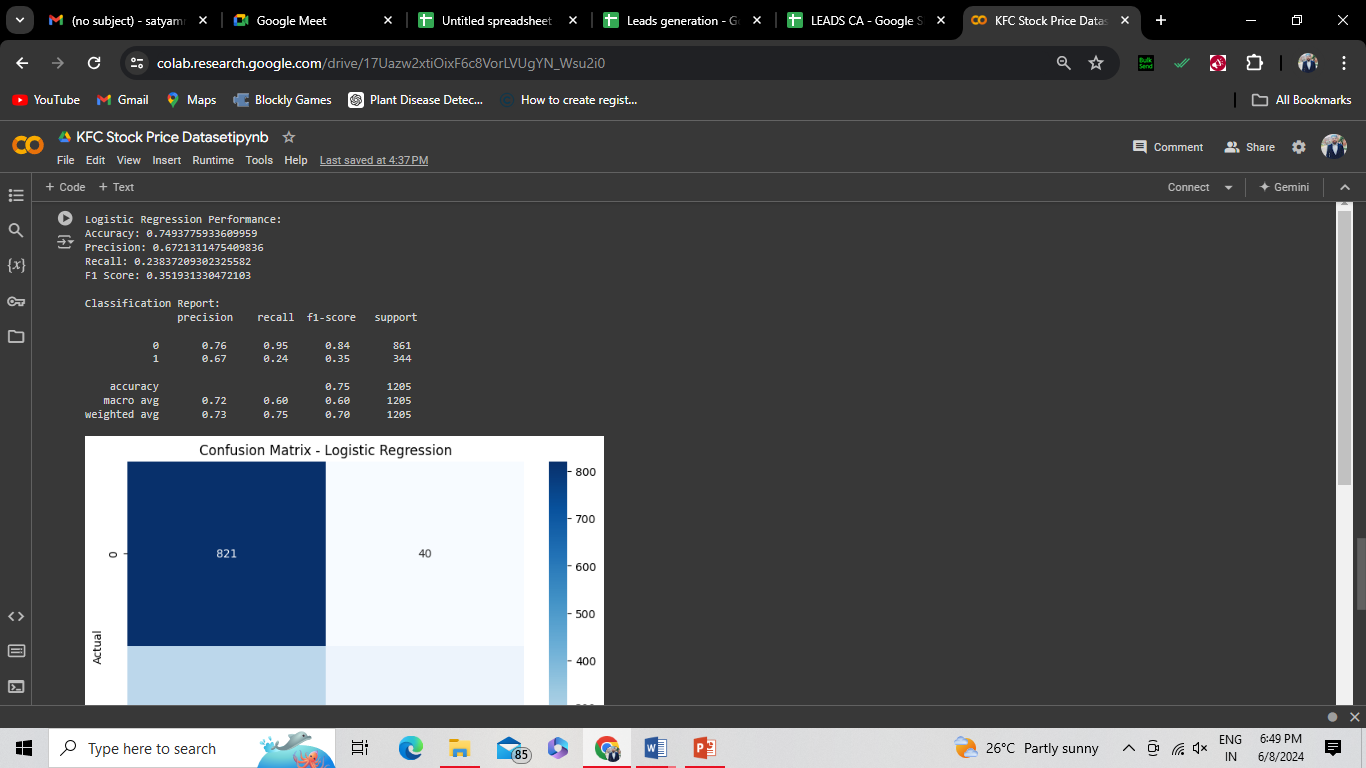
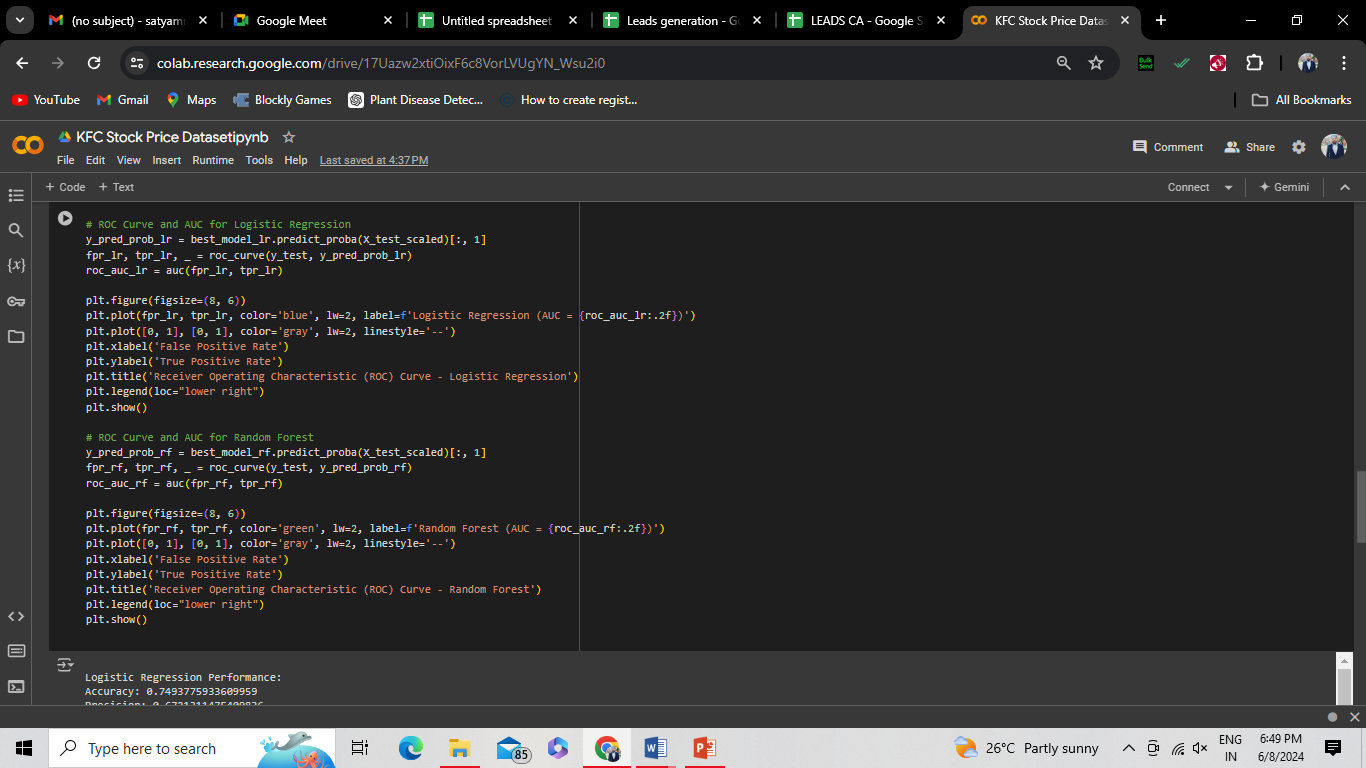
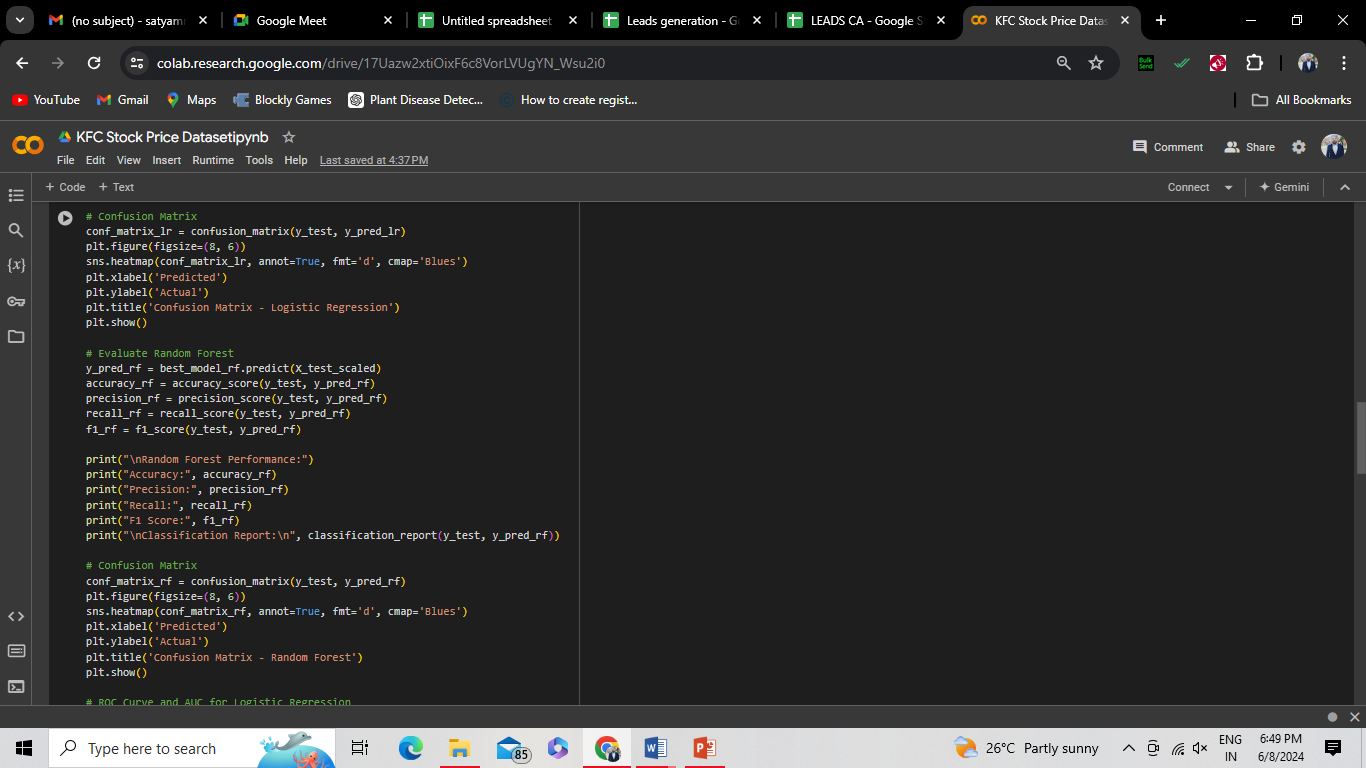
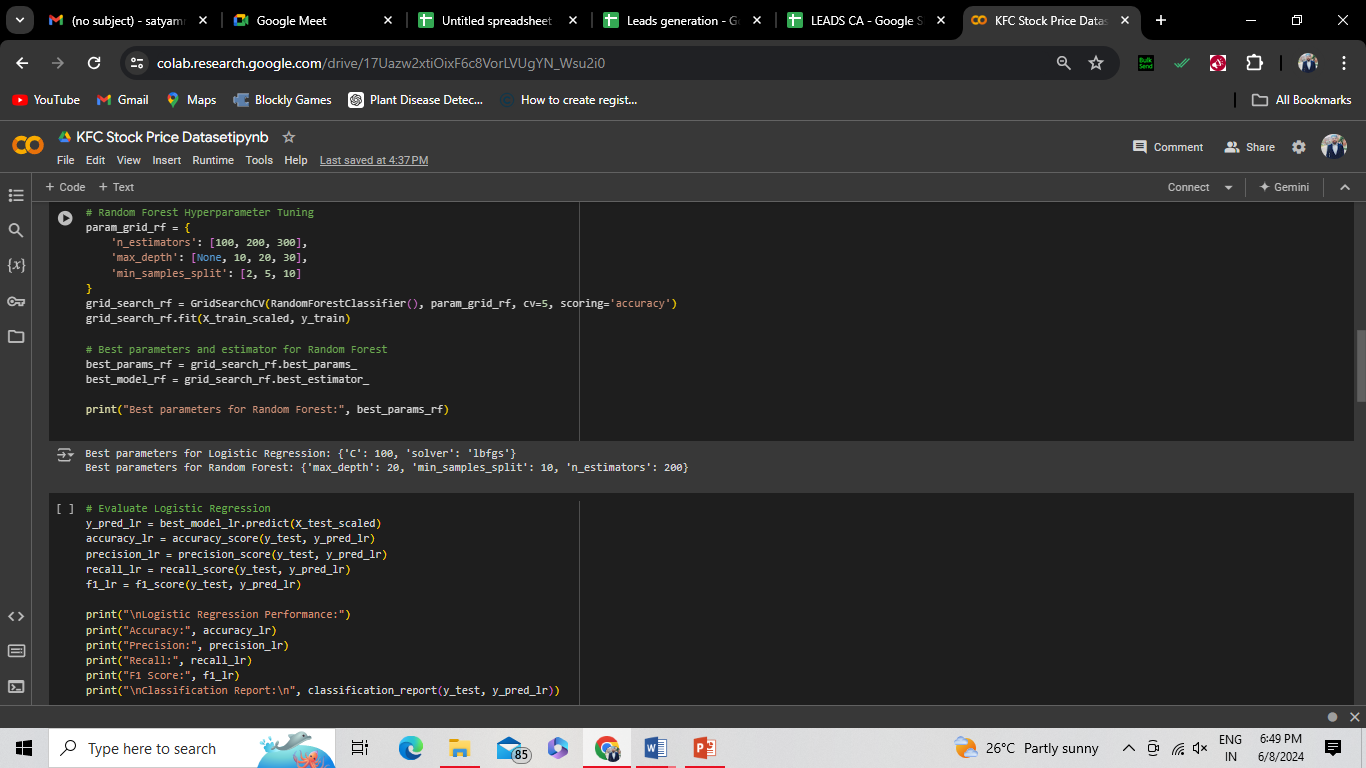
**Step 2: Model Selection and Hyper parameter Tuning**

We'll use GridSearchCV to find the best hyper parameters for logistic regression and random forest.



### Step 3: Model Evaluation

Evaluate the tuned models on the test set and compare their performance.



**Explanation:**

1. **Data loading and Preprocessing**: Load the data, preprocess it by calculating price change and creating a target variable, and split the data into training and testing sets. Standardize the features for model training.
2. **Model Selection and Hyper parameter Tuning**: Use GridSearchCV to find the best hyper parameters for logistic regression and random forest models.
3. **Model Evaluation**:
   * Evaluate the best logistic regression and random forest models using metrics such as accuracy, precision, recall, and F1 score.
   * Print confusion matrices and classification reports.
   * Plot ROC curves and calculate the AUC for both models to visualize their performance.

This code provides a comprehensive pipeline for fine-tuning and evaluating models on a stock price dataset. Adjust the parameters and methods as needed for your specific dataset and requirements.