**Dataset Report and Documentation**

**Dataset Overview**

The dataset comprises 50,000 entries and 30 columns. It contains employee information with the objective of identifying factors associated with burnout symptoms. The target variable for this dataset is Burnout\_Symptoms, which is a categorical variable indicating whether an employee experiences burnout ('yes''no', occasionally ).

**Columns Overview:**

| **Column Name** | **Description** | **Data Type** |
| --- | --- | --- |
| **ID** | Unique identifier for each employee | Integer (int) |
| **Age** | Age of the employee | Integer (int) |
| **Gender** | Gender of the employee (categorical: 'male'/'female') | Categorical (object) |
| **Marital\_Status** | Marital status of the employee | Categorical (object) |
| **Job\_Role** | Role of the employee within the company | Categorical (object) |
| **Experience\_Years** | Number of years of professional experience | Integer (int) |
| **Monthly\_Salary\_INR** | Monthly salary (in INR) | Float (float) |
| **Working\_Hours\_per\_Week** | Working hours per week | Integer (int) |
| **Commute\_Time\_Hours** | Commute time in hours | Float (float) |
| **Remote\_Work** | Whether the employee works remotely (True/False) | Boolean (bool) |
| **Stress\_Level** | Stress level score (higher value = higher stress) | Float (float) |
| **Health\_Issues** | Health-related issues (categorical: 'yes'/'no') | Categorical (object) |
| **Company\_Size** | Company size (categorical: 'small', 'medium', 'large') | Categorical (object) |
| **Department** | Department in which the employee works | Categorical (object) |
| **Sleep\_Hours** | Average hours of sleep per night | Float (float) |
| **Physical\_Activity\_Hours\_per\_Week** | Hours of physical activity per week | Float (float) |
| **Mental\_Health\_Leave\_Taken** | Whether the employee has taken mental health leave (True/False) | Categorical (object) |
| **Manager\_Support\_Level** | Manager support level (higher value = higher support) | Integer (int) |
| **Work\_Pressure\_Level** | Work pressure level (higher value = higher pressure) | Float (float) |
| **Annual\_Leaves\_Taken** | Annual leave days taken | Float (float) |
| **Work\_Life\_Balance** | Work-life balance score (higher value = better balance) | Float (float) |
| **Family\_Support\_Level** | Family support level (higher value = more support) | Integer (int) |
| **Job\_Satisfaction** | Job satisfaction score (higher value = more satisfaction) | Integer (int) |
| **Performance\_Rating** | Performance rating score (higher value = better performance) | Float (float) |
| **Team\_Size** | Team size (number of employees in the team) | Integer (int) |
| **Training\_Opportunities** | Whether the employee has training opportunities (True/False) | Boolean (bool) |
| **Gender\_Bias\_Experienced** | Whether gender bias was experienced (True/False) | Boolean (bool) |
| **Discrimination\_Experienced** | Whether discrimination was experienced (True/False) | Boolean (bool) |
| **Burnout\_Symptoms** | Target variable indicating if the employee experiences burnout ('yes'/'no') | Categorical (object) |
| **Location** | Employee's location (categorical) | Categorical (object) |

**Data Preprocessing**

**Missing Values Handling**

Missing values in the dataset were handled with mode and mean imputation

| **Column** | **Missing Values Count** | **Imputation Strategy** |
| --- | --- | --- |
| Job\_Role | 1 | Replaced with Mode |
| Commute\_Time\_Hours | 1 | Replaced with Mean |
| Stress\_Level | 1 | Replaced with Mean |
| Health\_Issues | 12,541 | Replaced with |
| Mental\_Health\_Leave\_Taken | 1 | Replaced with Mode |
| Work\_Pressure\_Level | 1 | Replaced with Mean |
| Annual\_Leaves\_Taken | 1 | Replaced with Mean |
| Work\_Life\_Balance | 2 | Replaced with Mean |
| Performance\_Rating | 1 | Replaced with Mean |
|  |  |  |

**Dropping Column**

We dropped certain columns that we thought unnecessary for the analysis and they are not directly connected to target column

Dropped Columns: Remote\_Work, CompanySize, Location and ID

**High and Low Cardinality Feature Handling**

The features were separated based on their cardinality:

The **high cardinality** features include 'Job\_Role' (6 unique values) and 'Department' (6 unique values) and 'Mental\_Health\_Leave\_Taken' (6 unique values)

The **low cardinality** features include 'Gender' (3 unique values), 'Marital\_Status' (4 unique values), 'Health\_Issues' (3 unique values), and 'Burnout\_Symptoms' (4 unique values). Python

**Encoding**

For handling categorical variables, **Label Encoding** was used for all categorical features to convert them into numerical values.

