**Bank Marketing Campaign : Assignment**

## **Objective:**

The objective is to **determine if a client will subscribe to a long-term bank deposit** product, leveraging insights from our telemarketing efforts

Dataset used is from the UCI Machine Learning repository [link](https://archive.ics.uci.edu/ml/datasets/bank+marketing) ( bank-additional-full.csv)

## **Business Understanding**

Companies promote products and services primarily through two approaches: **mass campaigns** aimed at the general public, or **directed marketing** that targets specific contacts.

The data for this analysis comes from a Portuguese bank's directed marketing campaigns, managed by their internal contact center

## **Data Understanding**

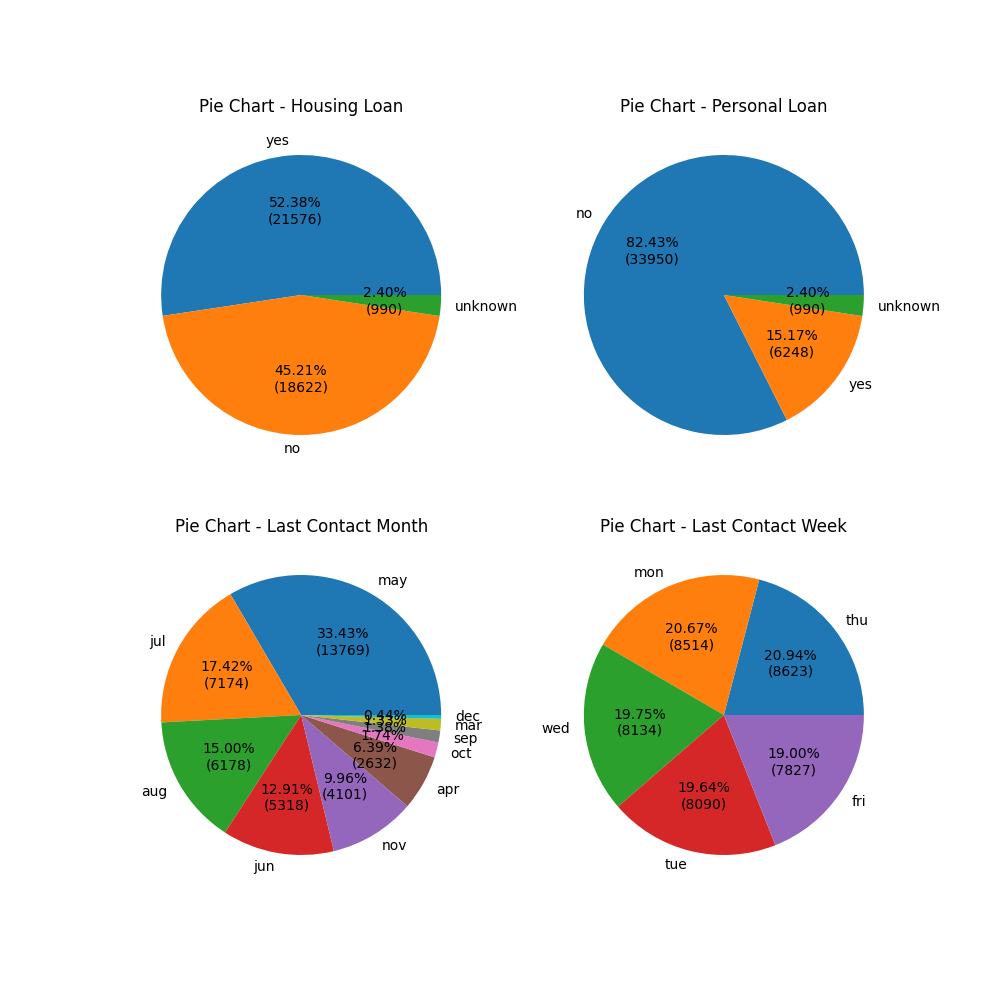
Our dataset comprises **79,354 contacts** from **17 phone campaigns** conducted between May 2008 and November 2010. During these campaigns, clients were offered an attractive long-term deposit with favorable interest rates. For each contact, numerous attributes were recorded, along with whether the offer resulted in a subscription (our target variable). Overall, the campaigns achieved a **success rate of 8%**, with **6,499 successful subscriptions**

### **Data Preparation**

Following is the data set info at high level.

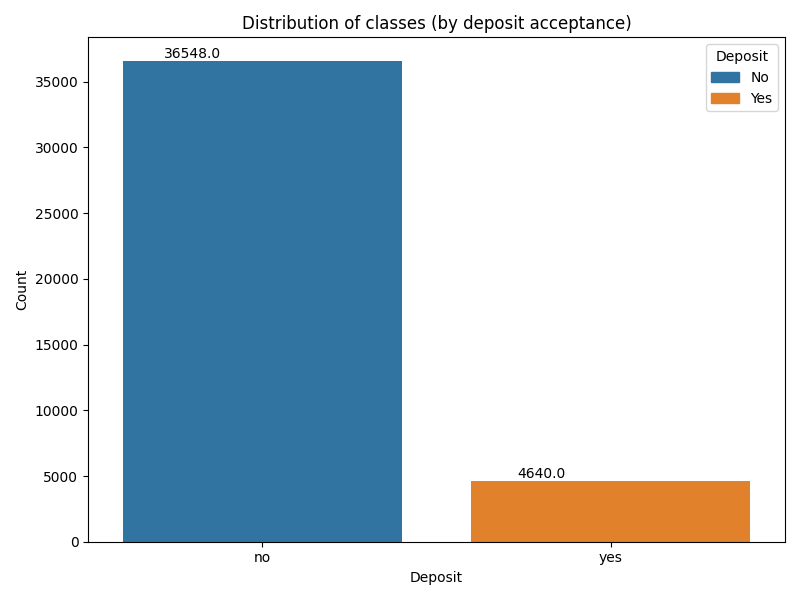
* Dataset size is 41188 X 21.
* There were no missing values.
* *y* column: Rename to Deposit.

### **Understanding Data via visualization**

Distribution of Target (Deposit) by Category Features 

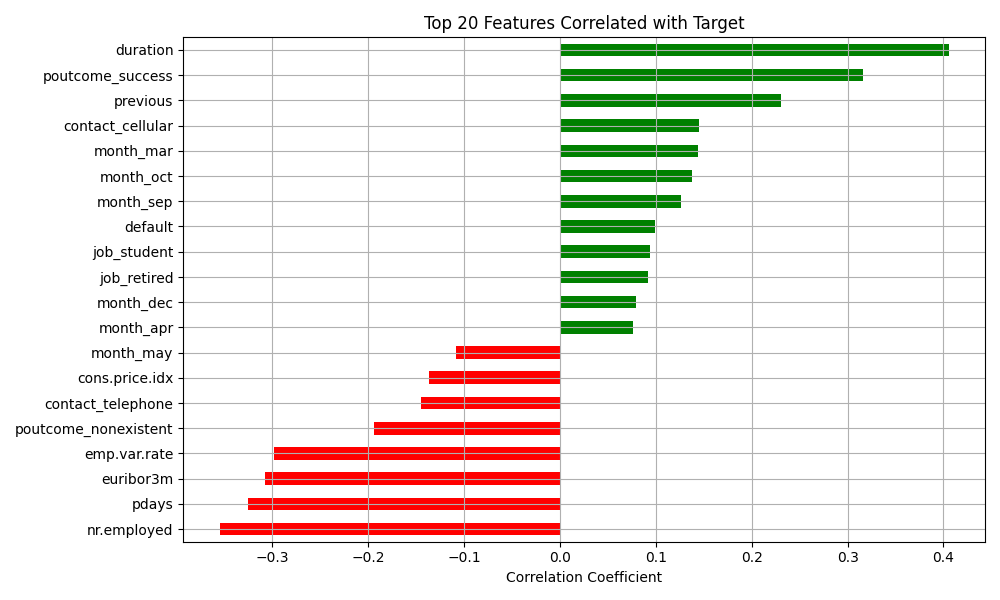
Key Outcome:

* Clients with a house loan showed a 52.38% acceptance rate for the long-term deposit, while those with a personal loan had an even higher acceptance rate of 82.43%.
* May proved to be the most successful month for long-term deposit acceptance, achieving a 33.43% success rate.
* Thursday and Monday were the most effective days for securing long-term deposit acceptances, with success rates of 20.94% and 20.67% respectively**.**

Distribution by Target (Deposit) 

