GSTN ANALYTICS HACKATHON

PREDICTIVE MODEL DEVELOPMENT

VINYAS SHETTY C V

Date – 09 / 10 / 2024

INTRODUCTION

The Hackathon spans 45 days from the start of registration to the final submission date for developed prototypes. Participants will receive a dataset containing 900,000 records with around 21 attributes each. This data is anonymized and labeled, including trained, validated, and non-validated datasets,

Given a dataset D with:

- **Dtrain**: Training data matrix (R(m \times n))
- **Dtest**: Test data matrix (R(m1 \times n))
- **Ytrain**: Target variable matrix (R(m \times 1))
- **Ytest**: Target variable matrix (R(m1 \times 1))

The goal is to build a predictive model (F_{\theta}(X) \rightarrow Y_{\text{pred}}) that accurately predicts (Y_i) for new inputs (X_i).

Steps:

1. **Model Construction**:
    - Define (F_{\theta}(X)) to map input features (X) to predicted outputs (Y_{\text{pred}}).
    - Design (F_{\theta}(X)) to capture the relationship between input features and the target variable.
2. **Training**:
    - Optimize (\theta) by minimizing a loss function (L(Y, F_{\theta}(X))) using (Dtrain)
3. **Testing**:
    - Apply the optimized model (F_{\theta}^*(X)) to (Dtest) to generate predictions (Y_{\text{pred}}).
4. **Performance Optimization**:

- Evaluate performance using accuracy or other metrics (M) on (Y_{\text{pred_test}}).

5. **Submission**:
    - Present (Y_{\text{pred_test}}) with a detailed report including:
        - Modeling approach (commented code, citations, etc.).
        - Key performance indicators as per hackathon metrics.

METHODOLOGY & RESULTS

1. Data Preprocessing and Exploration

- **Data Cleaning:**
    - Checked for missing values.
    - Used Simple Imputer for imputation.
    - Analyzed the distribution of columns with missing values.
    - Applied median imputation for columns with extreme skewness and mean imputation for columns with slight skewness.
    - Checked for outliers to prepare for feature scaling.

2. Feature Scaling

- **Distribution Analysis:**
    - Conducted normality tests on columns.
- **Scaling Techniques:**
    - Used MinMaxScaler for columns not following a normal distribution.
    - Applied RobustScaler for columns with more than 19% outliers.

3. Feature Engineering

- **Techniques Used:**
    - **Forward Feature Selection**
    - **Pearson Correlation Coefficient**
    - **Recursive Feature Elimination**

a. Pearson Correlation Coefficient:

- Measures the linear relationship between two variables.
- Helps identify and remove highly correlated features to reduce multicollinearity, improving model performance and interpretability.

b. Forward Feature Selection:

- Starts with no features and iteratively adds the feature that provides the best performance improvement.
- Performance is measured using the accuracy of a decision tree classifier.
- Returns selected features in the order they were added, along with their corresponding scores.

c. Recursive Feature Elimination:

- Selects features by recursively considering smaller and smaller sets of features.
- Trains the model on the initial set of features and removes the least important features.
- Repeats the process until the desired number of features is reached.

4. Model Training

- **Baseline Models:**
    - Decision Tree
    - Random Forest
- **Ensemble Techniques:**
    - Gradient Boosting
    - AdaBoost

Results

Feature Set: Pearson Correlation Coefficient

- **Decision Tree:**
    - Training Set: Accuracy: 0.9988, F1 Score: 0.9988, Precision: 0.9988, Recall: 0.9988, ROC AUC Score: 0.9966
    - Test Set: Accuracy: 0.9640, F1 Score: 0.9640, Precision: 0.9640, Recall: 0.9640, ROC AUC Score: 0.8943
- **Random Forest:**
    - Training Set: Accuracy: 0.9988, F1 Score: 0.9988, Precision: 0.9988, Recall: 0.9988, ROC AUC Score: 0.9987
    - Test Set: Accuracy: 0.9688, F1 Score: 0.9692, Precision: 0.9696, Recall: 0.9688, ROC AUC Score: 0.9199
- **Gradient Boost:**
    - Training Set: Accuracy: 0.9725, F1 Score: 0.9733, Precision: 0.9750, Recall: 0.9725, ROC AUC Score: 0.9519
    - Test Set: Accuracy: 0.9726, F1 Score: 0.9734, Precision: 0.9752, Recall: 0.9726, ROC AUC Score: 0.9534
- **AdaBoost:**
    - Training Set: Accuracy: 0.9685, F1 Score: 0.9679, Precision: 0.9676, Recall: 0.9685, ROC AUC Score: 0.8903

- Test Set: Accuracy: 0.9691, F1 Score: 0.9686, Precision: 0.9683, Recall: 0.9691, ROC AUC Score: 0.8939

Feature Set: Forward Feature Selection

- **Decision Tree:**
  - Training Set: Accuracy: 0.9930, F1 Score: 0.9930, Precision: 0.9931, Recall: 0.9930, ROC AUC Score: 0.9858
  - Test Set: Accuracy: 0.9696, F1 Score: 0.9698, Precision: 0.9701, Recall: 0.9696, ROC AUC Score: 0.9180
- **Random Forest:**
  - Training Set: Accuracy: 0.9930, F1 Score: 0.9930, Precision: 0.9932, Recall: 0.9930, ROC AUC Score: 0.9897
  - Test Set: Accuracy: 0.9742, F1 Score: 0.9746, Precision: 0.9753, Recall: 0.9742, ROC AUC Score: 0.9410
- **Gradient Boost:**
  - Training Set: Accuracy: 0.9764, F1 Score: 0.9770, Precision: 0.9782, Recall: 0.9764, ROC AUC Score: 0.9580
  - Test Set: Accuracy: 0.9765, F1 Score: 0.9772, Precision: 0.9785, Recall: 0.9765, ROC AUC Score: 0.9601
- **AdaBoost:**
  - Training Set: Accuracy: 0.9741, F1 Score: 0.9748, Precision: 0.9760, Recall: 0.9741, ROC AUC Score: 0.9505
  - Test Set: Accuracy: 0.9743, F1 Score: 0.9749, Precision: 0.9762, Recall: 0.9743, ROC AUC Score: 0.9521