**Imbalanced Classes Check**

We checked for class imbalance in the target variable Burnout\_Symptoms by comparing the distribution of the classes by Smote

**Duplicate Values Check and Used Klib** for data cleaning and reduced size of dataset

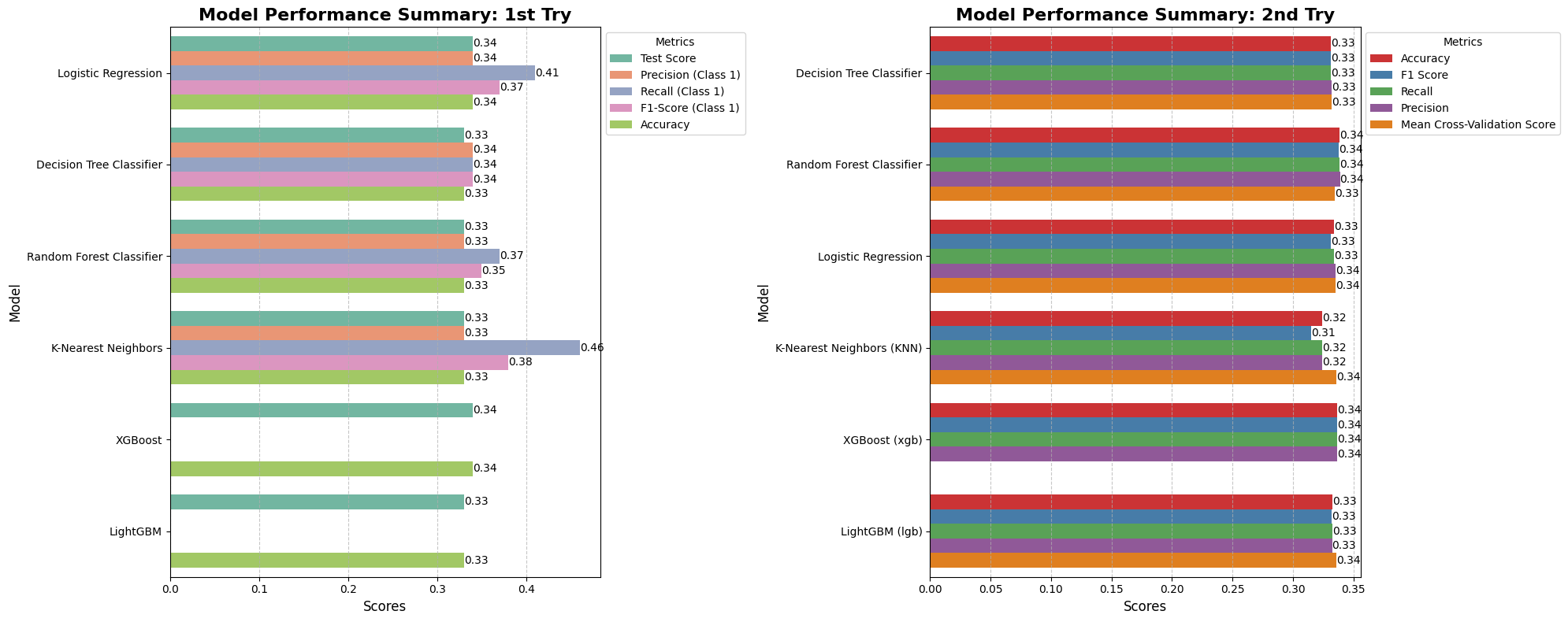
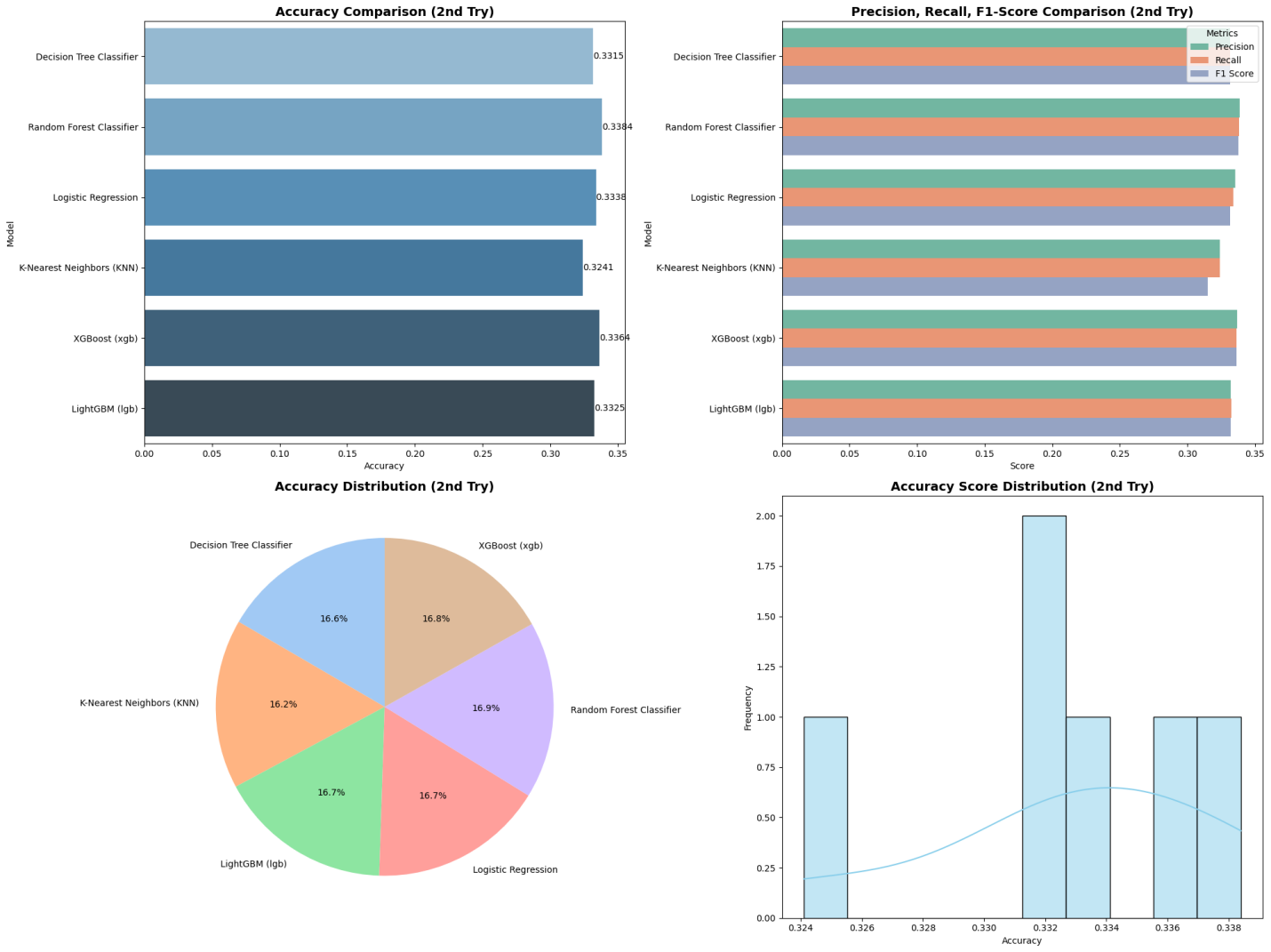
**Results:**

**Model Performance Summary: 1st Try**

| **Model** | **Best Cross-Validation Score** | **Test Score** | **Precision (Class 1)** | **Recall (Class 1)** | **F1-Score (Class 1)** | **Accuracy** |
| --- | --- | --- | --- | --- | --- | --- |
| **Logistic Regression** | 0.33 | 0.34 | 0.34 | 0.41 | 0.37 | 0.34 |
| **Decision Tree Classifier** | 0.30 | 0.33 | 0.34 | 0.34 | 0.34 | 0.33 |
| **Random Forest Classifier** | 0.33 | 0.33 | 0.33 | 0.37 | 0.35 | 0.33 |
| **K-Nearest Neighbors** | 0.34 | 0.33 | 0.33 | 0.46 | 0.38 | 0.33 |
| **XGBoost** | - | 0.34 | - | - | - | 0.34 |
| **LightGBM** | - | 0.33 | - | - | - | 0.33 |