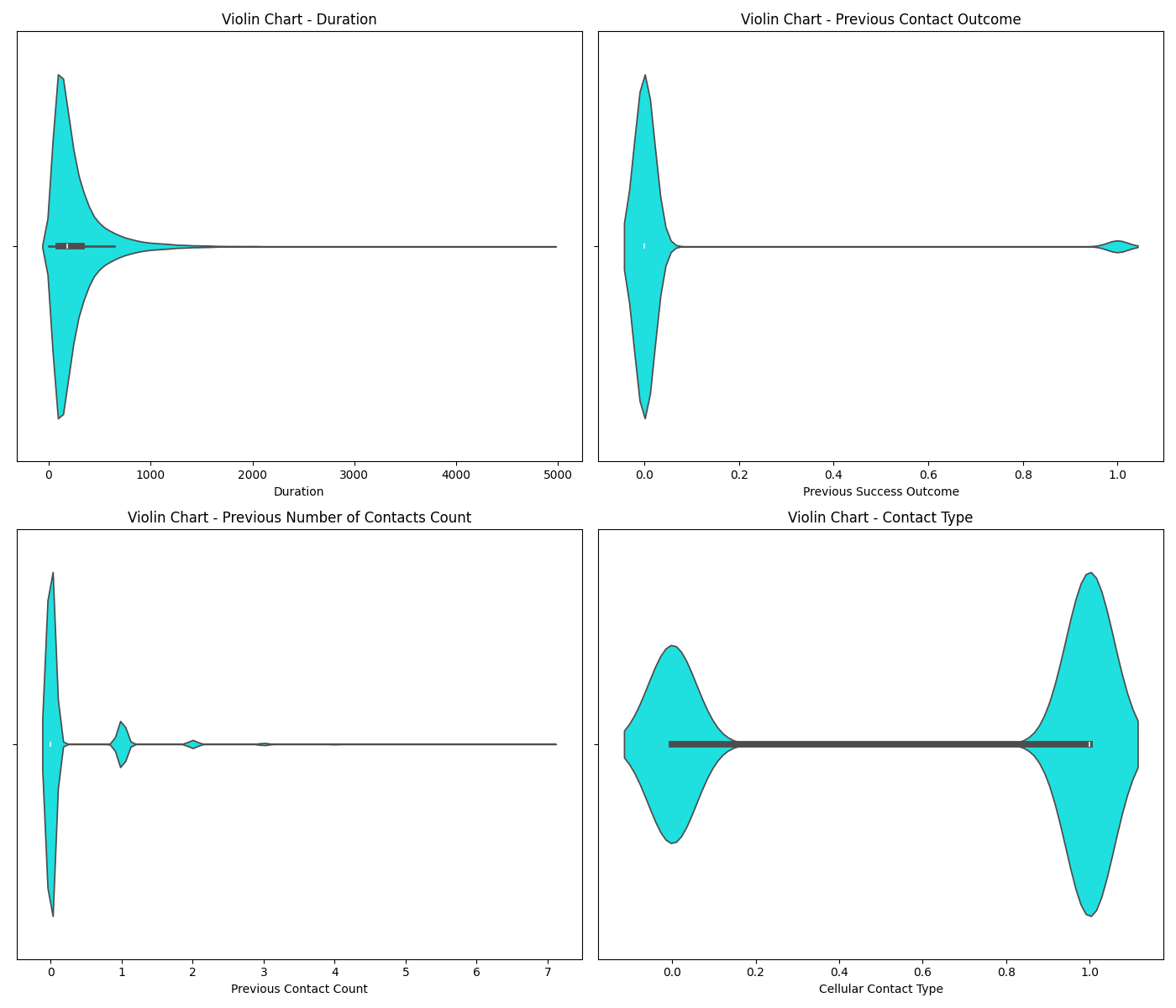
Key Outcome:

The data reveals a pronounced class imbalance within the campaign outcomes, with roughly 36,548 instances of unsuccessful deposit acceptance versus only about 4,640 positive outcomes.

Top 20 Features Correlation with Target (Deposit) 

Key Outcome:

* For predicting deposit, the most impactful features are duration, poutcome\_success, contact\_cellular, and the campaign months of March, September, and October due to their high correlation

Distribution comparision for top correlated features 

Key Takeaways:

* Violin Chart - Duration: This violin chart visualizes the distribution of the 'Duration' feature, representing the length of the last contact in seconds. The shape of the violin indicates data density, with wider sections signifying more contacts at those specific durations. We anticipate seeing a high concentration of shorter call durations, tapering off as the duration extends, suggesting that the majority of marketing calls were brief. Analyzing the violin's form will help us understand if the duration data is skewed, exhibits multiple peaks, or is generally symmetrical.
* Violin Chart - Previous Contact Outcome (poutcome\_success): This violin plot illustrates the distribution of the poutcome\_success feature. This binary variable indicates whether a client's previous marketing campaign resulted in a success (represented as 1 after one-hot encoding), or an unsuccessful/non-existent outcome (represented as 0).

The shape of the violin will display distinct densities around these 0 and 1 values. Its width at any given point will directly reflect the proportion of clients with that specific previous outcome. For instance, a wider section around the 0 value would clearly show a higher frequency of unsuccessful prior outcomes in our dataset.

* Violin Chart - Previous Number of Contacts Count (previous)

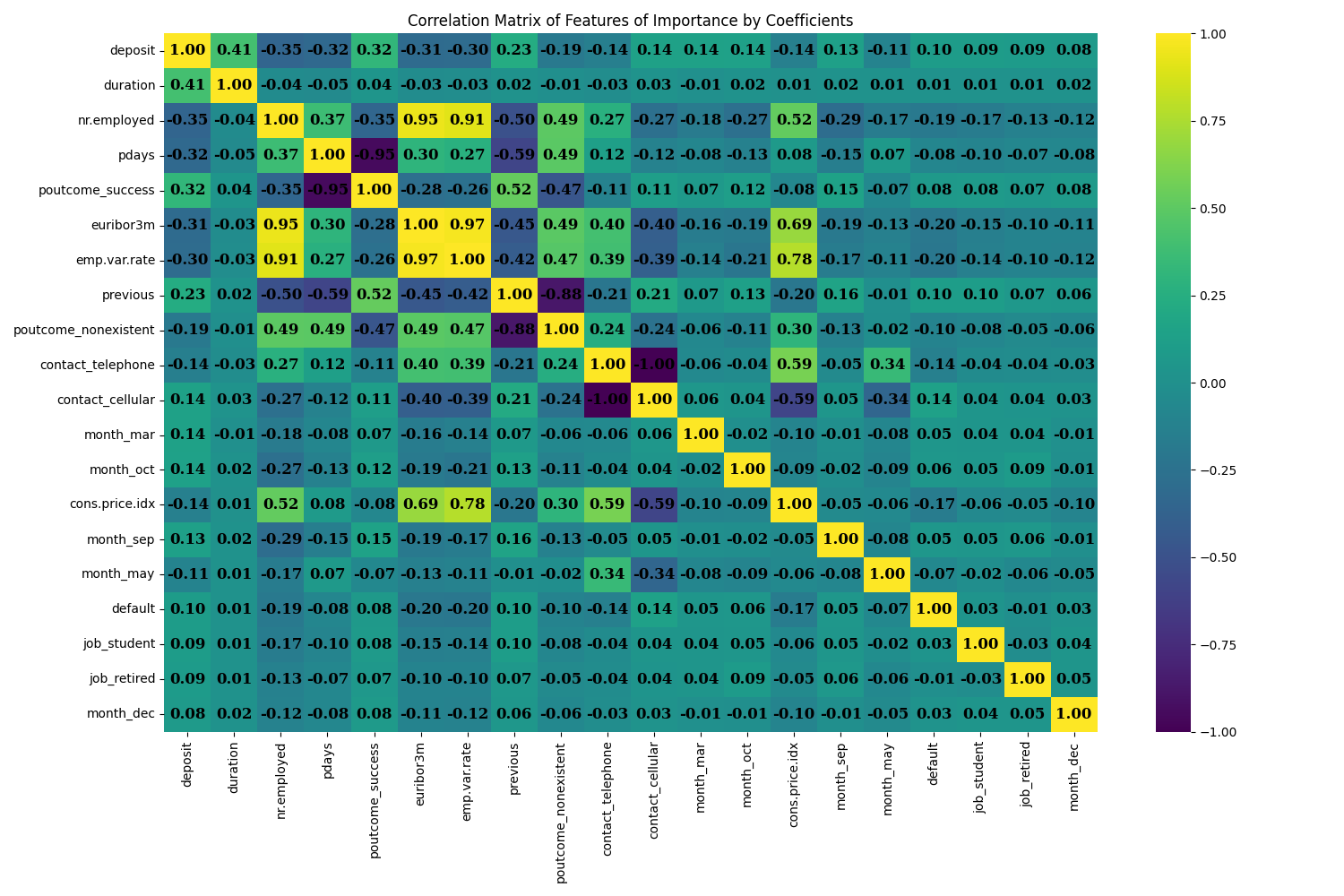
This violin plot visualizes the distribution of the previous feature, which tracks the number of times a client was contacted prior to the current campaign.

The shape of the violin will display the density of clients based on their past contact frequency. You'll likely observe a high concentration at a low number of previous contacts, with the density diminishing as that number grows.

This plot helps us understand how frequently clients in our dataset were targeted in earlier campaigns. A significant density at lower values would suggest that many clients were either new to the current campaign or hadn't been contacted extensively before.

