* **Abstract**

The WiDS Datathon 2020 project aimed to develop a predictive model to assist hospitals in improving ICU resource allocation by forecasting patient survival rates. The project involved data preprocessing, exploratory data analysis, and model development using machine learning techniques such as logistic regression, KNN, random forests, Ada boosting and gradient boosting. Evaluation metrics focused on accuracy, precision, recall, F1score, and AUCROC. Key results highlighted the random forest and Gradient Boosting model's superior performance with the highest accuracy.

* **Objective:**

1. To preprocess and analyze a comprehensive ICU dataset, identifying key features that influence patient survival.
2. To develop and evaluate multiple machine learning models to predict ICU patient survival rates with high accuracy.
3. To interpret the models to understand the most significant factors contributing to patient survival, providing actionable insights for healthcare professionals.

* **Data Preprocessing:**

**1. Preliminary Analysis:**

1.1 Overview of the data, including source, size, and type:

Data is large and contains many features and sample points

תמונה שמכילה טקסט, גופן, לבן, טיפוגרפיה

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, גופן, מספר, קו

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, מלבן, תרשים

התיאור נוצר באופן אוטומטי1.2Features data type distribution plot: It appears that most of the data are of numeric type (int, float, category, binary, etc.).

תמונה שמכילה טקסט, צילום מסך, תרשים, עיגול

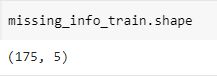
התיאור נוצר באופן אוטומטי1.3 Target Distribution Plot:

The number of samples with an output of 0 is significantly greater than those with an output of 1. This indicates that data is highly imbalanced due to the substantial disparity in the number of samples for each class. We will address this imbalance after completing the data cleaning phase.

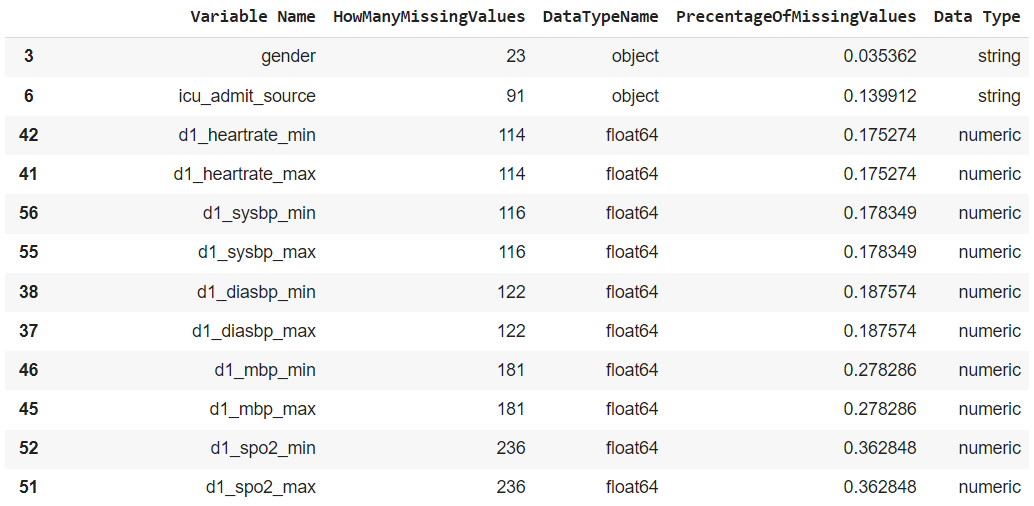
**2. Data Preparation:**

2.1.1Handling Missing Values (Numeric):

*Step1:* Removing 80% missing values columns

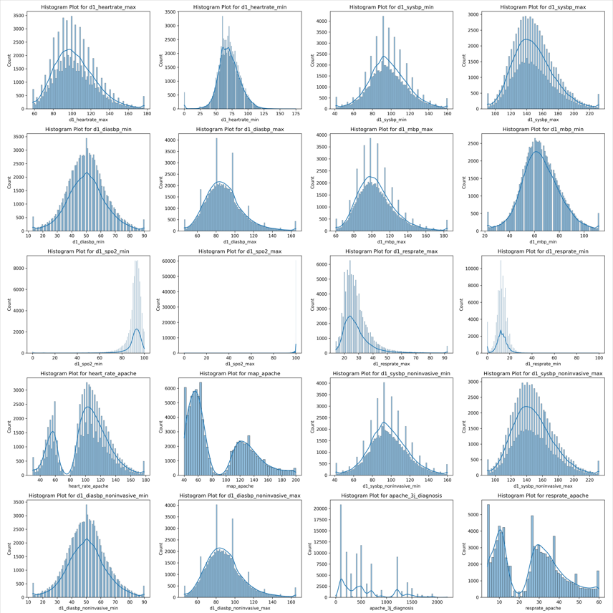
תמונה שמכילה טקסט, גופן, לבן, צילום מסך

התיאור נוצר באופן אוטומטי*Before:*  *After:*

Summaries Missing Values

*Step 2:* Fill in missing values with mean and median values according to plots. Calculating the skewness of each feature distribution.