**Model Performance Summary: 2nd Try**

| **Model** | **Accuracy** | **F1 Score** | **Recall** | **Precision** | **Cross-Validation Scores** | **Mean Cross-Validation Score** |
| --- | --- | --- | --- | --- | --- | --- |
| **Decision Tree Classifier** | 0.3315 | 0.3315 | 0.3315 | 0.3317 | [0.3306, 0.3362, 0.3282, 0.331, 0.3349] | 0.3322 |
| **Random Forest Classifier** | 0.3384 | 0.3378 | 0.3384 | 0.3388 | [0.3255, 0.3348, 0.3327, 0.3384, 0.3423] | 0.3347 |
| **Logistic Regression** | 0.3338 | 0.3314 | 0.3338 | 0.3353 | [0.3319, 0.3322, 0.3371, 0.3424, 0.334] | 0.3355 |
| **K-Nearest Neighbors (KNN)** | 0.3241 | 0.3149 | 0.3241 | 0.3239 | [0.3347, 0.3385, 0.3429, 0.3302, 0.334] | 0.3361 |
| **XGBoost (xgb)** | 0.3364 | 0.3364 | 0.3364 | 0.3366 | [0.3292, 0.3379, 0.3293, 0.3409] | - |
| **LightGBM (lgb)** | 0.3325 | 0.3321 | 0.3325 | 0.3323 | [0.3338, 0.3318, 0.3385, 0.3372, 0.3371] | 0.3357 |