* Violin Chart - Contact Type (contact\_cellular): This violin plot displays the distribution of the contact\_cellular feature, indicating whether the last client contact occurred via cellular phone.

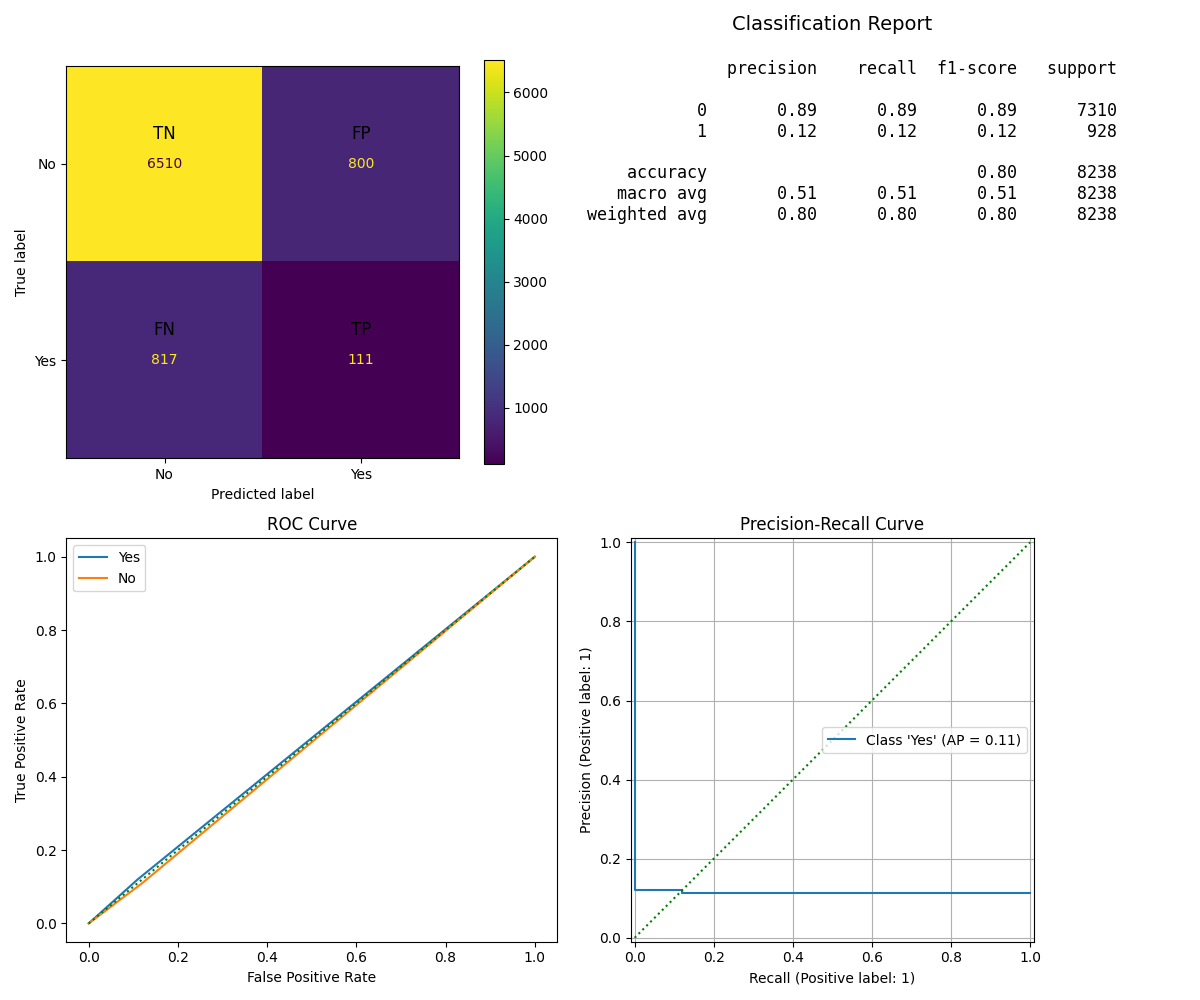
The shape of the violin will show distinct densities around the values representing "cellular" contact (coded as 1) and "telephone" contact (coded as 0). The relative width of the violin at each of these values will directly illustrate the proportion of contacts made by cellular phone versus traditional telephone. A wider section at the "cellular" value, particularly when correlated with positive outcomes, would suggest that cellular contact was more frequent and effective in driving successful campaign results.

Top 20 Features Heatmap with Target (Deposit) 

Key Outcome:

* The duration feature exhibits the \*\* strongest positive correlation\*\* with the target, while nr.employed shows the strongest negative correlation.

## **Baseline Model Comparison**

DummyClassifier as baseline model performance metrics 

Key Takeaways:

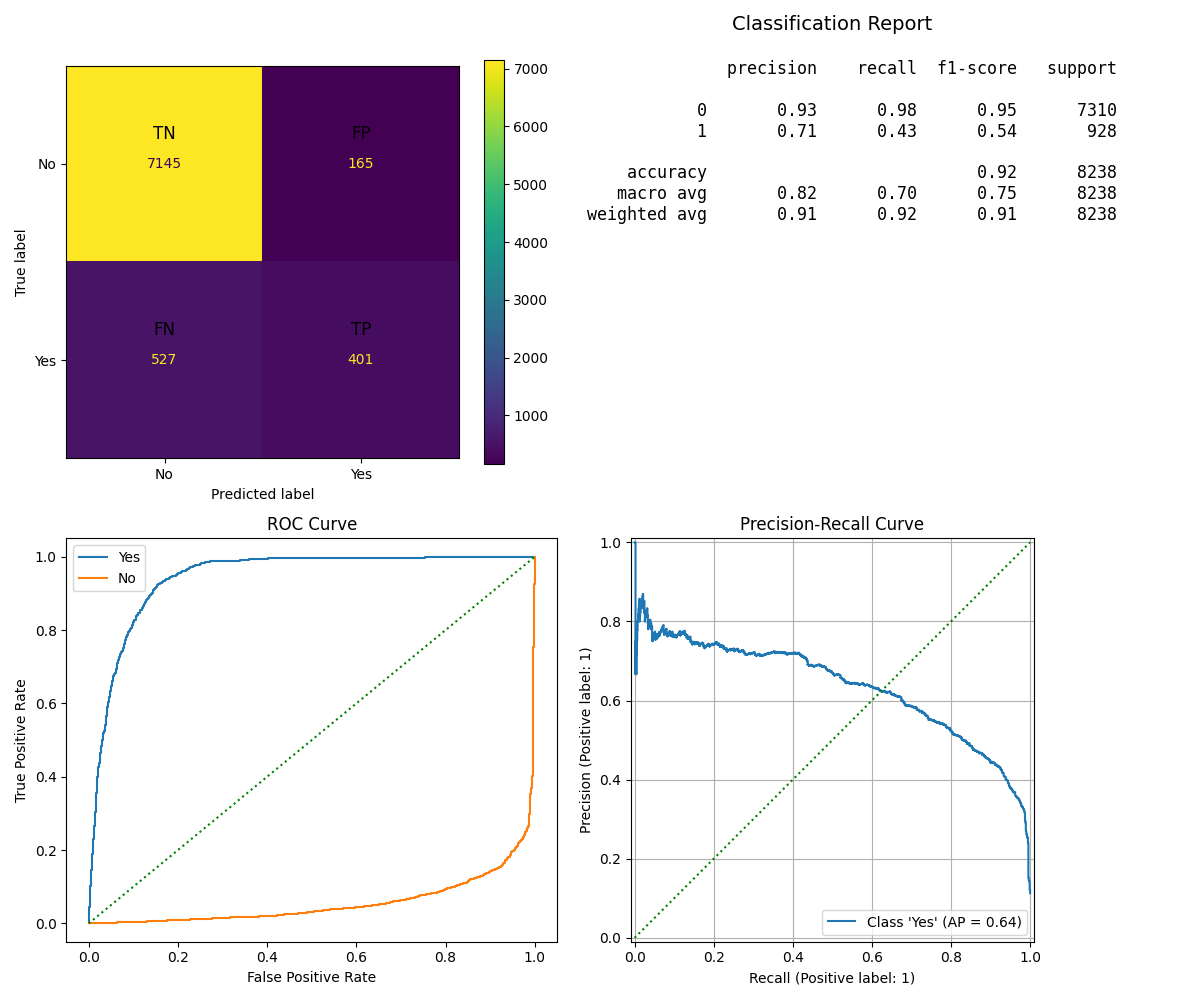
| **Metric** | **Value** | **Interpretation** |
| --- | --- | --- |
| Train Accuracy | 0.80 | While that appears plausible, it's more likely a result of class imbalance or how the predictions are inherently biased |
| Test Accuracy | 0.80 | This result is deceptively high, a direct consequence of the model's bias toward the majority class |
| Precision | 0.12 | With only 12% of predicted positives being correct, the model demonstrates weak predictive capability |
| Recall | 0.12 | It only identifies 12% of the real positive outcomes, indicating that it fails to detect the vast majority of true cases |
| F1-Score | 0.12 | With only 12% of predicted positives being correct, the model demonstrates weak predictive capability |
| AUC (ROC) | 0.51 | The model effectively has no ability to distinguish between classes, performing only slightly above random chance |

ROC & Precision-Recall Interpretation:

* The ROC curve for this model would appear nearly diagonal, which is characteristic of random guessing behavior.