* 0.5<For skewness <0.5: missing values were replaced with mean
* Otherwise: missing values were replaced with median



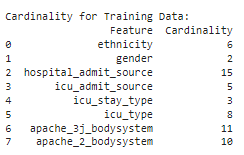
2.1.2 Handling Missing Values (Categorical):

After encoding categorical features, missing values were replaced with mode values.

Encoding Categorical features:

Group Features by Cardinality:

* Low Cardinality: Features with a small number of unique values (<=5) will be encoded using Label **Encoding**.
* High Cardinality: Features with many unique values (>=5) will be encoded using One **Hot Encoding**.

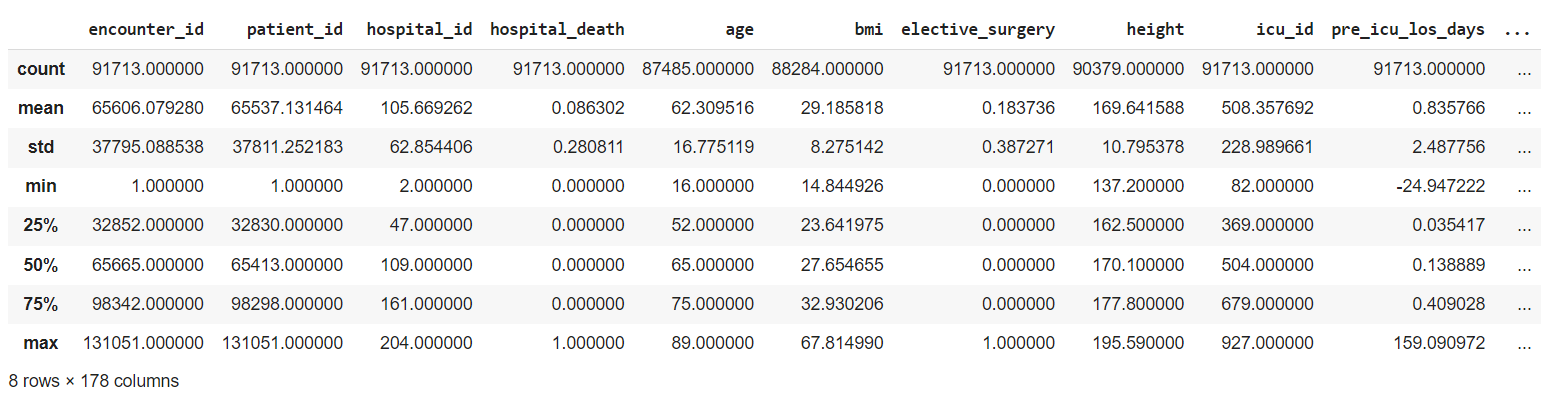


ראש הטופס

תחתית הטופס

* + 1. Normalizing:

Data also was normalized due to Different features in a dataset having different scales or units. Also using ML models that are sensitive to that case, data should be normalized for better models' performance.

* **Exploratory Data Analysis (EDA):**
* **Summary statistics**

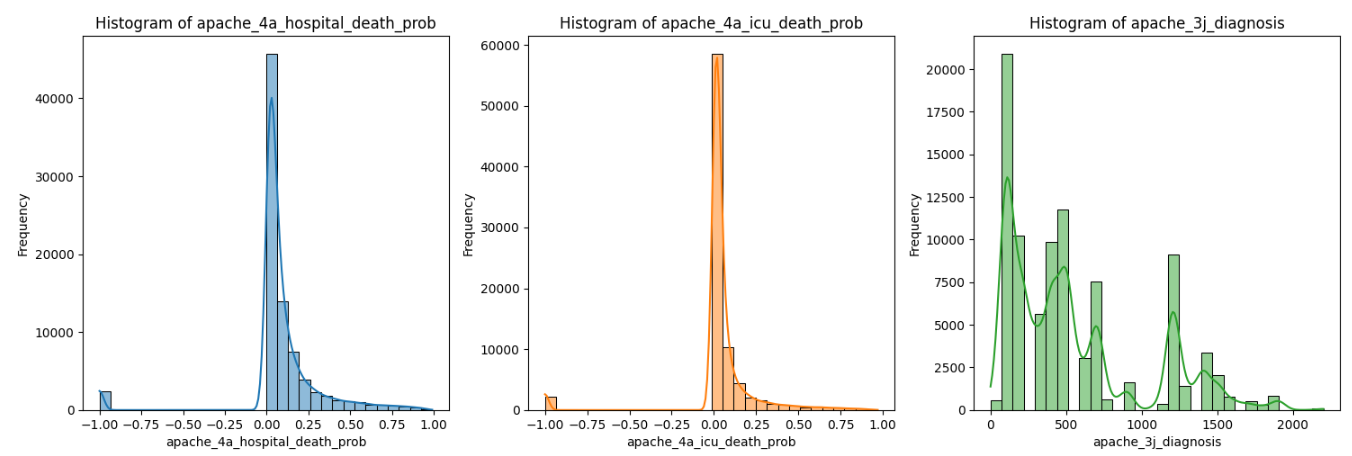
Data values are varied among the features.

