Mohammad Mahdi Hemmatyar
Senior Machine learning
Snapp

1. Data Cleaning and Preprocessing

Purpose:
In this section, I clean the data by handling missing values, converting data types, and applying any necessary transformations. Additionally, I encode categorical variables (e.g., LabelEncoder or get_dummies()) to convert them into a format suitable for model training. This preprocessing step is essential for ensuring the data is consistent, clean, and usable for machine learning algorithms.

Why do this?:
Data cleaning ensures that there are no missing or incorrect values that could cause errors during training. Encoding categorical features is crucial because machine learning models typically require numerical input.

2. Handling Imbalanced Data

Purpose:
In this section, I deal with imbalanced datasets using techniques like SMOTE (Synthetic Minority Over-sampling Technique) and Random Undersampling. This ensures that the model does not become biased towards the majority class, which can severely affect classification accuracy and AUC.

Why do this?:
When data is imbalanced, models tend to favor the majority class, leading to poor performance on the minority class. Handling imbalanced data ensures that the model generalizes Ill across all classes.

3. Feature Engineering

Purpose:
Here, I construct new features or transform existing ones (e.g., creating features like Created_hour, or normalizing text using hazm). This step improves the quality of the data that will be fed into the model and can significantly enhance the model's predictive power.

Why do this?:

Feature engineering helps in capturing more useful information from the dataset. Features like Created_hour might add new insights that allow the model to learn better patterns from the data.

4. Text Preprocessing

Purpose:
In this section, I preprocess text data using tools like Hazm or Transformers. This involves steps such as normalization, tokenization, or removing stopwords to prepare text-based features for modeling.

Why do this?:
Raw text data contains noise (e.g., punctuation, stopwords) and needs to be transformed into a format that machine learning or deep learning models can handle. Preprocessing improves model performance by eliminating unnecessary information.

5. Model Training (Random Forest and XGBoost)

Purpose:
Here, I train classification models such as Random Forest and XGBoost using the preprocessed dataset. Both models are trained using the best set of hyperparameters (found via grid search). I use these models to learn patterns in the data and make predictions.

Why use Random Forest and XGBoost?:
Random Forest is an ensemble model that prevents overfitting by averaging multiple decision trees, and XGBoost is a powerful gradient boosting algorithm known for handling structured/tabular data effectively. Both are robust classifiers that perform well on a variety of datasets.

6. Hyperparameter Tuning (GridSearchCV)

Purpose:
In this section, I use GridSearchCV to perform hyperparameter optimization. This allows us to find the best combination of model hyperparameters (e.g., number of trees, depth, etc.) that yields the best performance.

Why do this?:
Tuning hyperparameters is essential for optimizing the model's performance. Without tuning, the model might not learn the best representation of the data, which can result in suboptimal performance.

7. Model Evaluation (AUC and Accuracy)

Purpose:
Here, I evaluate the model using metrics such as AUC (Area Under the Curve), ROC Curve, and Accuracy. This section shows how well the model performs on the test set, providing insights into its predictive power.

Why use AUC and Accuracy?:
AUC is particularly useful for imbalanced classification problems as it focuses on the ranking of predictions rather than just the raw accuracy. Accuracy is a straightforward metric but may not always reflect true model performance in imbalanced datasets, which is why AUC is also crucial.

8. Plotting the ROC Curve

Purpose:
In this section, I plot the ROC Curve to visually assess the model's performance in distinguishing between classes. The curve shows the tradeoff between the True Positive Rate and the False Positive Rate at various thresholds.

Why plot ROC?:
The ROC Curve helps us visualize how well the model separates the classes across different thresholds. It provides a more comprehensive picture of model performance than a single point metric like accuracy.

9. Feature Importance Visualization

Purpose:
Here, I visualize the importance of each feature in the model. This allows us to understand which features contribute the most to the prediction and model decisions.

Why visualize feature importance?:
Feature importance helps identify which features are most useful in making predictions. This can inform future feature engineering efforts and simplify the model by removing irrelevant features.