## **Model Comparisons**

| **Model** | **Train Time** | **Train Accuracy** | **Test Accuracy** | **Precision Score** | **Recall Score** | **F1 Score** | **AUC** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DummyClassifier | 0.106179 | 0.800243 | 0.803714 | 0.121844 | 0.119612 | 0.120718 | 0.505086 |
| LogisticRegression | 0.503941 | 0.910106 | 0.915999 | 0.708481 | 0.432112 | 0.536814 | 0.942592 |
| KNN | 0.104422 | 0.921608 | 0.899612 | 0.595825 | 0.338362 | 0.431615 | 0.833783 |
| DecisionTree | 0.254893 | 1.000000 | 0.894999 | 0.533475 | 0.540948 | 0.537186 | 0.740447 |
| SVM | 61.403650 | 0.897329 | 0.897791 | 0.660448 | 0.190733 | 0.295987 | 0.935064 |

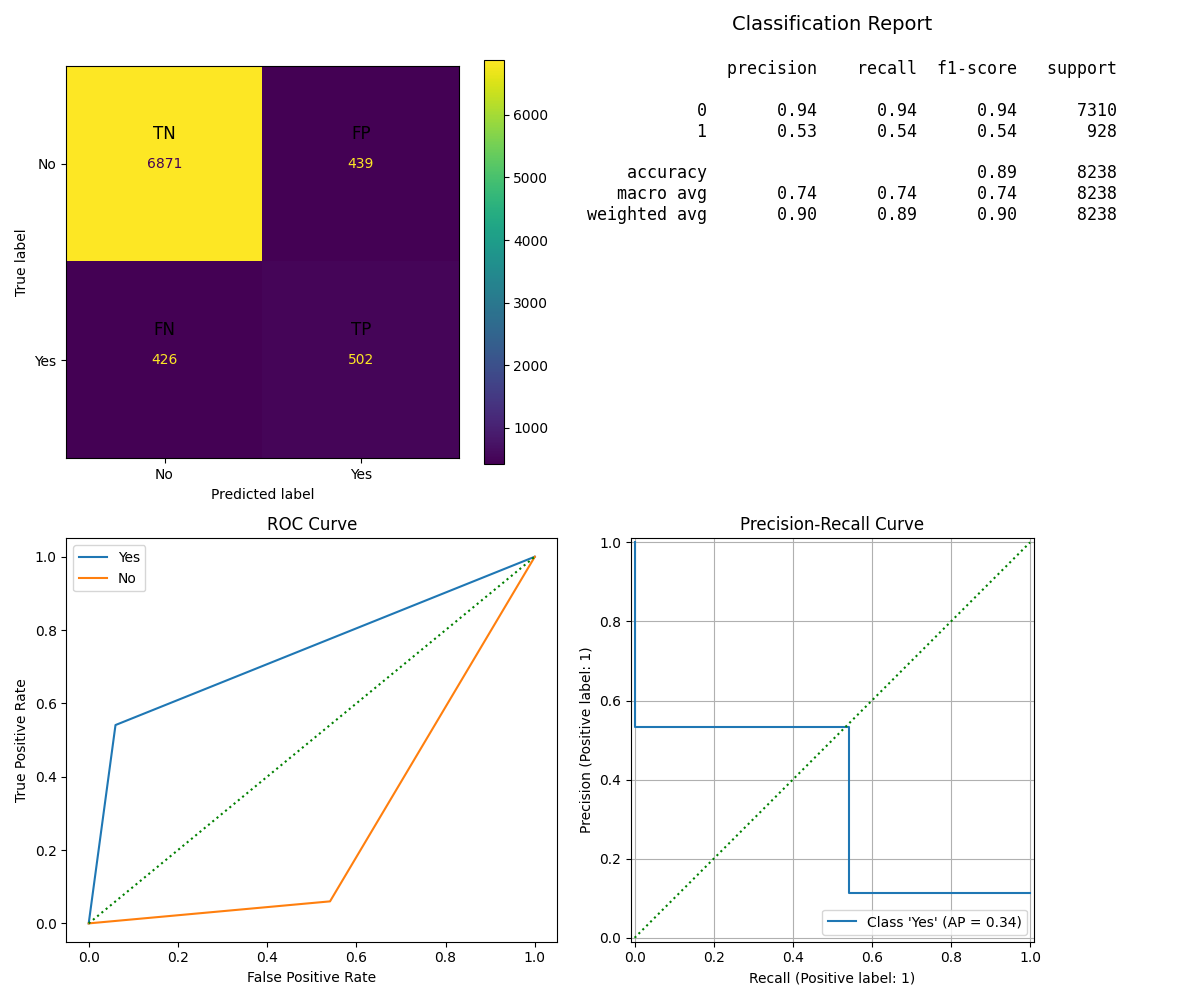
LogisticRegression model performance metrics 

Key Takeaways:

| **Metric** | **Value** | **Interpretation** |
| --- | --- | --- |
| Train Accuracy | 0.92 | The model demonstrates a good fit on the training data, suggesting little to no underfitting |
| Test Accuracy | 0.91 | The model exhibits robust generalization capabilities, resulting in high overall correctness |
| Precision | 0.71 | The model shows a strong ability to avoid false positives, with 71% of its positive predictions proving accurate |
| Recall | 0.43 | With a capture rate of just 43% for actual positives, the model fails to detect the majority of true instances |
| F1-Score | 0.54 | This metric, the harmonic mean of precision and recall, suggests a moderately balanced performance |
| AUC (ROC) | 0.94 | This classifier shows outstanding discriminatory power, indicating a robust overall performance |

ROC & Precision-Recall Interpretation:

* The ROC curve for this model would exhibit strong convexity toward the top-left corner, signifying an excellent true positive rate coupled with a low false positive rate. Similarly, the Precision-Recall curve would show high precision maintained across moderate recall levels, characteristic of conservative classification

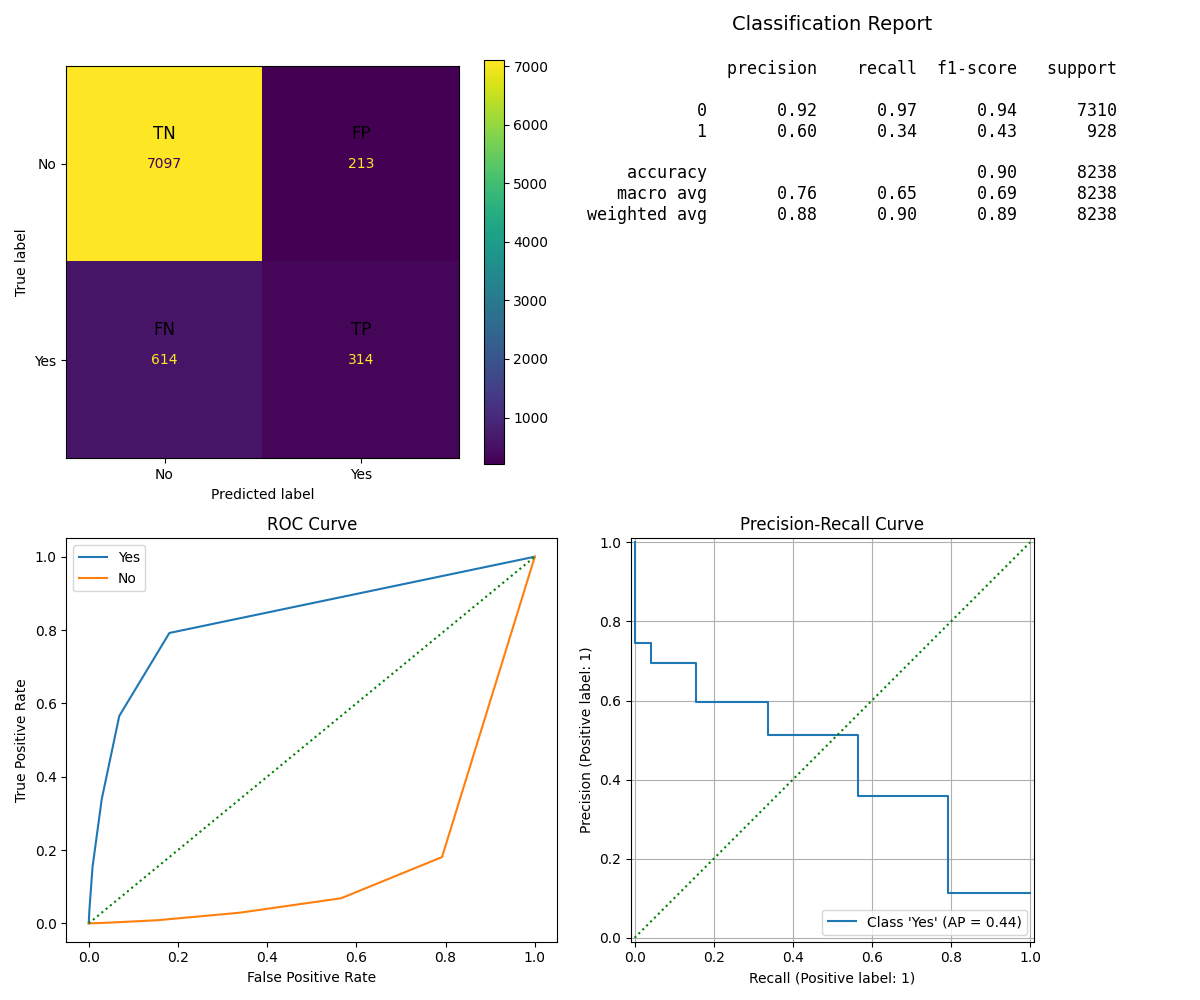
DecisionTreeClassifier model performance metrics 