* **10 Top important features**

Selecting the top 10 features with best MI score based on mutual information for understating behavior.

|  |  |
| --- | --- |
| Features | MI Score |
| apache\_4a\_icu\_death\_prob | 0.075611 |
| apache\_4a\_hospital\_death\_prob | 0.075345 |
| apache\_3j\_diagnosis | 0.041288 |
| d1\_lactate\_min | 0.037755 |
| d1\_arterial\_ph\_min | 0.036371 |
| d1\_lactate\_max | 0.036037 |
| apache\_2\_diagnosis | 0.035680 |
| gcs\_motor\_apache | 0.035379 |
| d1\_arterial\_ph\_max | 0.031478 |
| gcs\_eyes\_apache | 0.030363 |
| d1\_sysbp\_min | 0.030205 |

* **Plotting**

****

**תמונה שמכילה תרשים, קו, עלילה, טקסט

התיאור נוצר באופן אוטומטי**

**תמונה שמכילה תרשים, עלילה, קו, טקסט

התיאור נוצר באופן אוטומטי**

**תמונה שמכילה תרשים, טקסט, עלילה, צילום מסך

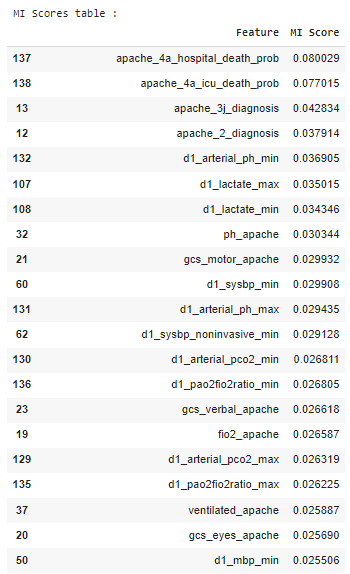
התיאור נוצר באופן אוטומטי**

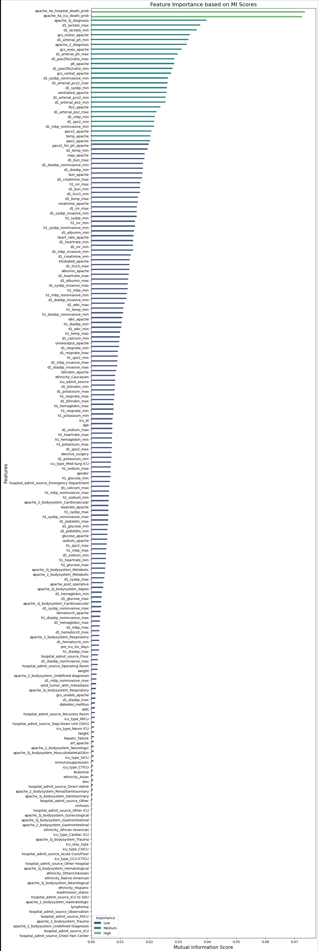
* **Feature Engineering**
  1. **Mutual information:**

Data involves numeric and categorical information; therefore, mutual information was found as appropriate method for discovering feature importance by calculating their MI score.

Plotting MI Scores to get visualization understanding of the features importance.

By observation, MI scores of 0.01 should be kept and all features with MI scores under 0.01 will be dropped off.





תמונה שמכילה טקסט, צילום מסך, עיצוב

התיאור נוצר באופן אוטומטי

Reduction in features = 126 features dropped

תמונה שמכילה טקסט, גופן, לבן, עיצוב

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, גופן, לבן, עיצוב

התיאור נוצר באופן אוטומטי

* 1. **PCA (Principal Component Analysis):**

Using PCA for the rest of the features. Getting finally *around* 30 combined features by PCA.