Feature Set: Recursive Feature Elimination

- **Decision Tree:**

- Training Set: Accuracy: 0.9990, F1 Score: 0.9990, Precision: 0.9990, Recall: 0.9990, ROC AUC Score: 0.9970
- Test Set: Accuracy: 0.9661, F1 Score: 0.9660, Precision: 0.9660, Recall: 0.9661, ROC AUC Score: 0.8991

- **Random Forest:**
  - Training Set: Accuracy: 0.9990, F1 Score: 0.9990, Precision: 0.9990, Recall: 0.9990, ROC AUC Score: 0.9989
  - Test Set: Accuracy: 0.9739, F1 Score: 0.9744, Precision: 0.9751, Recall: 0.9739, ROC AUC Score: 0.9411
- **Gradient Boost:**
  - Training Set: Accuracy: 0.9753, F1 Score: 0.9759, Precision: 0.9772, Recall: 0.9753, ROC AUC Score: 0.9556
  - Test Set: Accuracy: 0.9752, F1 Score: 0.9759, Precision: 0.9774, Recall: 0.9752, ROC AUC Score: 0.9577
- **AdaBoost:**
  - Training Set: Accuracy: 0.9722, F1 Score: 0.9728, Precision: 0.9739, Recall: 0.9722, ROC AUC Score: 0.9421
  - Test Set: Accuracy: 0.9724, F1 Score: 0.9730, Precision: 0.9742, Recall: 0.9724, ROC AUC Score: 0.9437

Conclusion:

- **Forward Feature Selection**: Gradient Boost Accuracy: 97.65%, AdaBoost Accuracy: 97.43%
- **Recursive Feature Elimination**: Gradient Boost Accuracy: 97.52% (slight overfitting), AdaBoost Accuracy: 97.24%

So we will be considering Forward Feature Selection for model building .

5. Hyperparameter Tuning

- **Techniques Used:**
    - RandomSearchCV
    - Bayesian Optimization
- **Results:**
    - RandomSearchCV parameters provided better accuracy.
    - **Model Performance for Training Set:**
        - Accuracy: 0.9761
        - F1 Score: 0.9768
        - Precision: 0.8298
        - Recall: 0.9391
        - ROC AUC Score: 0.9595
    - **Model Performance for Test Set:**
        - Accuracy: 0.9761
        - F1 Score: 0.9768
        - Precision: 0.8287
        - Recall: 0.9412
        - ROC AUC Score: 0.9605
- **Conclusion:**

- Since RandomSearchCV parameters yielded better results, we will use these parameters for the ADA-Boost model: {n_estimators=100, learning_rate=1, algorithm='SAMME.R'}.

6. Checking for Overfitting

- **Approach:**
  - Compared training and test set scores.
  - Used k-fold cross-validation and validation dataset.
- **Cross-Validation Results:**
  - Cross-Validation Accuracy Scores: [0.9740, 0.9738, 0.9739, 0.9733, 0.9731, 0.9739, 0.9727, 0.9742, 0.9746, 0.9752]
  - Mean Accuracy: 0.9739
  - Standard Deviation: 0.00068
- **Interpretation:**
  - High Mean Accuracy indicates consistent performance across folds.
  - Low Standard Deviation suggests stable performance with minimal variation.
  - Validation Accuracy: 0.9734
  - The high validation accuracy indicates the model is not overfitting and performs well on unseen data.

7. Model Simplification

- **Objective:**
  - Remove features with zero importance to handle large datasets efficiently.
- **Advantages:**
  - **Reduced Complexity:** Faster training and prediction times.
  - **Simplified Data Preprocessing:** Enhances robustness.

- **Improved Interpretability:** Easier to explain and gain insights.

- **Conclusion:**
  - Removing features with zero importance streamlines the model, making it more efficient, interpretable, and robust.

FINAL CONCLUSION

- **Runtime Comparison:**
  - Gradient Boosting training time: 94.11 seconds
  - AdaBoost training time: 23.10 seconds
  - AdaBoost had a good feature importance score.
- **Model Performance:**
  - **Gradient Boosting & AdaBoosting** models show balanced performance between train and test sets, indicating they are less likely to overfit or underfit.

    - **Random Forest & Decision Tree** show signs of overfitting.
    - **Forward Feature Selection** provides higher accuracy compared to other feature sets.
- **Training Time and Interpretability:**
    - **AdaBoost** is generally faster and more interpretable.
    - AdaBoost scales better with larger datasets due to its fast training time, allowing for quick iterations and model updates.
    - AdaBoost provides better feature interpretability, aiding in decision-making based on feature selection.
- **Final Model Choice:**
    - **Forward Feature Selection** with AdaBoost and randomsearchcv parameters is chosen for the final model due to its balance of accuracy, training time, and interpretability.
    - We have attained **Accuracy of 97.5** percent with the above feature set and model .

PLAGIARISM DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person except where due acknowledgment is made. I have cited all relevant work, libraries, and resources used in this project.

Signature: Vinyas Shetty

Date: 09/09/2024