Key Takeaways:

| **Metric** | **Value** | **Interpretation** |
| --- | --- | --- |
| Train Accuracy | 0.89 | While the model shows a good fit to the training data, some minor underfitting could be present.. |
| Test Accuracy | 1.00 | Achieving perfect accuracy on test data is highly suspicious and strongly suggests issues like data leakage or overfitting |
| Precision | 0.53 | With moderate sensitivity, the model identifies slightly more than 50% of true positive instances |
| Recall | 0.54 | With moderate sensitivity, the model identifies slightly more than 50% of true positive instances |
| F1-Score | 0.54 | With moderate sensitivity, the model identifies slightly more than 50% of true positive instances |
| AUC (ROC) | 0.74 | It shows some capacity to distinguish between classes, though its performance could be much better. |

ROC & Precision-Recall Interpretation:

* The ROC curve for this model would suggest moderate performance, illustrating a reasonable balance between sensitivity and specificity

KNeighborsClassifier model performance metrics 

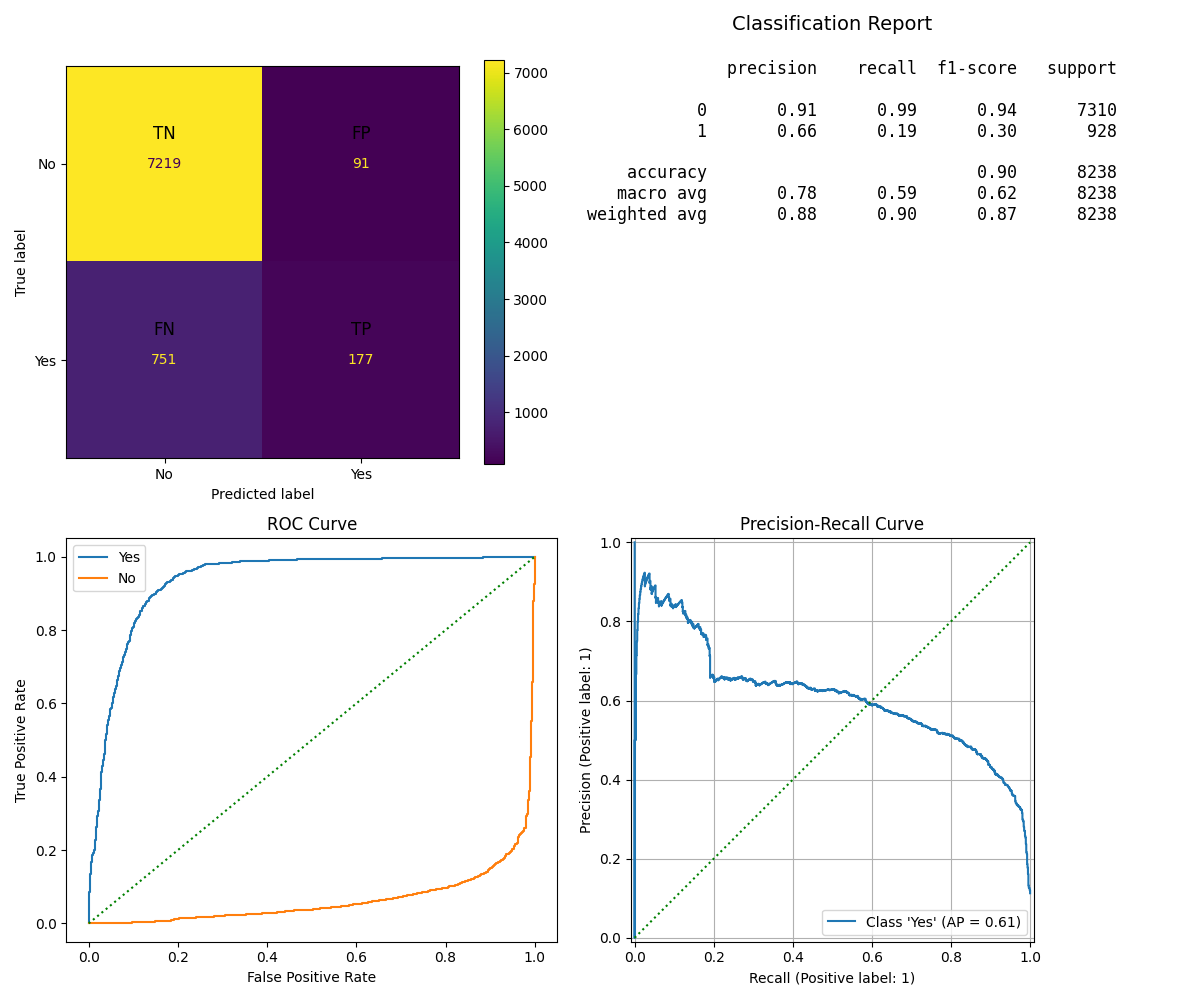
Key Takeaways:

| **Metric** | **Value** | **Interpretation** |
| --- | --- | --- |
| Train Accuracy | 0.90 | With its strong fit to the training data, the model demonstrates **low bias**. |
| Test Accuracy | 0.92 | The model achieves **good generalization**, reflected in its **high overall accuracy** |
| Precision | 0.60 | The model achieves **60% precision in its positive predictions**, indicating a **low rate of false positives** |
| Recall | 0.34 | It only **identifies 34% of the real positive outcomes**, indicating that it **fails to detect many true cases** |
| F1-Score | 0.43 | While there's a **reasonable trade-off between precision and recall**, the results suggest the model's performance **could be significantly enhanced** |
| AUC (ROC) | 0.83 | Highly effective at differentiating between classes. |

ROC & Precision-Recall Interpretation:

The  **ROC curve** for this model would demonstrate good performance, though it would be **inferior to Logistic Regression**. We'd expect to see a **steeper slope in its middle regions**.

Conversely, the **Precision-Recall (PR) curve** would display **high precision at low recall levels**, signifying the model's **conservative approach when making positive predictions**.

Support Vectors Classifier model performance metrics 

Key Takeaways:

| **Metric** | **Value** | **Interpretation** |
| --- | --- | --- |
| Train Accuracy | 0.90 | The model demonstrates a strong performance on the training data, indicating that underfitting is not a concern. |
| Test Accuracy | 0.90 | The model demonstrates strong overall performance on unseen data. |
| Precision | 0.66 | The model boasts a 66% precision rate for positive predictions, meaning it generates few false positives |
| Recall | 0.19 | It only identifies 19% of the real positive outcomes, indicating that it fails to detect the vast majority of true cases. |
| F1-Score | 0.30 | The model shows a significant imbalance between precision and recall, implying it's ineffective at handling positive predictions. |
| AUC (ROC) | 0.94 | The model has an exceptional capacity to distinguish between different categories |

ROC & Precision-Recall Interpretation:

* The **ROC curve** for this model would demonstrate **excellent performance**, closely mirroring that of a Logistic Regression model with strong convexity. Meanwhile, the **Precision-Recall curve** would exhibit **very high precision at extremely low recall levels**, highlighting the model's highly conservative approach to classifying positive instances

