**Classifier Performance Evaluation Documentation**

horizontal line

# 

# Objective

The goal of this project is to classify resumes based on the skills extracted from them using machine learning classifiers: Support Vector Machine (SVM), Random Forest, and XGBoost. The classifiers are evaluated using both cross-validation and test set performance metrics.

## Data Preparation

* **Data Loading**: The dataset UpdatedResumeDataSet.csv is loaded using Pandas.
* **Text Cleaning**: A cleaning function is applied to preprocess the resume text by removing URLs, special characters, and converting text to lowercase.
* **Skill Extraction**: A predefined list of skills is used to extract relevant skills from the resumes.
* **TF-IDF Vectorization**: The skills are transformed into a TF-IDF representation for model training.
* **Label Encoding**: The categorical labels (categories of resumes) are encoded using LabelEncoder.

**Model Training and Evaluation**

Three classifiers are implemented:

* Support Vector Machine (SVM)
* Random Forest Classifier
* XGBoost Classifier

**Model Training**

The models are trained using an 80/20 train-test split of the data.

**Evaluation Metrics**

The following metrics are used to evaluate the classifiers:

* Accuracy: The proportion of correct predictions.
* Recall: The ratio of correctly predicted positive observations to all actual positives.
* Precision: The ratio of correctly predicted positive observations to the total predicted positives.
* F1 Score: The weighted average of Precision and Recall.

**Cross-Validation and Test Set Performance**

The classifiers were evaluated using cross-validation with 5 folds and also on a separate test set.

make\_scorer is used to create custom scorer functions that specify zero\_division=0 to handle cases where there might be no positive predictions, preventing division by zero errors.

**Results Summary**

#### **Cross-Validation Performance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classifier | Accuracy | Recall | Precision | F1 Score |
| SVM | 0.7542 | 0.6981 | 0.8137 | 0.7147 |
| Random Forest | 0.7633 | 0.7120 | 0.8278 | 0.7234 |
| XGBoost | 0.7542 | 0.6990 | 0.8343 | 0.7197 |

#### 

#### **Test Set Performance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classifier | Accuracy | Recall | Precision | F1 Score |
| SVM | 0.7358 | 0.7299 | 0.8783 | 0.7555 |
| Random Forest | 0.7358 | 0.7299 | 0.8769 | 0.7550 |
| XGBoost | 0.7306 | 0.7232 | 0.8636 | 0.7469 |

#### 

**Analysis of Results**

**Cross-Validation:**

* The Random Forest classifier achieved the highest accuracy (0.7633) and F1 Score (0.7234), indicating it performed best among the three models in a cross-validated setting.
* SVM and XGBoost had identical accuracy, with slightly lower scores compared to Random Forest.

**Test Set:**

* On the test set, all classifiers showed a drop in performance compared to cross-validation, which is typical when moving from training/validation to unseen data.
* The SVM classifier had the highest precision (0.8783) on the test set, indicating it had a lower false positive rate.

## Conclusion

The Random Forest classifier demonstrated the best performance in terms of accuracy and F1 score during cross-validation. However, the SVM model exhibited a higher precision on the test set, suggesting that while it may be less accurate overall, it was better at minimizing false positives. Further hyperparameter tuning and feature engineering could improve the performance of these models.