**Predict High-Potential HR Service Leads**

**Introduction:**

This challenge involves predicting which companies are likely to become high-potential leads for HR services based on their funding and hiring trends. The dataset includes synthetic data designed to mimic real-world information, allowing participants to develop and test predictive models. The objective is to classify companies as either a "hot lead" or not using machine learning techniques.

**Problem Statement:**

The goal is to predict whether a company is a "hot lead" for HR services based on various financial, hiring, and growth-related features. The target variable, “is\_hot\_lead”, indicates whether a company engaged with HR services within a specific period. Participants will submit predicted probabilities for each company in the holdout dataset.

**Dataset:**

The dataset includes the following features:

| **Feature** | **Description** | **Data Type** |
| --- | --- | --- |
| company\_id | Unique identifier for the company. | String |
| company\_name | Name of the company. | String |
| industry | Industry the company operates in. | String |
| funding\_rounds | Number of funding rounds the company has undergone. | Integer |
| total\_funding | Total amount of funding the company has received. | Float |
| last\_funding\_date | Date of the company's most recent funding round. | Date |
| job\_postings\_30d | Number of job postings by the company in the last 30 days. | Integer |
| employee\_growth\_pct | Percentage change in employee count over the last six months. | Float |
| hiring\_roles | Types of roles the company is hiring for. | String |
| industry\_growth\_rate | Growth rate of the company's industry. | Float |
| regional\_employment\_trend | Employment trend in the company's region. | Float |
| funding\_per\_employee | Ratio of total funding to employee count. | Float |
| days\_since\_last\_funding | Number of days since the company's last funding round. | Integer |
| growth\_momentum | Composite feature combining funding and hiring activity. | Float |
| is\_hot\_lead | Binary variable indicating whether the company is a high-potential lead for HR services. | Int |

**Data Preprocessing:**

* **Dealing with Infinite Values**: inf values in numerical columns were replaced with NaN and dropped the NaN values.
* **Encoding Categorical Variables**: Label encoding was applied to categorical features to convert them into numerical format.
* **Feature Scaling**: Numerical features were standardized using StandardScaler to ensure consistency across different scales.
* **Class Imbalance Handling**: The dataset exhibited an imbalance in the target variable. To mitigate this, **SMOTE (Synthetic Minority Over-sampling Technique)** was applied to generate synthetic samples for the minority
* Since there are three datasets available, performed all the preprocessing steps mentioned before in all the datasets.

**Feature Engineering:**

Feature engineering was performed to extract and create new features that enhance model performance:

* **Extracting Features from Datetime Columns**:
  + last\_funding\_year, last\_funding\_month, last\_funding\_day
  + last\_funding\_day\_of\_week, last\_funding\_day\_of\_year, last\_funding\_week\_of\_year
  + last\_funding\_is\_weekend (binary indicator for weekends)
  + The original last\_funding\_date column was then removed.

**Model Training:**

Multiple machine learning models were trained to evaluate their performance:

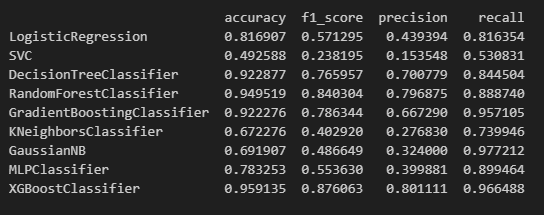
* **Logistic Regression**: Used with class\_weight="balanced" to handle imbalanced data.
* **Support Vector Machine (SVC)**: Trained with an RBF kernel and probability estimation, with class\_weight="balanced".
* **Decision Tree**: Implemented with class\_weight="balanced" to reduce bias towards the majority class.
* **Random Forest**: A robust ensemble method trained with n\_estimators=100 and class\_weight="balanced".
* **Gradient Boosting**: Applied with n\_estimators=100 to enhance generalization.
* **K-Nearest Neighbors (KNN)**: Trained with n\_neighbors=5 for classification.
* **Naïve Bayes**: Implemented as a probabilistic model without class weighting.
* **Multi-Layer Perceptron (MLP)**: Used with hidden\_layer\_sizes=(100,) and max\_iter=500.
* **XGBoost**: Applied with n\_estimators=100, learning\_rate=0.1, and eval\_metric="logloss", optimizing scale\_pos\_weight to handle class imbalance.