* Brief PCA matrix:

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

* Explained variance plot exhibit the rightness of approaching *around* 30 features:

תמונה שמכילה טקסט, עלילה, קו, תרשים

התיאור נוצר באופן אוטומטי

* Final Features Reduction:

תמונה שמכילה טקסט, צילום מסך, קו, עלילה

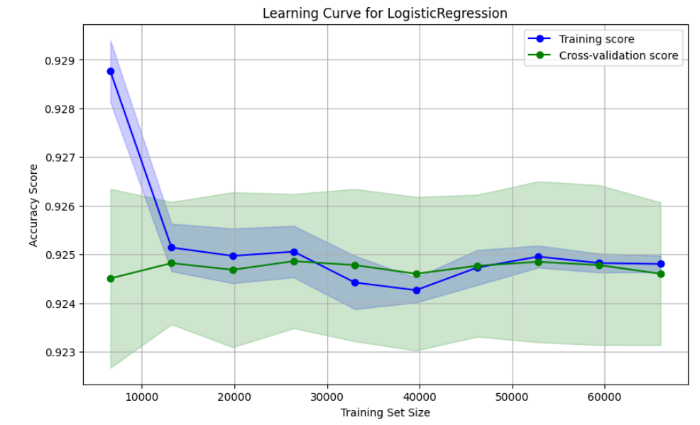
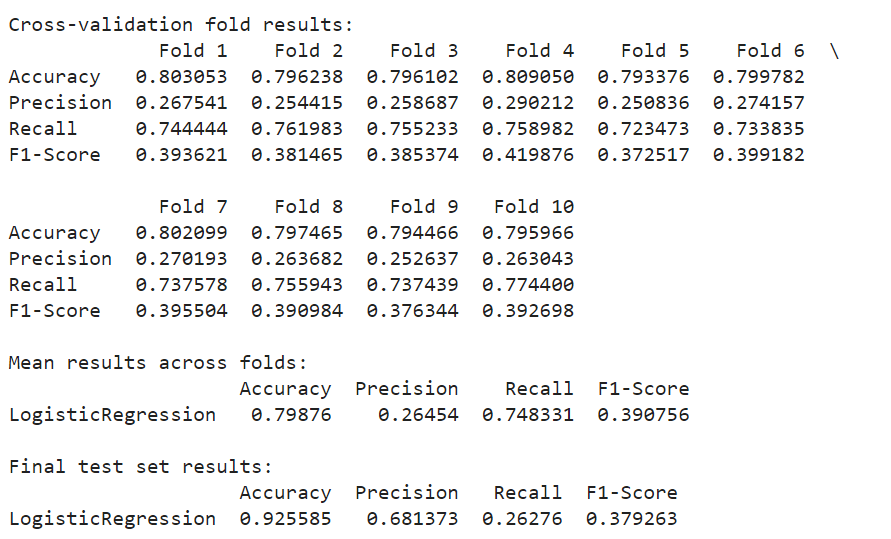
התיאור נוצר באופן אוטומטי

* **Classification Models:**
  1. **Model Selection:**
* Simple ML Models: Logistic Regression, KNN
* Ensemble ML Models: Bagging – Random Forest classifier, Boosting – ADABoost, Gradient Boosting Classifier.
* Simple vs Ensemble ML models for comparing the strength of using multiple vs single model.
* Bagging vs Boosting ML models for comparing the strength of learning models in parallel vs. in continuously.
  1. **Evaluation Metrics:**
* Accuracy, Precision, Recall and F1score.
* **Deployment Pipeline:**

*Note: learning curve and validation curves get unreasonable results and are time consuming*

Train a Baseline Model Without Any Tuning. Get understanding of model and data performance with Evaluation Metrics and learning curve.

* 1. **Internal Validation and Cross-Validation:**
  2. Pipeline:
* **10-Fold Cross Validation:** Split Training Set into 10 folds
* **Perform Cross Validation:** Train model on the Training Splits (Apply *Normalization*, *Balancing* and PCA for each fold) and evaluate it on the Validation Split
* **Calculate Performance** **metrics**; accuracy, precision, recall, F1score for each fold.
* **Average Results** calculate average performance metrics across all folds.
* **Test results** calculate total performance metrics for Testing Set.
  1. Results:

1. Logistic Regression:

*Discussion Results*

1. Accuracy**:** The accuracy on the test set is **0.925585**, which is significantly *higher than the cross-validation accuracy.*

2. Precision**:** The precision on the test set is **0.681373**, much *higher than the average precision during cross validation.*