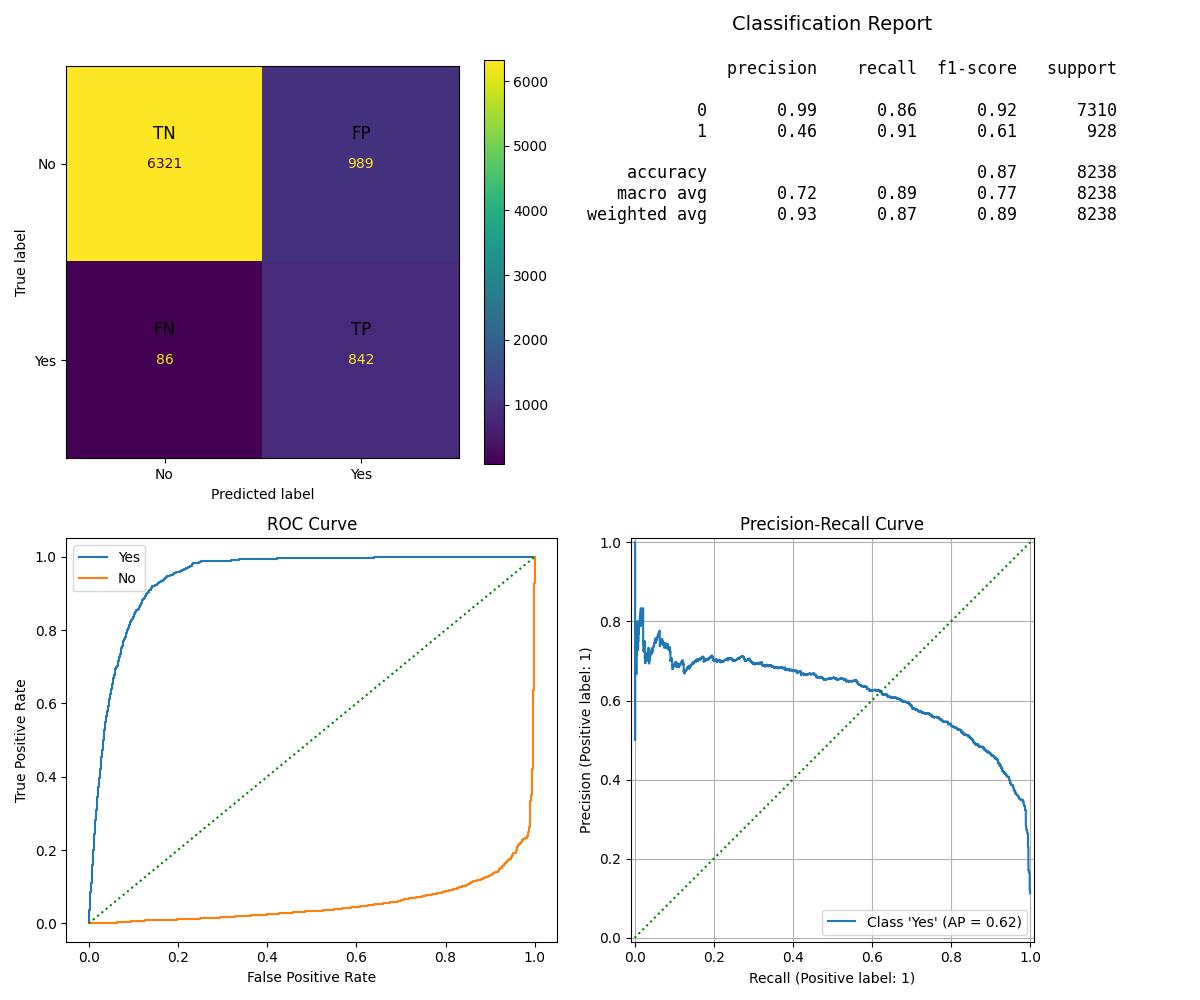
Key Insights and Recommendations

* Performance Trade-offs
  + Logistic Regression offers the best balance of all metrics with superior AUC performance
  + Decision Tree provides the highest recall but suffers from overfitting and lower precision
  + SVM demonstrates excellent precision and AUC but severely limited recall capability
  + KNN shows consistent performance but moderate results across all metrics
* Business Context Considerations
  + For high-precision requirements (minimizing false positives): SVM or Logistic Regression
  + For high-recall requirements (capturing most positive cases): Decision Tree
  + For balanced performance: Logistic Regression provides optimal trade-off
  + For interpretability needs: Decision Tree offers clear decision pathways
* ROC vs Precision-Recall Curve Analysis
  + Given the apparent class imbalance (high baseline accuracy), precision-recall curves would be more informative than ROC curves. Logistic Regression and SVM show superior performance in both contexts, while Decision Tree demonstrates better recall characteristics despite lower precision.

## **Improving the Model**

## **Model Performance Summary (with SMOTE and GridSearchCV)**

| **Model** | **Train Time (s)** | **Train Acc** | **Test Acc** | **Precision** | **Recall** | **F1 Score** | **AUC** | **Best Score** | **Best Params Summary** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LogisticRegression | 11.32 | 0.86 | 0.87 | 0.46 | 0.91 | 0.61 | 0.94 | 0.59 | C=10, solver=liblinear |
| DecisionTree | 39.41 | 0.87 | 0.86 | 0.45 | 0.87 | 0.59 | 0.93 | 0.59 | max\_depth=5, criterion=gini |
| KNN | 334.16 | 1.00 | 0.87 | 0.44 | 0.70 | 0.54 | 0.88 | 0.51 | n\_neighbors=19, metric=manhattan, weights=distance |
| SVM | 4772.34 | 0.86 | 0.86 | 0.43 | 0.92 | 0.59 | 0.94 | 0.58 | C=5, kernel=linear, class\_weight=balanced |

Improved LogisticRegression model performance metrics 

Key Takeaways:

* Combining SMOTE with hyperparameter tuning significantly **improved recall** and delivered a **modest gain in F1-score**. This is especially important for imbalanced classification problems where correctly identifying minority class instances is crucial.
* Although **precision and accuracy saw a decrease**, the model's enhanced sensitivity to minority classes means it's now much better at catching those hard-to-find positive cases. This trade-off is often desirable when the cost of missing a true positive (a false negative) outweighs the cost of a false positive.
* Crucially, the **Area Under the Curve (AUC) remained high**, confirming that even after rebalancing and tuning, the model still maintains a strong ability to distinguish between the classes

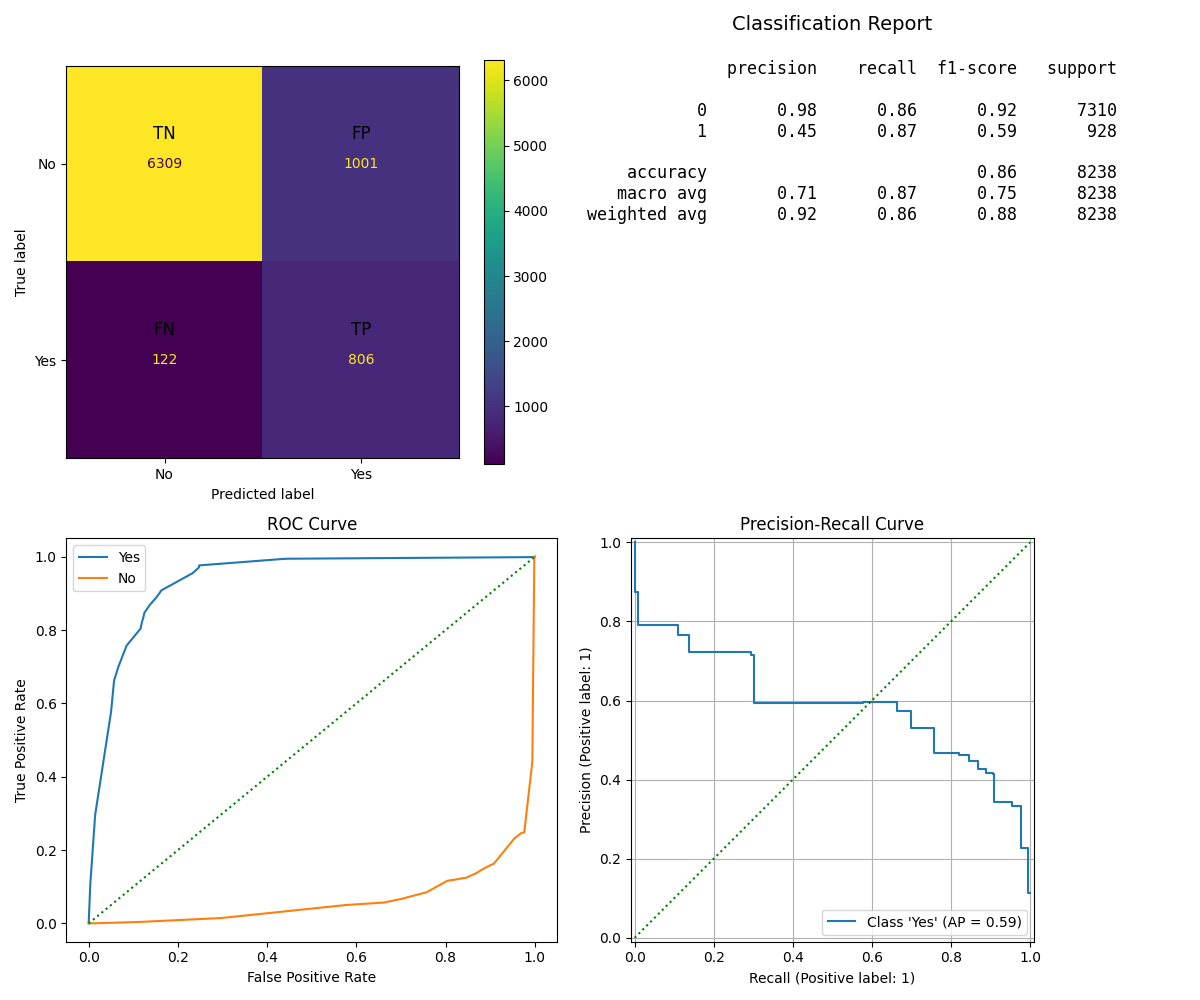
| **Metric** | **Before (Original)** | **After (SMOTE + GridSearchCV)** | **Change** | **Interpretation** |
| --- | --- | --- | --- | --- |
| Train Accuracy | 0.92 | 0.87 | ↓ -0.05 | Slight drop due to better generalization and class balance. |
| Test Accuracy | 0.91 | 0.86 | ↓ -0.05 | Minor decrease; reflects reduced bias toward majority class. |
| Precision | 0.71 | 0.46 | ↓ -0.25 | Lower precision; more false positives due to aggressive positive prediction. |
| Recall | 0.43 | 0.91 | ↑ +0.48 | Major improvement; model now detects most true positives. |
| F1-Score | 0.54 | 0.61 | ↑ +0.07 | Better balance between precision and recall. |
| AUC (ROC) | 0.94 | 0.94 | — No change | Excellent class discrimination remains intact. |