**Model Testing & Evaluation:**

The models were tested using a separate test dataset. The evaluation included:

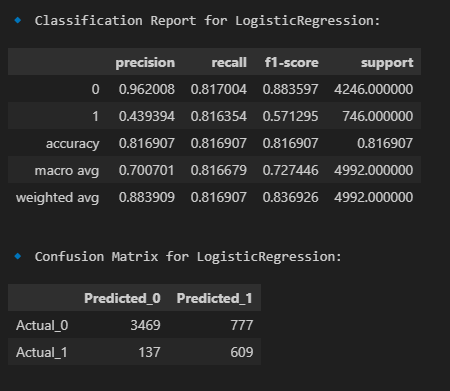
* **Confusion Matrix Analysis**: Understanding false positives and false negatives.
* **Primary Metric (F1-Score)**: Balancing precision and recall for imbalanced data.
* **Secondary Metrics**:
  + **Accuracy**: Overall correctness of predictions.
  + **Precision**: Proportion of true positives among predicted positives.
  + **Recall**: Proportion of actual positives correctly identified.

**Evaluation Metrics Before Hyper-Parameter Tuning:**

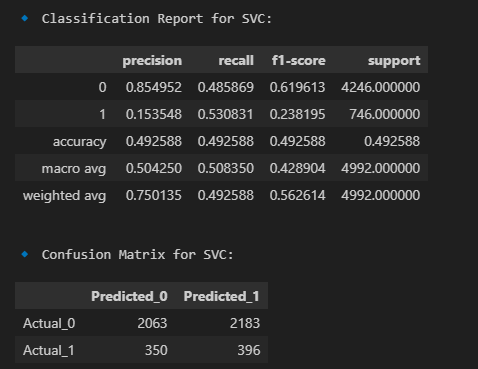


**Classification report and Confusion Matrix for model:**

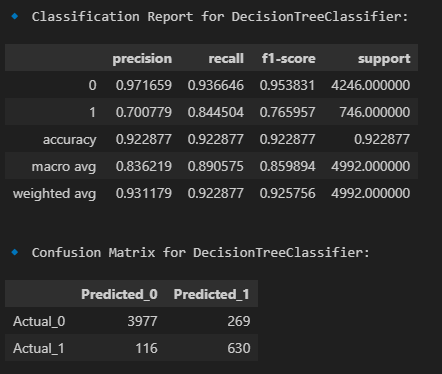
**Logistic Regression**

****

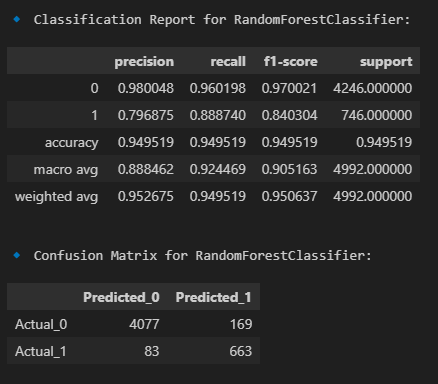
**SVC**

****

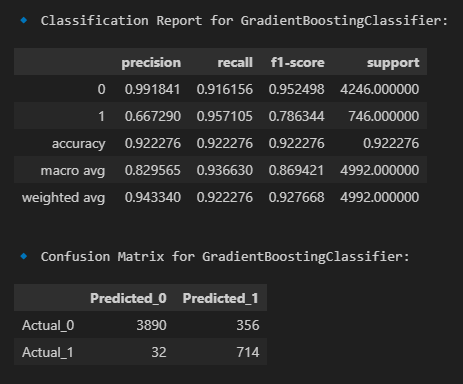
**Decision Tree Classifier**

****

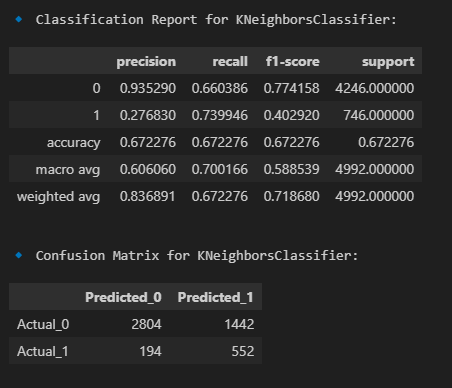
**Random Forest Classifier**

****

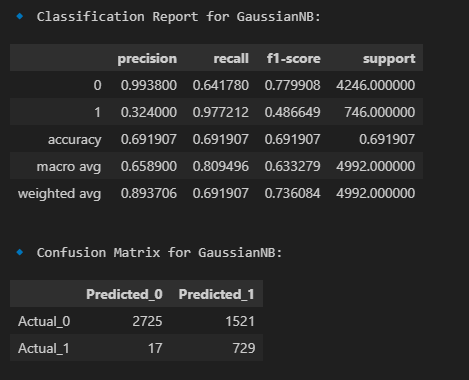
