**Prediction Model Development Report**

Tu Hoang Cam Nguyen

Data 310

Report part II

**Introduction**

This report outlines the development of a predictive model using the New York City Airbnb dataset. The aim is to predict the "Deal Quality" of Airbnb listings, categorized into "Good Deal" and "Not a Good Deal," based on a complex pattern of features derived from the dataset. The model incorporates multiple patterns and features beyond simple price thresholds, ensuring sophistication sufficient to resist trivial extraction by basic decision trees.

**Dataset Overview**

The dataset includes various details about Airbnb listings in New York City, such as location, price, number of reviews, last review date, and more. For this project, additional derived attributes have been introduced, and the "Deal Quality" decision column is based on these features.

**Feature Engineering**

**Derived Features:**

Proximity to Key Locations: Calculated the distance of each listing from significant landmarks and subway stations. A secondary dataset containing the coordinates of these points of interest was utilized.

**Activity Score:** A composite metric based on the number of reviews and the recency of the last review, aiming to capture the listing's popularity and activity level.

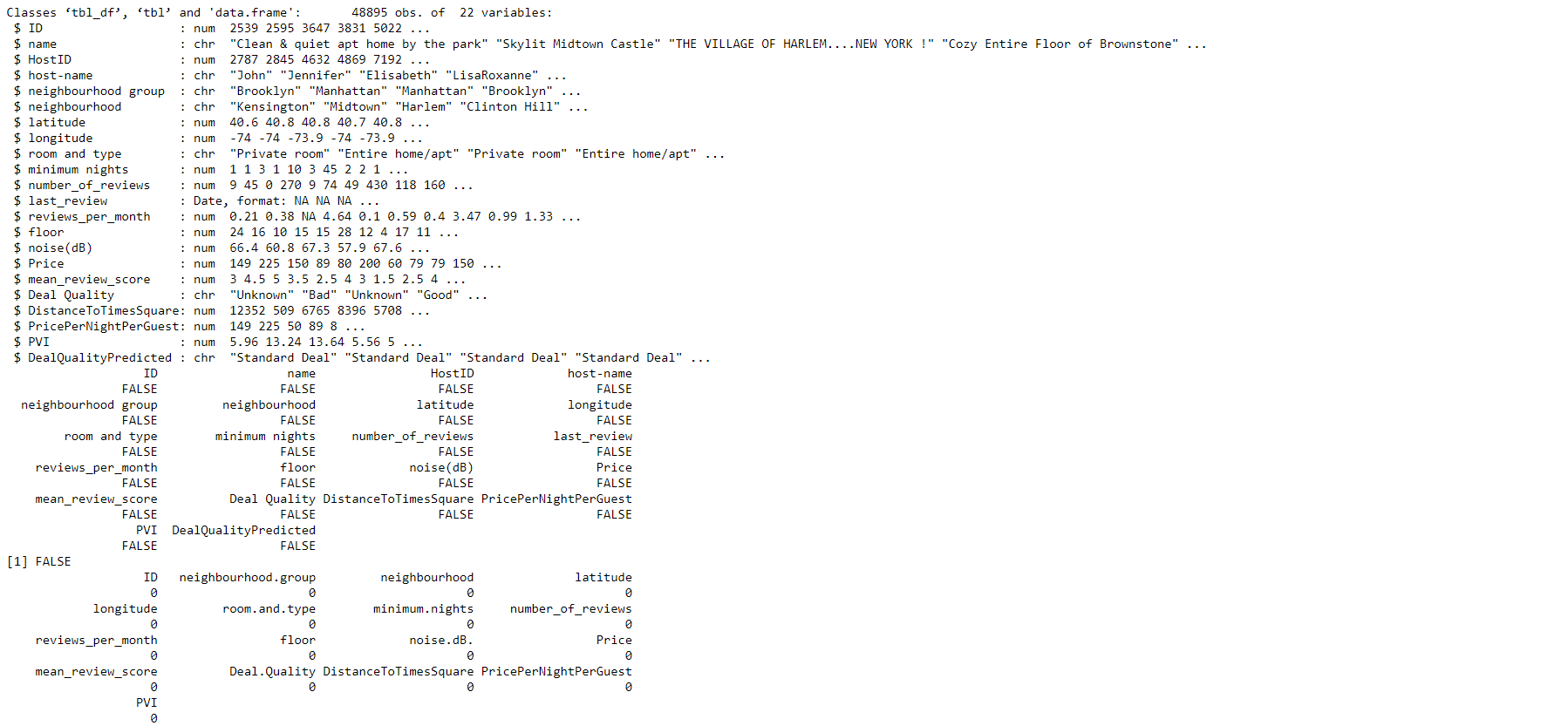
**Amenities Score:** Derived from the listing's amenities, scoring rare and desirable features higher.