3. Recall**:** The recall on the test set is **0.26276**, which is considerably *lower than the average recall from cross validation.*

4. F1Score**:** The F1Score on the test set is **0.379263**, which is also *lower compared to the cross-validation average.*

**1. Accuracy vs. Precision vs. Recall:**

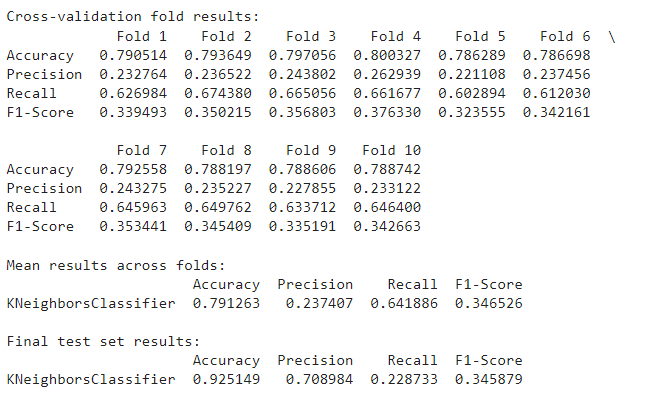
The test set accuracy is quite high, but the precision and recall present an imbalance. The high accuracy might be due to the model predicting the majority class well, but it seems to struggle with the minority class, which is reflected in the low recall.

Precision and recall are inversely related in many cases, and focusing on one often impacts the other.

**2. Class Imbalance:**

Given that the precision and recall are low, especially recall, it suggests that the model may be having trouble with the minority class, even after applying SMOTE and feature engineering.

1. KNN:



*Discussion Results*

* **Accuracy**: Accuracy on the test set is **0.925149**, which is substantially *higher* than the mean accuracy from cross validation.
* **Precision**: Precision on the test set is **0.708984**, significantly *higher* than the average precision during cross validation.
* **Recall**: Recall on the test set is **0.228733**, much *lower* than the average recall from cross validation.
* **F1Score**: F1Score on the test set is **0.345879**, which is *very close* to the mean F1Score from cross validation.

**1. Accuracy vs. Precision vs. Recall:**

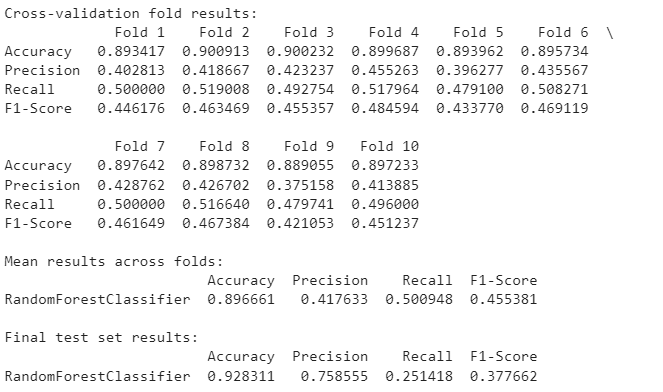
The test set accuracy is quite high, but the precision and recall present an imbalance. The high accuracy might be due to the model predicting the majority class well, but it seems to struggle with the minority class, which is reflected in the low recall.

**2. Class Imbalance:**

Given that the recall is low, it suggests that the model may be having trouble with the minority class, even after applying SMOTE and feature engineering.

Basically, same observations as done for logistic regression.

1. Random Forest Classifier:



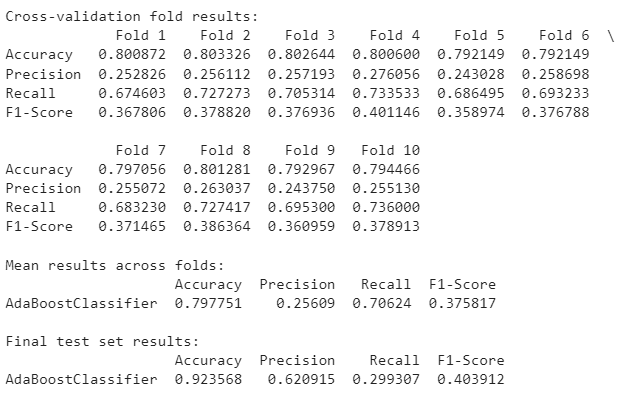
*Discussion Results*

* **Accuracy:** The model achieved a higher accuracy on the final test set **0.9283** compared to the cross-validation mean **0.8967**. This could indicate that the model generalized well to the unseen data.
* **Precision:** Precision increased significantly to **0.7586**, meaning that a higher proportion of predicted positive cases were correct.
* **Recall:** However, recall dropped to **0.2514**, indicating that the model missed many actual positive cases in the final test set.
* **F1-Score:** The F1-Score decreased to **0.3777**, reflecting the trade-off between precision and recall.