**Gradient Boosting Classifier**

****

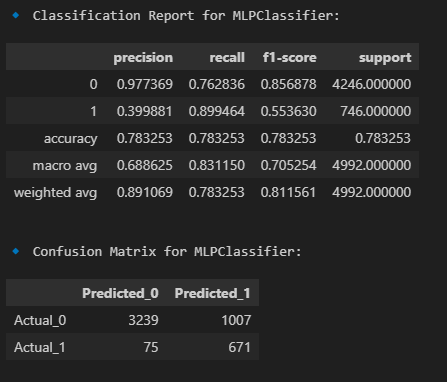
**K-Neighbors Classifier**

****

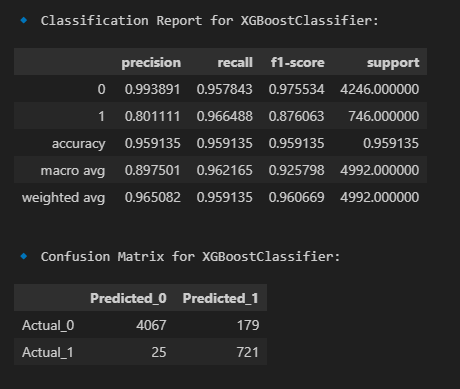
**GaussianNB**

****

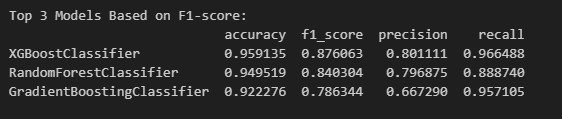
**MLP Classifier**

****

**Xgb Classifier**

****

**Best Models:**

****

Out of these three XGBClassifier have higher accuracy, f1 score, precision, recall values.so need to perform hyperparameter tuning for that model

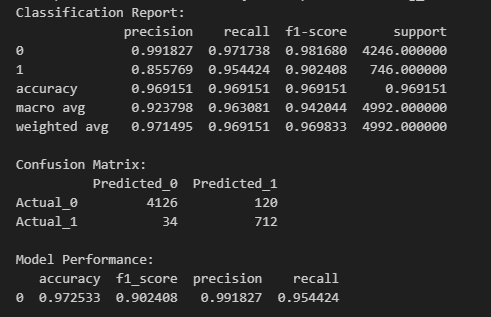
**Hyperparameter Tuning:**

Hyperparameter tuning was performed using **RandomizedSearchCV** for XGBoost:

* **XGBoost**: Optimized n\_estimators, learning\_rate, max\_depth, subsample, min\_child\_weight, colsample\_bytree, gamma, reg\_alpha, and reg\_lambda using a randomized search strategy.

**Best parameters for XGBoost:**

{'subsample': 0.9, 'reg\_lambda': 0, 'reg\_alpha': 1.5, 'n\_estimators': 400, 'min\_child\_weight': 3, 'max\_depth': 10, 'learning\_rate': 0.01, 'gamma': 0.2, 'colsample\_bytree': 0.8}

****

**Submission:**

The submission.csv is stored in submission folder.