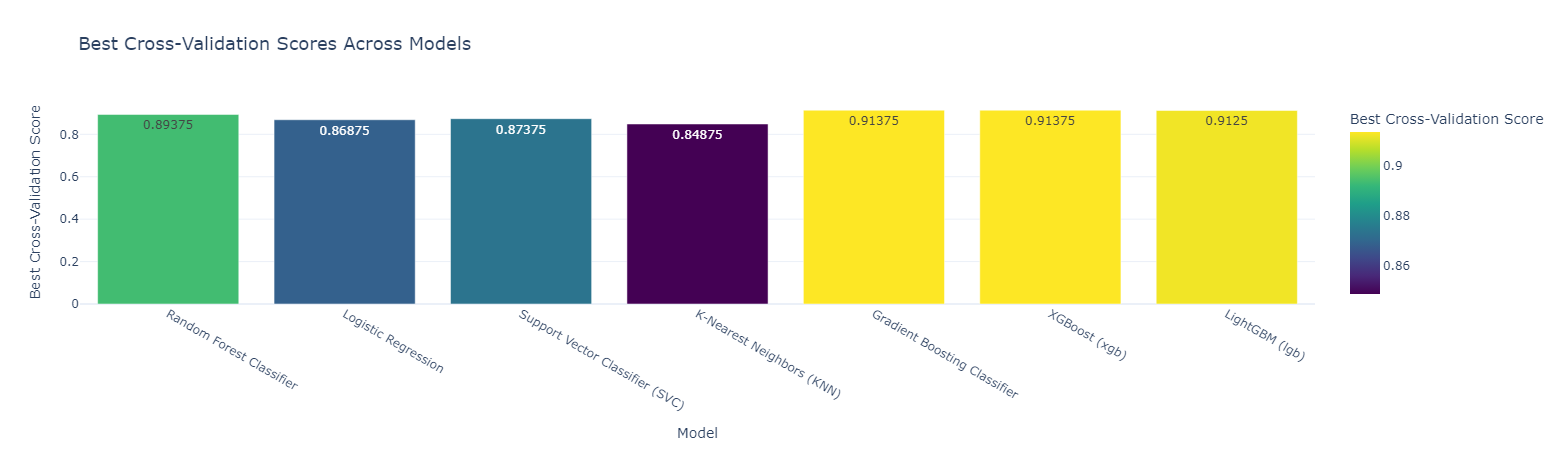
**After Hyperparameter Tuning**

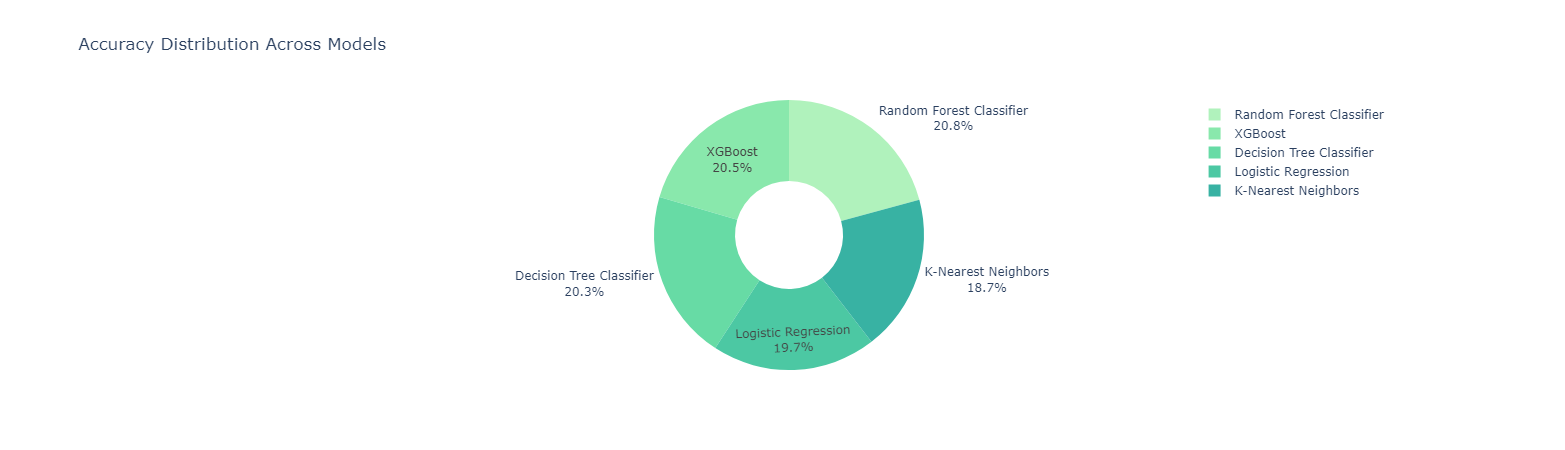
**Model Performance Summary: 1st Try**

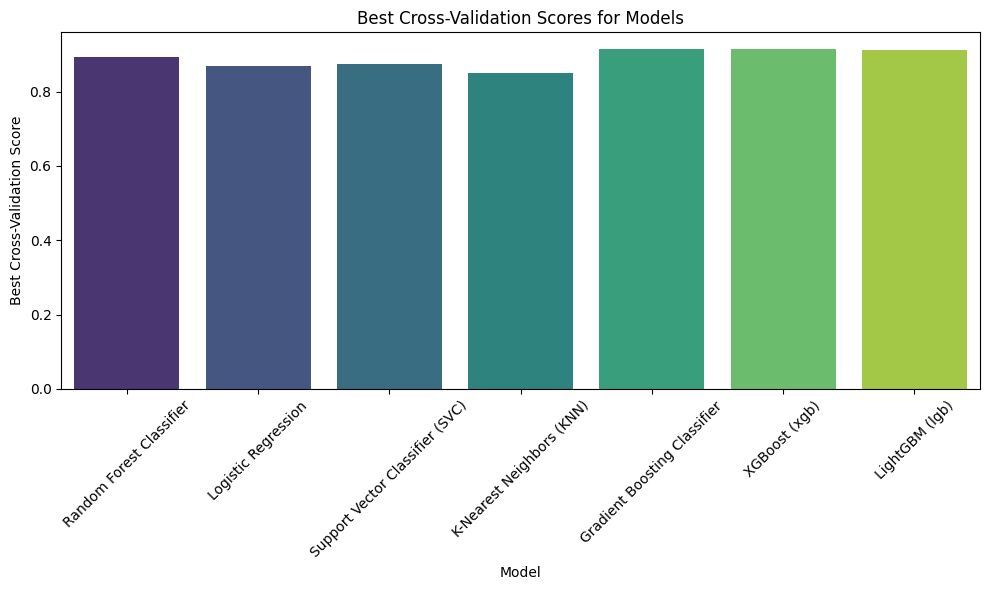
| **Model** | **Best Cross-Validation Score** | **Test Score** | **Precision** | **Recall** | **F1-Score** | **Accuracy** |
| --- | --- | --- | --- | --- | --- | --- |
| **Random Forest Classifier** | 0.89375 | 0.90 | 0.95 | 0.86 | 0.90 | 0.90 |
| **Logistic Regression** | 0.87 | 0.85 | 0.91 | 0.80 | 0.86 | 0.85 |
| **Decision Tree Classifier** | 0.86 | 0.88 | 0.93 | 0.83 | 0.88 | 0.88 |
| **K-Nearest Neighbors** | 0.81 | 0.81 | 0.88 | 0.75 | 0.81 | 0.81 |
| **XGBoost** | - | 0.885 | - | - | - | 0.885 |
|  |  |  |  |  |  |  |