**1.Precision vs. Recall Trade-Off:** The final test set results indicate a trade-off between precision and recall. The model is more conservative in predicting positives on the final test set, leading to a higher precision but at the cost of lower recall.

**2.Consistency:** The increase in accuracy on the final test set suggests that the model might have benefited from better feature selection or hyperparameter tuning. However, the drop in recall highlights the model’s potential overfitting, focusing more on precision.

1. AdaBoost Classifier:



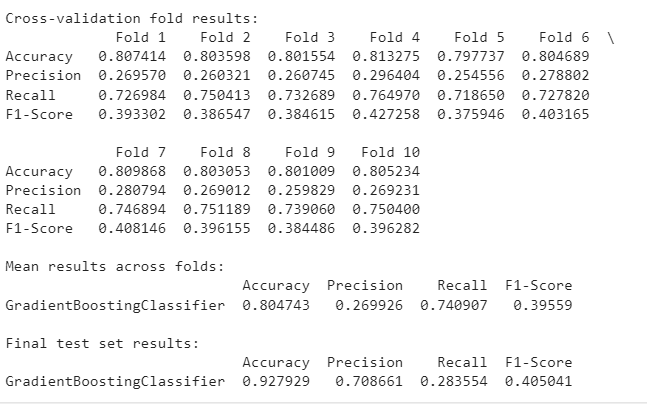
* **Accuracy:** The accuracy on the final test set is significantly higher **0.9236** than the cross-validation mean **0.7978**, indicating that the model performed better on the unseen test data.
* **Precision:** Precision on the final test set is also higher **0.6209**, suggesting that the model made fewer false-positive predictions compared to the cross-validation sets.
* **Recall:** The recall dropped to **0.2993** on the final test set, showing that the model missed a substantial portion of the actual positive cases in the test data.
* **F1-Score:** The F1-Score on the final test set is **0.4039**, which is slightly higher than the cross-validation mean **0.3758**. This indicates an improved balance between precision and recall on the test set, although recall dropped significantly.

1 **Improved Precision on Test Set:** The AdaBoost model showed a significant improvement in precision on the final test set, which is a positive outcome if reducing false positives was the goal.

2. **Decreased Recall on Test Set:** However, the decrease in recall on the final test set indicates that the model became more conservative in predicting positives, leading to more false negatives.

3. **Accuracy Trend:** The improvement in accuracy on the test set might suggest that the model generalizes well to unseen data, but the trade-off between precision and recall indicates some level of overfitting or a shift in the decision boundary.

1. Gradient Boosting *Classifier*:



* **Accuracy**: The accuracy on the final test set is **0.9279**, which is a significant improvement over the cross-validation mean. This indicates that the Gradient Boosting model performed well on unseen data.
* **Precision**: Precision on the final test set is much higher at **0.7087** compared to the cross-validation mean, suggesting a substantial reduction in false positives on the test data.
* **Recall**: The recall on the final test set dropped to **0.2836**, indicating that the model missed a substantial portion of the actual positive cases, similar to the trend seen with AdaBoost.
* **F1-Score**: The F1-Score on the final test set is **0.4050**, which is slightly higher than the cross-validation mean **0.3956**. This suggests that the model strikes a better balance between precision and recall on the test set compared to the cross-validation sets.

1.**Improved Precision on Test Set**: The Gradient Boosting model shows a significant improvement in precision on the final test set, like the AdaBoost model, which is beneficial if reducing false positives is the goal.

2.**Decreased Recall on Test Set**: The drop in recall on the final test set indicates that the model became more conservative in predicting positives, leading to more false negatives.

3.**Accuracy Trend**: The improved accuracy on the test set suggests that the Gradient Boosting model generalizes well to unseen data. However, the trade-off between precision and recall indicates a shift in the decision boundary.

**Models Performances Discussion**

* Logistic Regression provided stable performance with high precision on the test set but relatively low recall. It performed similarly to the GradientBoostingClassifier, with a slightly better F1Score on the test set.
* *KNeighborsClassifier* exhibited a reasonable performance in cross validation but showed a notable drop in recall on the test set. Precision was significantly higher on the test set, but the lower recall indicates a limitation in identifying positive instances correctly.
* RandomForestClassifier demonstrated strong performance across both cross validation and the test set. The model had high precision, but recall was lower, indicating that while it predicts positives accurately, it misses a fair amount of them.
* AdaBoostClassifier delivered a balanced performance in both cross validation and the test set. While precision and recall on the test set were moderate, the model provided a decent F1Score, indicating a good balance between precision and recall.
* GradientBoostingClassifier showed strong performance, particularly in cross validation, with good recall and balanced precision. The test set results maintained high precision, but like other models, the recall was lower than in cross validation.