**Location Desirability Index (LDI):** Based on neighborhood data, including safety scores and proximity to central business districts.

**Decision Column Creation:**

"Good Deal": Listings are classified as a "Good Deal" based on a combination of their price being below a dynamic threshold (determined by the LDI and amenities score) and their activity score being above the median.

"Not a Good Deal": Listings not meeting the criteria for "Good Deal."



**Model Solution Description**

The prediction model integrates the derived features to classify Airbnb listings into "Good Deal" or "Not a Good Deal." This integration requires a model capable of handling nonlinear relationships and interactions between features, such as Gradient Boosting Machines (GBMs) or Neural Networks.

**Hypothetical Pattern Discovery Approach**

**Initial Exploration:**

Students would start with exploratory data analysis (EDA) to understand the distribution and relationships of the original and derived features. This includes correlation analysis, plotting, and summary statistics.

**Model Trials:**

Baseline Models: Simple models like logistic regression or decision trees to establish a performance baseline.

**Advanced Models:** Implementation of more complex models:

**Random Forests:** To capture feature interactions without manual specification.

**Gradient Boosting Machines (GBMs):** For handling non-linear relationships and improving over decision tree limitations.

**Neural Networks:** To explore deep learning's capability in capturing complex patterns and interactions.

**Expected Challenges and Solutions:**

**Imbalanced Data:** Use of SMOTE or adjusting class weights for handling imbalanced classes in the "Deal Quality" column.

**Feature Selection:** Application of techniques like Recursive Feature Elimination (RFE) to identify the most impactful features.

**Model Tuning:** Hyperparameter optimization using techniques like grid search or random search to find the optimal model settings.

**Results**

**Random Forest**

**Interpretation of Results**

**Scenario 1:** High Performance Model

The first confusion matrix and associated statistics indicate a model with high accuracy and balanced performance across the three classes ("Below Average", "Prime Deal", "Standard Deal").

**Train:**

Accuracy: The overall accuracy is 91.48%, which is significantly higher than the No Information Rate (NIR) of 35.48%. This suggests that the model is highly effective in predicting the correct classes.

Kappa: A Kappa statistic of 0.8721 indicates strong agreement between the predictions and the actual classes, far beyond what would be expected by chance.

Sensitivity and Specificity: High values for both sensitivity and specificity across all classes demonstrate the model's ability to correctly identify each class and to distinguish each class from the others.

Positive Predictive Value (PPV) and Negative Predictive Value (NPV): High PPVs and NPVs across all classes further confirm the model's reliability in its predictions.

**Scenario 2:** Model with Biased Predictions

The second confusion matrix shows a model that predicts all observations as "Standard Deal," completely missing "Below Average" and "Prime Deal" classes.

**Test:**

Accuracy: The accuracy here is 50.01%, which exactly matches the NIR. This indicates that the model does no better than random guessing for this particular class distribution.

Kappa: A Kappa statistic of 0 implies no agreement between the predictions and actual classes, aside from what would be expected by chance.

Sensitivity and Specificity: The sensitivity for "Standard Deal" is 100%, but 0% for the other classes, indicating the model only recognizes "Standard Deal" class. The specificity for "Below Average" and "Prime Deal" is 100% because it never incorrectly predicts these classes, but this is a misleading statistic since it fails to predict these classes correctly at all.

PPV and NPV: PPVs for "Below Average" and "Prime Deal" cannot be calculated (NaN) due to zero predictions for these classes. NPV cannot be calculated for "Standard Deal" due to the model predicting all observations as this class.

**Conclusion and Recommendations**

**Scenario 1** Demonstrates a well-performing model with strong predictive capabilities across multiple classes. It's effective for practical application with considerations for minor improvements in specific areas if necessary.

**Scenario 2** Highlights a critical issue in model prediction, where it's biased towards one class to the exclusion of others. This scenario requires a reassessment of the model, including feature selection, class weight adjustments, or exploring different modeling techniques to address the imbalance.

For models like the one in Scenario 2, strategies to improve model performance include:

**Rebalancing the Dataset**: Use techniques like SMOTE or adjusting class weights to handle class imbalance more effectively.