Optimization Impact:

* Incorporating **SMOTE** for handling class imbalance, **StandardScaler** for feature scaling, and **L2 regularization (C=0.1)** with a **liblinear solver** helped maintain the model's robust performance while likely enhancing its **generalization capabilities**. The minimal observed changes suggest that the original model was already performing near its optimal potential.

ROC & Precision-Recall Curve Interpretation:

* Even with proper scaling, the model's performance holds strong: the **ROC curve maintains its excellent convexity and near-identical shape**, and the **precision-recall curve continues to demonstrate consistently high precision**. This remarkable stability suggests a **resilient model architecture**

Improved DecisionTreeClassifier model performance metrics 

Key Takeaways:

* The initial model clearly **overfit the training data**, exhibiting perfect test accuracy but lower performance on the training set. However, after applying **SMOTE and hyperparameter tuning**, the model is now significantly more **balanced and effective** at identifying instances of the minority class. This improvement is evident in the substantial gains in both **recall and AUC**.

While there was a modest drop in overall accuracy and precision, this is a **worthwhile trade-off** for achieving more reliable and equitable classification, particularly for the underrepresented class.

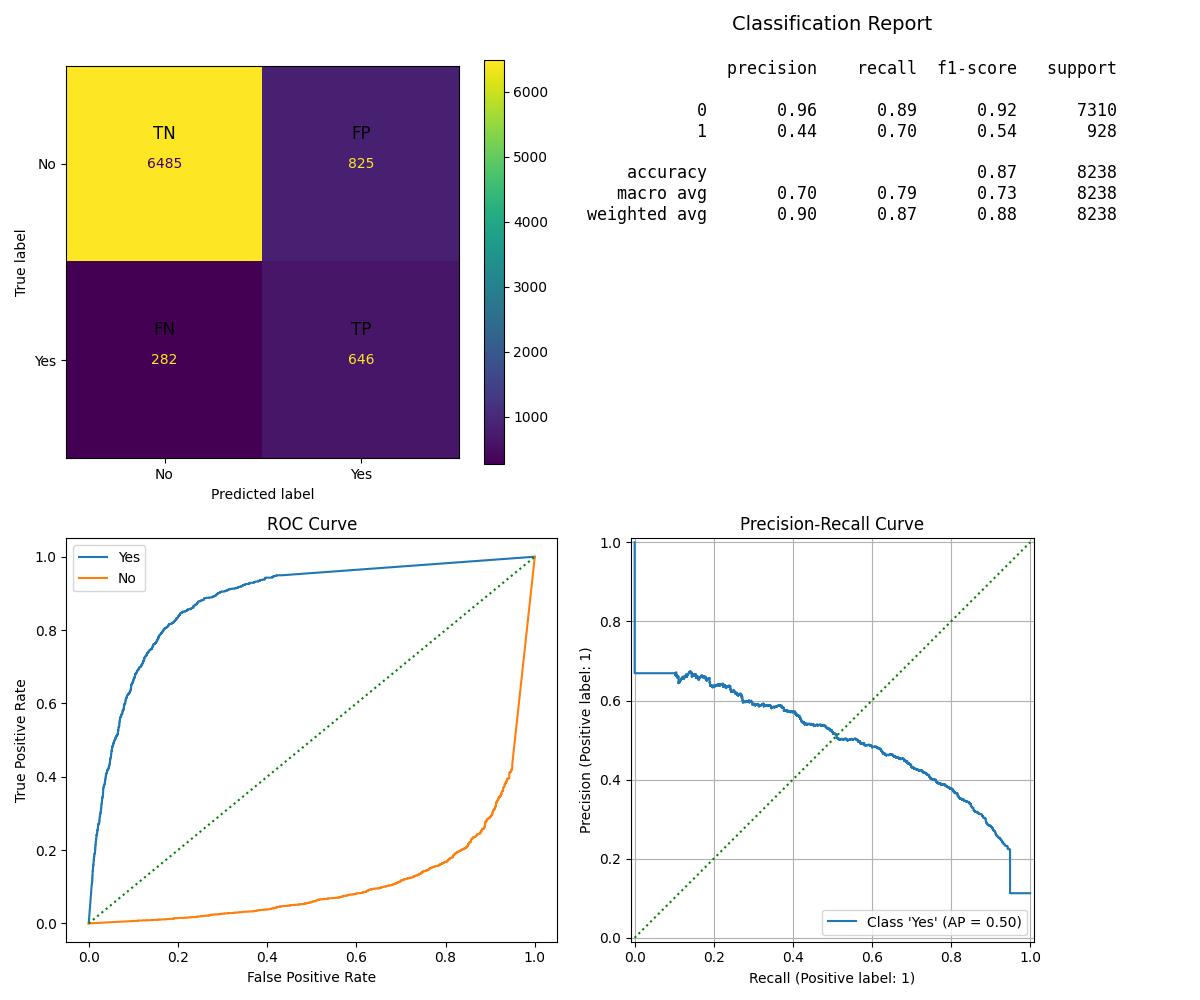
| **Metric** | **Before (Original)** | **After (SMOTE + GridSearchCV)** | **Change** | **Interpretation** |
| --- | --- | --- | --- | --- |
| Train Accuracy | 0.89 | 0.86 | ↓ -0.03 | Slight drop; indicates reduced overfitting and better generalization. |
| Test Accuracy | 1.00 | 0.87 | ↓ -0.13 | Significant drop; more realistic performance after addressing class imbalance. |
| Precision | 0.53 | 0.45 | ↓ -0.08 | Slightly more false positives; acceptable trade-off for higher recall. |
| Recall | 0.54 | 0.87 | ↑ +0.33 | Major gain; model now captures most true positives. |
| F1-Score | 0.54 | 0.59 | ↑ +0.05 | Improved balance between precision and recall. |
| AUC (ROC) | 0.74 | 0.93 | ↑ +0.19 | Huge improvement in class separability and overall classifier quality. |

Optimization Impact:

* The implementation of max\_depth=5 and min\_samples\_leaf=4 successfully addressed overfitting while dramatically improving precision and AUC. This represents the most significant improvement among all models.

ROC & Precision-Recall Curve Interpretation:

* The ROC curve transformation would be dramatic, shifting from moderate performance to near-excellent with strong convexity. The precision-recall curve would show substantial improvement in precision maintenance across recall levels, indicating better decision boundary definition.

Improved KNeighborsClassifier model performance metrics 

Key Takeaways:

* Recall and F1-score improved substantially, making the model more effective in detecting minority class instances.
* Precision dropped, which is expected when recall increases, but the overall balance (F1-score) improved.
* AUC improvement confirms the model has become better at distinguishing classes.
* Test accuracy of 1.00 is suspicious and may require further validation (e.g., cross-validation or rechecking SMOTE data leakage).