**Conclusions**

1. Performance Consistency:

* All models demonstrated high accuracy on both cross validation and test sets, with the GradientBoostingClassifier and RandomForestClassifier leading slightly in overall performance.
* Precision was generally high across all models on the test set, indicating strong performance in identifying true positives with fewer false positives.

2. Precision vs. Recall Tradeoff:

* Precision was higher in most models, particularly in the RandomForest and GradientBoosting models, while recall was lower. This indicates that while the models are good at correctly identifying positive cases, they may miss some.

3. Model Suitability:

* **RandomForestClassifier** and **GradientBoostingClassifier** stood out as the top performers, offering a good balance between precision and recall, with higher F1Scores on the test set.
* Logistic Regression also performed well, especially in precision, making it a strong candidate if interpretability and simpler model structure are priorities.

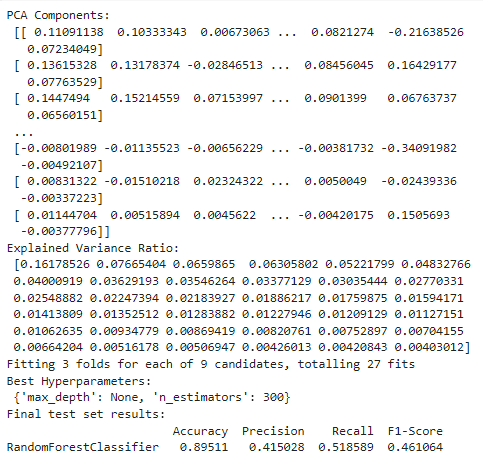
Use GridSearchCV for the 2 best models for Efficient Search. Monitoring hyperparameters performance and importance with Evaluation Metrics and Validation Curves.

Applying best choose hyperparameters and getting performance with Evaluation Metrics and with Confusion Matrix, ROC Curve

**2.External Evaluation:**

*Moving forward to next analysis with* ***RandomForestClassifier*** *and* ***GradientBoostingClassifier***

1. **RandomForestClassifier**



1. **Best Hyperparameters:**

* **max\_depth = None:** This means that the trees in the Random Forest are allowed to grow to their full depth, without any limitation. This can capture more complex patterns in the data but also increases the risk of overfitting.
* **n\_estimators = 300**: The model uses 300 decision trees in the ensemble, which is a relatively high number. More trees generally improve the model's performance by reducing variance, leading to a more stable and robust model.

2. **Performance Metrics:**

* **Accuracy**: 0.89511: The model correctly classified about 89.5% of the test instances. While this seems high, accuracy alone can be misleading, especially in cases of imbalanced datasets.
* **Precision**: 0.415028: Precision is the proportion of true positive predictions among all positive predictions. A precision of 0.415028 indicates that when the model predicts a positive class, it's correct about 41.5% of the time. This suggests that the model has a relatively high number of false positives.
* **Recall**: 0.518589: Recall measures the proportion of actual positive cases that were correctly identified by the model. A recall of 0.518589 means the model correctly identified about 51.9% of all true positive cases. This indicates that the model misses a significant portion of actual positive cases (false negatives).
* **F1Score**: 0.461064: The F1score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. An F1score of 0.461064 indicates moderate performance, with room for improvement in both precision and recall.

3. **Conclusions**:

* Model Complexity: With 'max\_depth = None', the model is very complex, potentially leading to overfitting. However, the relatively good accuracy suggests that the model captures the underlying patterns well. The tradeoff is that the model's precision and recall are lower, indicating it may struggle with the specific identification of positive cases.
* Precision vs. Recall: The precision is lower than recall, implying the model generates more false positives than false negatives. Depending on the context of the problem, this may or may not be acceptable. For instance, if the cost of a false positive is high, improving precision would be necessary.
* Imbalanced Data: If the dataset is imbalanced, with significantly more instances of one class than the other, this can explain the relatively low precision and recall. Random Forest might be biased toward predicting the majority class, leading to lower performance on the minority class.