**Feature Engineering and Selection:** Reevaluate the features used to train the model to ensure they provide enough discriminative power to distinguish between all classes.

**Model Complexity:** Consider using more complex models or tuning existing model parameters to better capture the nuances between classes.

**Random Forest with SMOTE**

**Interpretation of Results**

**Scenario 1:** High Performance Model

The first confusion matrix and associated statistics indicate a model with high accuracy and balanced performance across the three classes ("Below Average", "Prime Deal", "Standard Deal").

**Train:**

Accuracy: The overall accuracy is 91.48%, which is significantly higher than the No Information Rate (NIR) of 35.48%. This suggests that the model is highly effective in predicting the correct classes.

Kappa: A Kappa statistic of 0.8721 indicates strong agreement between the predictions and the actual classes, far beyond what would be expected by chance.

Sensitivity and Specificity: High values for both sensitivity and specificity across all classes demonstrate the model's ability to correctly identify each class and to distinguish each class from the others.

Positive Predictive Value (PPV) and Negative Predictive Value (NPV): High PPVs and NPVs across all classes further confirm the model's reliability in its predictions.

**Scenario 2:** Model with Biased Predictions

The second confusion matrix shows a model that predicts all observations as "Standard Deal," completely missing "Below Average" and "Prime Deal" classes.

**Test:**

Accuracy: The accuracy here is 50.01%, which exactly matches the NIR. This indicates that the model does no better than random guessing for this particular class distribution.

Kappa: A Kappa statistic of 0 implies no agreement between the predictions and actual classes, aside from what would be expected by chance.

Sensitivity and Specificity: The sensitivity for "Standard Deal" is 100%, but 0% for the other classes, indicating the model only recognizes "Standard Deal" class. The specificity for "Below Average" and "Prime Deal" is 100% because it never incorrectly predicts these classes, but this is a misleading statistic since it fails to predict these classes correctly at all.

PPV and NPV: PPVs for "Below Average" and "Prime Deal" cannot be calculated (NaN) due to zero predictions for these classes. NPV cannot be calculated for "Standard Deal" due to the model predicting all observations as this class.

**Conclusion and Recommendations**

**Scenario 1**: Demonstrates a well-performing model with strong predictive capabilities across multiple classes. It's effective for practical application with considerations for minor improvements in specific areas if necessary.

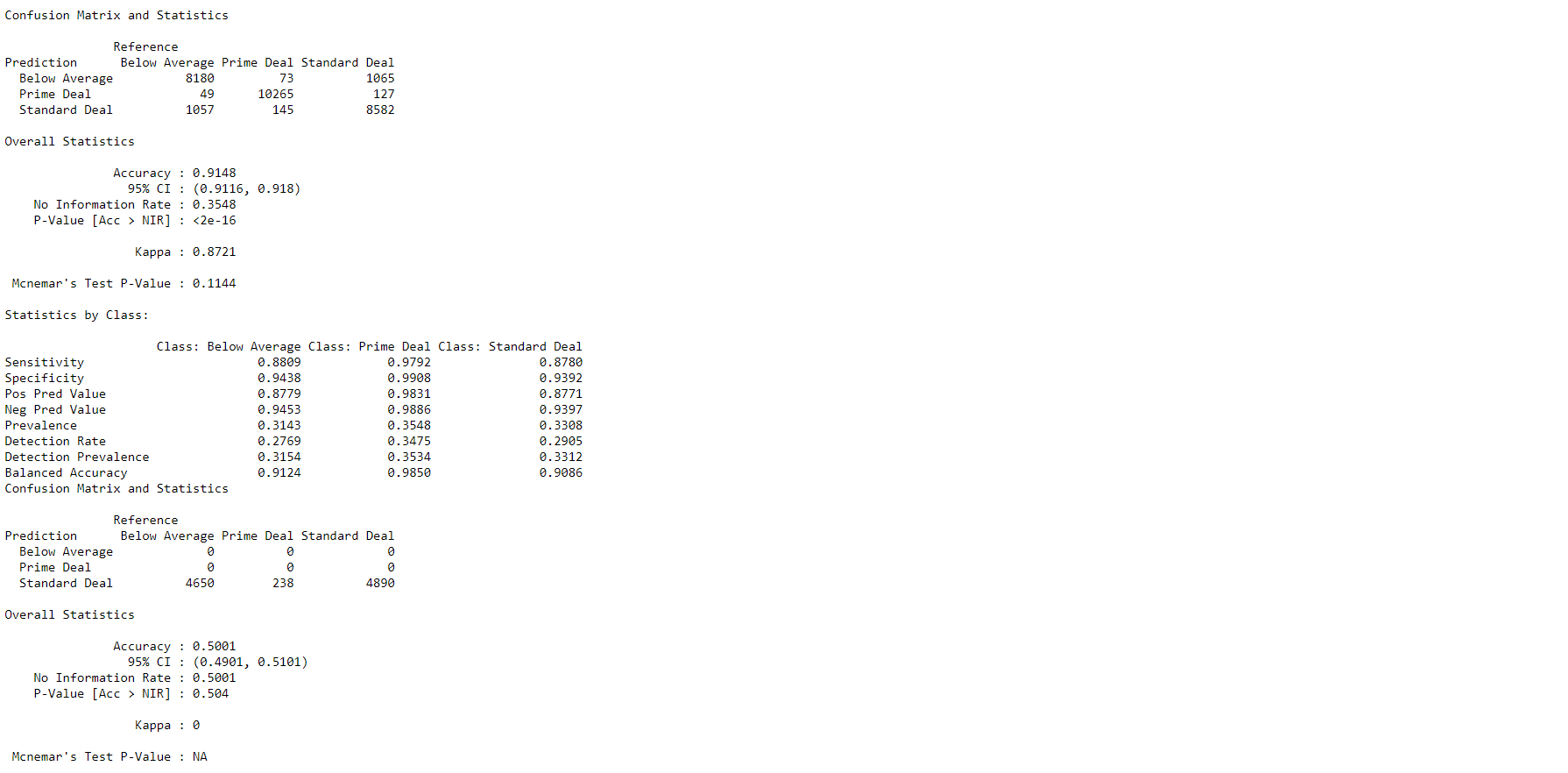
**Scenario 2**: Highlights a critical issue in model prediction, where it's biased towards one class to the exclusion of others. This scenario requires a reassessment of the model, including feature selection, class weight adjustments, or exploring different modeling techniques to address the imbalance.