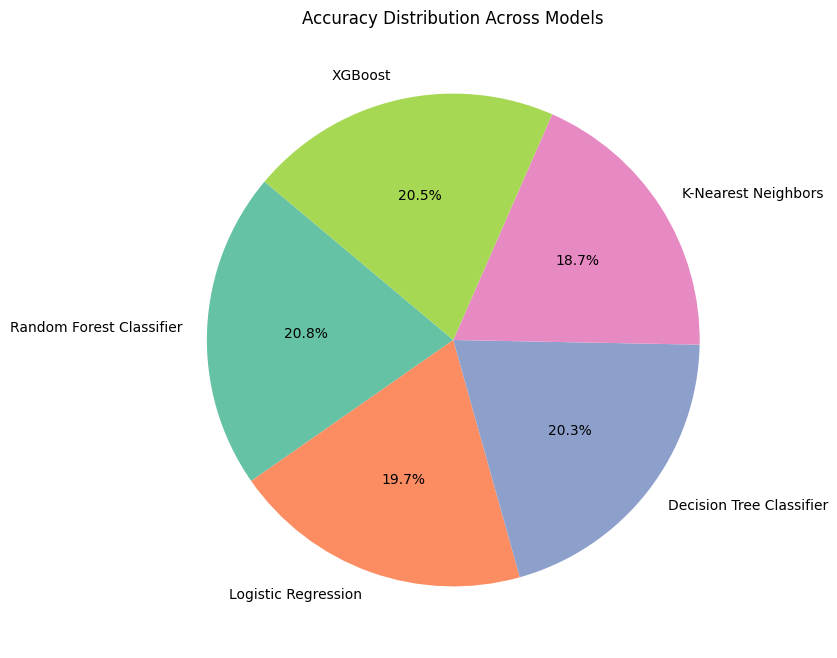
**Model Performance Summary: 2nd Try**

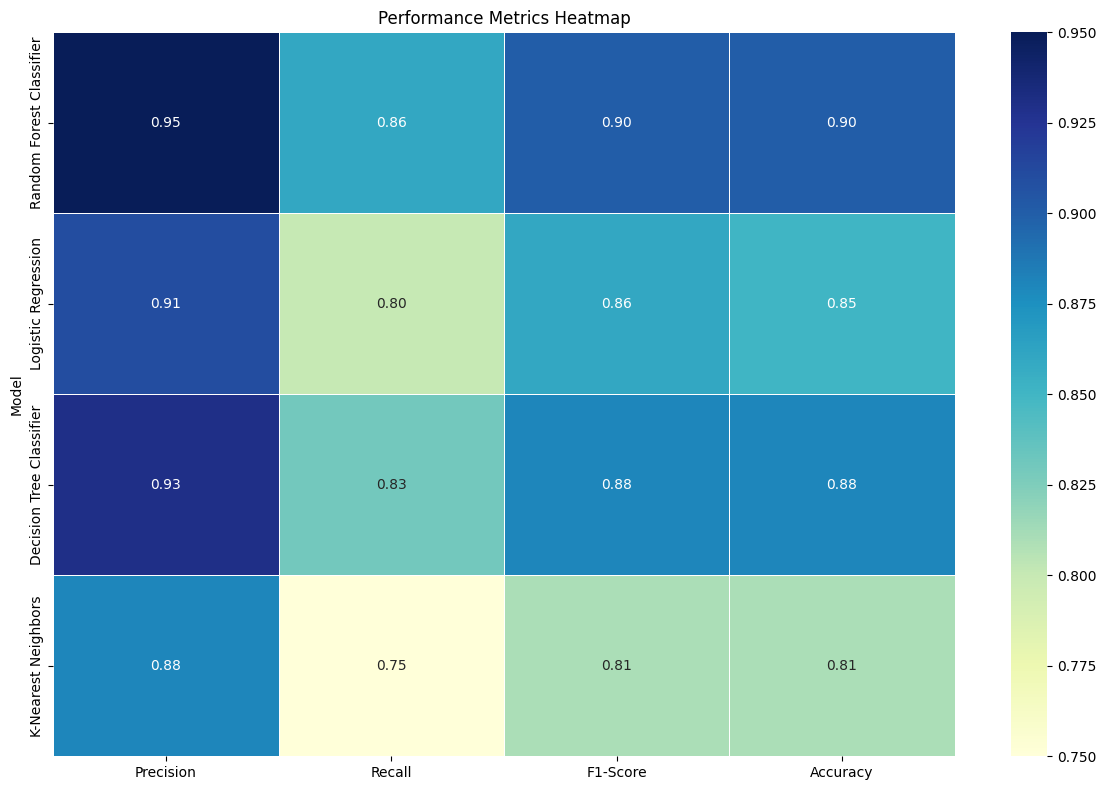
| **Model** | **Best Cross-Validation Score** | **Test Score** |
| --- | --- | --- |
| **Random Forest Classifier** | 0.89375 | 0.90 |
| **Logistic Regression** | 0.86875 | 0.865 |
| **Support Vector Classifier (SVC)** | 0.87375 | 0.88 |
| **K-Nearest Neighbors (KNN)** | 0.84875 | 0.83 |
| **Gradient Boosting Classifier** | 0.91375 | 0.88 |
| **XGBoost (xgb)** | 0.91375 | 0.885 |
| **LightGBM (lgb)** | 0.9125 | 0.895 |