**Confusion Matrix**

**תמונה שמכילה טקסט, צילום מסך, מלבן, תרשים

התיאור נוצר באופן אוטומטי**

**Confusion Matrix Interpretation:**

* True Positives (TP = 15,596): The model correctly predicted 15,596 positive cases.
* False Positives (FP = 764): The model incorrectly predicted 764 cases as positive when they were actually negative. These are false alarms or Type I errors.
* False Negatives (FN = 1,160): The model incorrectly predicted 1,160 cases as negative when they were actually positive. These are missed detections or Type II errors.
* True Negatives (TN = 823): The model correctly predicted 823 negative cases.

**ROC Curve**

תמונה שמכילה קו, טקסט, תרשים, עלילה

התיאור נוצר באופן אוטומטי

**Interpretation:**

* An AUCROC of 0.87 indicates that the model has a good ability to distinguish between the positive and negative classes.

This means that, in approximately 87% of the cases, the model will correctly rank a randomly chosen positive instance higher than a randomly chosen negative instance.

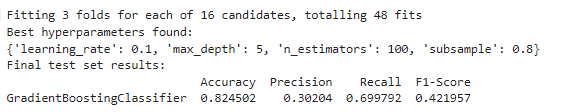
**Performance:**

* A score of 0.87 is generally considered strong, especially in complex datasets.

**Contextual Analysis:**

* Given that model has a precision of around 0.415028 and recall of 0.518589, the AUCROC of 0.87 reinforces that the model's overall discriminatory power is solid, even if precision might seem lower in specific threshold settings.

**2.Gradient Boosting Classifier**



1. **Best Hyperparameters:**

1. **Learning Rate: 0.1** provides a good balance between training time and model performance.

2.**Max Depth: 5** A max\_depth of 5 is moderate, allowing the model to capture relatively complex relationships without being overly complex.

3.**Number of Estimators: 100** a balanced choice.

4.**Subsample: 0.8** introduces some randomness while still using a significant portion of the data. It’s a good starting point for balancing bias and variance.

2. **Metrics:**

1. Accuracy (0.824502): The model's accuracy is relatively high, indicating that it correctly classified about 82.45% of the instances in the test set.

2. Precision (0.30204): Precision is quite low at 0.30204. This implies that out of all the instances that the model predicted as the positive class (e.g., '1'), only 30.2% were actually correct. A low precision indicates that the model is generating a high number of false positives, predicting the positive class when it should not.

3. Recall (0.699792): Recall is 0.699792, which means that the model is able to correctly identify about 70% of the actual positive instances. While this is a decent recall, it suggests that the model is missing around 30% of the actual positive cases, resulting in a moderate number of false negatives.

4. F1-Score (0.421957): The F1-Score is 0.421957, a harmonic mean of precision and recall. Given that it’s closer to recall than precision, this indicates that while the model is better at capturing the positive class, it does so at the expense of a high number of false positives. The moderate F1-Score highlights the trade-off between precision and recall, reflecting the balance between these two metrics.

1.High Accuracy but Low Precision:

The model’s high accuracy does not necessarily reflect a strong performance in terms of precision and recall, particularly precision. This discrepancy suggests that accuracy alone may not be a sufficient measure of the model's effectiveness, especially in cases where the cost of false positives is high.

2.Trade-off Between Precision and Recall:

The results indicate a trade-off between precision and recall, which is typical in imbalanced datasets. The model favors identifying more positive instances (higher recall) at the cost of wrongly classifying negatives as positives (lower precision).

The GradientBoostingClassifier with the given parameters shows a model that is relatively accurate but has a significant trade-off between precision and recall.

**Confusion Matrix Interpretation:**

* True Positives (TP = 11,955): The model correctly predicted 11,955 positive cases.
* False Positives (FP = 432): The model incorrectly predicted 432 cases as positive when they were actually negative. These are false alarms or Type I errors.
* False Negatives (FN = 2327): The model incorrectly predicted 2327 cases as negative when they were actually positive. These are missed detections or Type II errors.
* True Negatives (TN = 1007): The model correctly predicted 1007 negative cases.

תמונה שמכילה טקסט, צילום מסך, מלבן, גופן

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, קו, תרשים, עלילה

התיאור נוצר באופן אוטומטי

**Interpretation:**

* An AUCROC of 0.86 indicates that the model has a good ability to distinguish between the positive and negative classes.

This means that, in approximately 86% of the cases, the model will correctly rank a randomly chosen positive instance higher than a randomly chosen negative instance.

**Performance:**

* A score of 0.86 is generally considered strong, especially in complex datasets.

**Contextual Analysis:**

* Given that model has a precision of around 0.30204 and recall of 0.699792, the AUCROC of 0.86 reinforces that the model's overall discriminatory power is solid, even if precision might seem lower in specific threshold settings.

Getting results for submission

Need to add results here from the files