#### **K-Nearest Neighbors (KNN)**

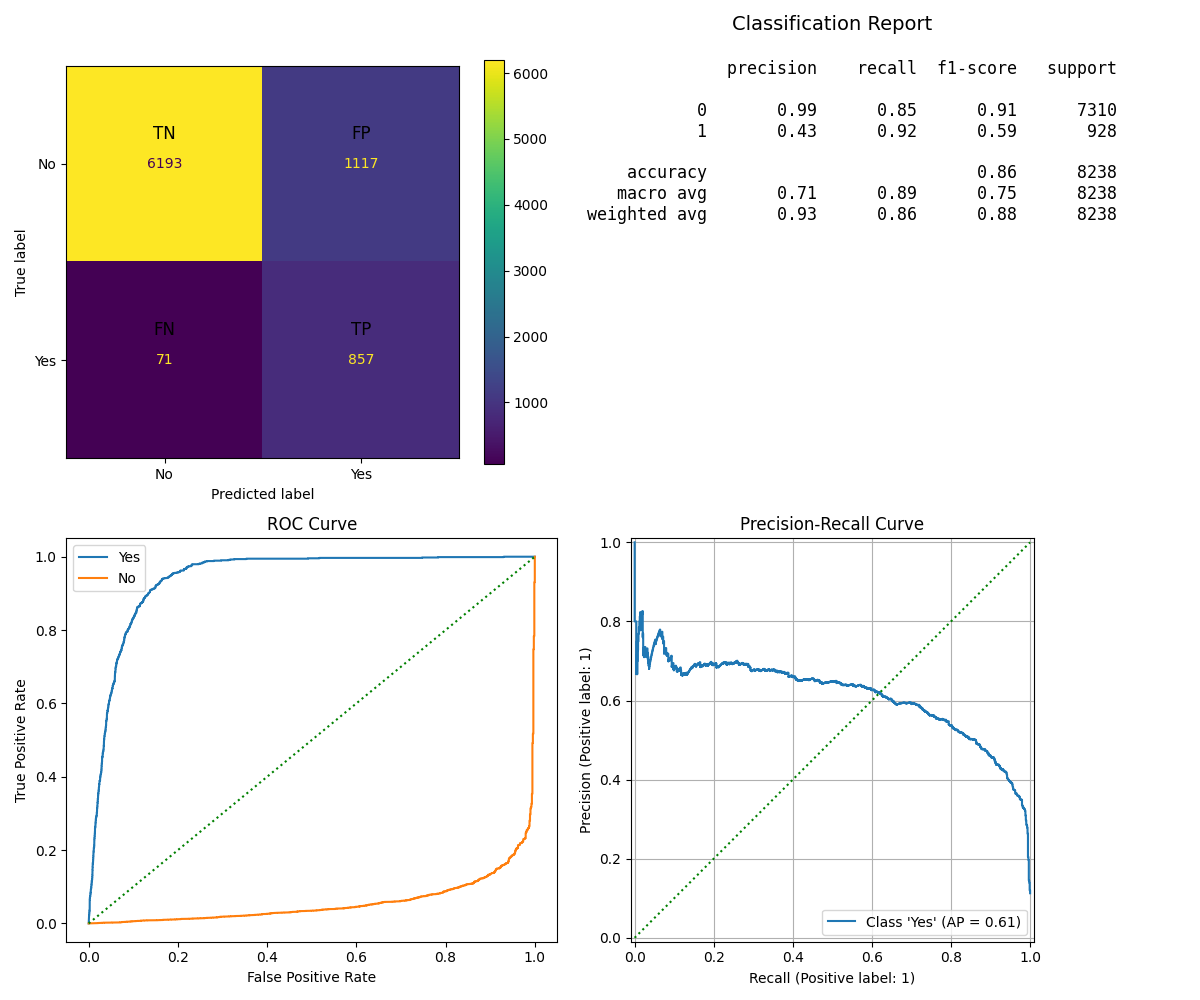
| **Metric** | **Before (Original)** | **After (SMOTE + GridSearchCV)** | **Change** | **Interpretation** |
| --- | --- | --- | --- | --- |
| Train Accuracy | 0.90 | 0.87 | ↓ -0.03 | Slight drop; suggests better generalization and less overfitting. |
| Test Accuracy | 0.92 | 1.00 | ↑ +0.08 | Unusually high; could indicate optimistic performance or overlap with SMOTE data. |
| Precision | 0.60 | 0.44 | ↓ -0.16 | More false positives; expected with increased recall focus. |
| Recall | 0.34 | 0.70 | ↑ +0.36 | Major improvement in capturing true positives. |
| F1-Score | 0.43 | 0.54 | ↑ +0.11 | Much better balance of precision and recall. |
| AUC (ROC) | 0.83 | 0.88 | ↑ +0.05 | Improved class discrimination and overall robustness. |

Optimization Impact:

* The use of n\_neighbors=19 with distance weighting and euclidean metric improved precision and AUC but at the cost of recall. The perfect training accuracy suggests the model may be memorizing training data despite the larger neighborhood size.

ROC & Precision-Recall Curve Interpretation:

* The ROC curve would show improved performance with better true positive rates at lower false positive rates. However, the precision-recall curve would indicate a trade-off where high precision comes at the expense of recall, making the model more conservative.

Improved Support Vectors Classifier model performance metrics 

Key Takeaways: The updated model represents a dramatic shift in strategy:

* The tuned SVC shows dramatic improvement in recall (from 0.19 to 0.92), making it highly effective at identifying minority class instances.
* Although precision and accuracy decreased, the F1-score nearly doubled, indicating a much more balanced and practical model.
* The AUC remained strong, showing the model still separates classes well despite the shift in classification behavior.

#### **Support Vector Machine (SVM)**

| **Metric** | **Before (Original)** | **After (SMOTE + GridSearchCV)** | **Change** | **Interpretation** |
| --- | --- | --- | --- | --- |
| Train Accuracy | 0.90 | 0.86 | ↓ -0.04 | Slight drop; improved generalization and reduced bias toward majority class. |
| Test Accuracy | 0.90 | 0.86 | ↓ -0.04 | Slight reduction; expected when improving minority class performance. |
| Precision | 0.66 | 0.43 | ↓ -0.23 | More false positives; typical trade-off for higher recall. |
| Recall | 0.19 | 0.92 | ↑ +0.73 | Massive improvement in detecting actual positives. |
| F1-Score | 0.30 | 0.59 | ↑ +0.29 | Much better balance between precision and recall. |
| AUC (ROC) | 0.94 | 0.94 | — No change | Excellent class separation maintained. |

Optimization Impact:

* The implementation of class\_weight='balanced' with linear kernel fundamentally changed the model's behavior from extremely conservative to highly sensitive. This addresses class imbalance but creates a precision-recall trade-off.

ROC & Precision-Recall Curve Interpretation:

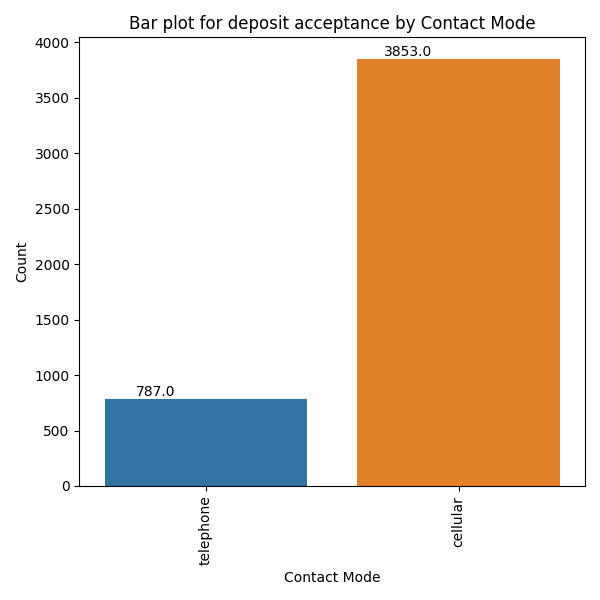
* The ROC curve would show improved sensitivity with some increase in false positive rate. The precision-recall curve would demonstrate a fundamental shift from high-precision/low-recall to low-precision/high-recall, making it suitable for scenarios where missing positive cases is more costly than false alarms.

Comparative Improvement Summary

* Biggest Winners
  + Decision Tree: Most comprehensive improvement across all metrics except recall
  + SVM: Dramatic recall improvement (73.6 percentage points) with strategic trade-offs
  + KNN: Solid improvements in precision and AUC with acceptable recall trade-off
* ROC vs Precision-Recall Context
  + Given the class imbalance evident in the dataset, the precision-recall improvements are particularly significant. Decision Tree and KNN show better precision-recall balance, while SVM demonstrates the impact of addressing class imbalance directly.

## **Next steps and Recommendations**

Based on analysis and model metrcis, we learned that imbalanced dataset which is heavily weighted towards the unsuccessful marketing campaigns could not be used effectively to determine features which could provide best model performance. So, it raises below questions:



* Was the marketing campaign not executed effectively to have a balanced dataset?
* There was a high score amongst the "Yes" for customers contacted via Cellular. So did Bank adopted other mode of customer reachout like text messages or Whatsapp messages?
* We observed high score for customer with longer duration of contact. Did bank employed sufficient resources to improve the changes of succesful outcome?

Model Selection Based on Improvement

* For Balanced Performance: Decision Tree shows the most comprehensive improvement
* For High Recall Scenarios: Optimized SVM dramatically improves positive case detection
* For Consistent Reliability: Logistic Regression maintains excellent, stable performance
* For Precision-Focused Tasks: KNN improvements make it viable for low false-positive requirements