****

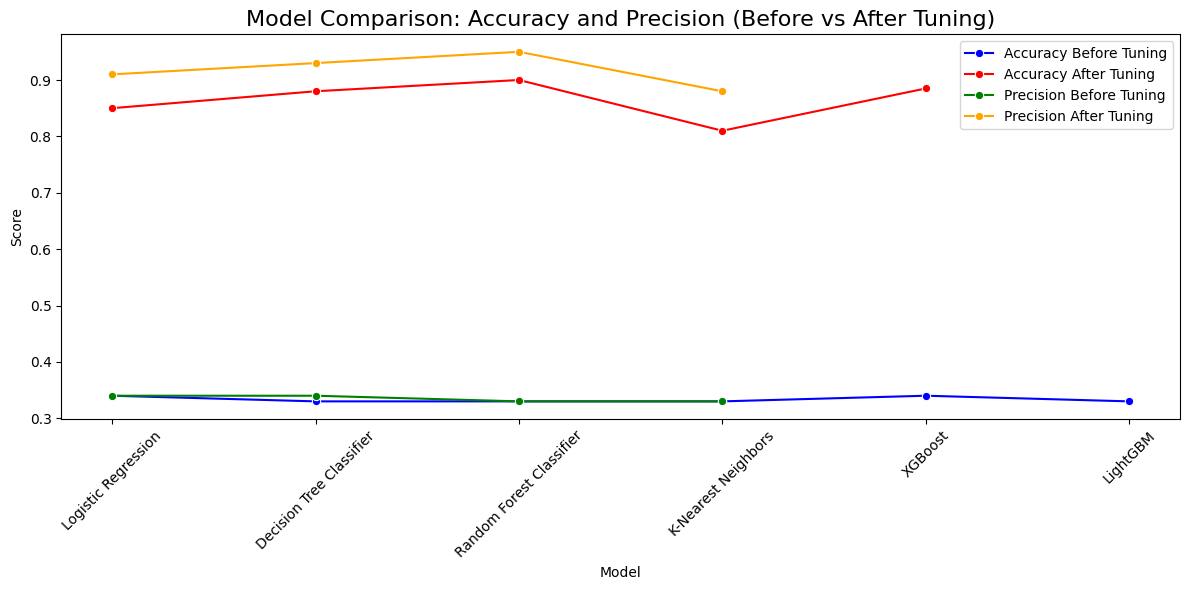
****

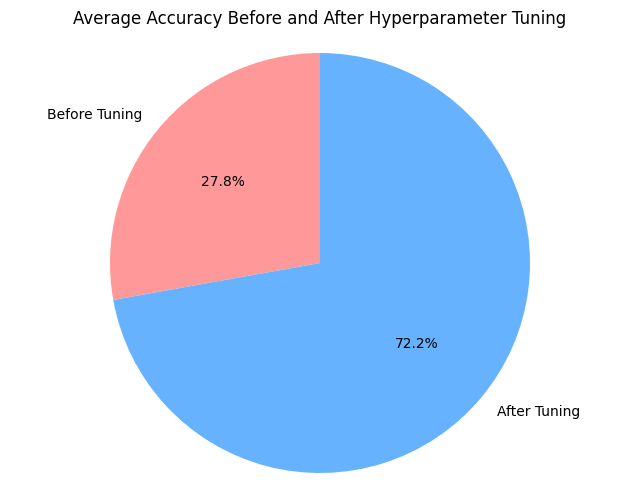


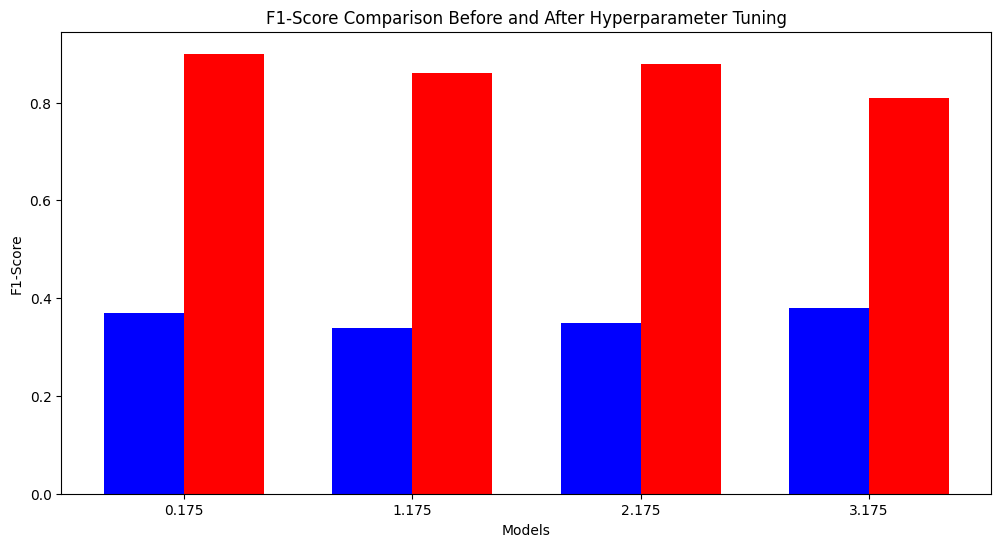


****

**Compare Before and After Hyperparameter Tuning**

****

****

****

**Summary:**

Hyperparameter tuning showed significant performance improvements, especially for Random Forest, Logistic Regression, and Decision Tree, improving both accuracy and F1-Score

Before Hyperparameter Tuning:

* Random Forest Classifier had the best accuracy at 33% but showed moderate performance overall.
* K-Nearest Neighbors had the highest recall (46%) but lacked balance in precision.
* Models like Logistic Regression and Decision Tree Classifier had similar performance with moderate accuracy (~33%).

After Hyperparameter Tuning:

* Random Forest Classifier achieved the best accuracy (90%), precision (0.95), and F1-Score (0.90).
* Logistic Regression improved to 85% accuracy and 0.86 F1-Score.
* K-Nearest Neighbors and Decision Tree Classifier also showed improvements in accuracy and F1-Score.

.