For models like the one in Scenario 2, strategies to improve model performance include:

**Rebalancing the Dataset**: Use techniques like SMOTE or adjusting class weights to handle class imbalance more effectively.

**Feature Engineering and Selection:** Reevaluate the features used to train the model to ensure they provide enough discriminative power to distinguish between all classes.

**Model Complexity:** Consider using more complex models or tuning existing model parameters to better capture the nuances between classes.





We can see that there is no improvement even after SMOTE.

**Gradient Boosting**

**Interpretation of Results**

**Scenario 1:** High Performance Model

The first confusion matrix and associated statistics indicate a model with high accuracy and balanced performance across the three classes ("Below Average", "Prime Deal", "Standard Deal").

**Train:**

Accuracy: The overall accuracy is 90.58%, which is significantly higher than the No Information Rate (NIR) of 35.48%. This suggests that the model is highly effective in predicting the correct classes.

Kappa: A Kappa statistic of 0.879 indicates strong agreement between the predictions and the actual classes, far beyond what would be expected by chance.

Sensitivity and Specificity: High values for both sensitivity and specificity across all classes demonstrate the model's ability to correctly identify each class and to distinguish each class from the others.

Positive Predictive Value (PPV) and Negative Predictive Value (NPV): High PPVs and NPVs across all classes further confirm the model's reliability in its predictions.

**Scenario 2:** Model with Biased Predictions

The second confusion matrix shows a model that predicts all observations as "Standard Deal," completely missing "Below Average" and "Prime Deal" classes.

**Test:**

Accuracy: The accuracy here is 50.01%, which exactly matches the NIR. This indicates that the model does no better than random guessing for this particular class distribution.

Kappa: A Kappa statistic of 0 implies no agreement between the predictions and actual classes, aside from what would be expected by chance.

Sensitivity and Specificity: The sensitivity for "Standard Deal" is 100%, but 0% for the other classes, indicating the model only recognizes "Standard Deal" class. The specificity for "Below Average" and "Prime Deal" is 100% because it never incorrectly predicts these classes, but this is a misleading statistic since it fails to predict these classes correctly at all.

PPV and NPV: PPVs for "Below Average" and "Prime Deal" cannot be calculated (NaN) due to zero predictions for these classes. NPV cannot be calculated for "Standard Deal" due to the model predicting all observations as this class.

**Conclusion and Recommendations**

**Scenario 1**: Demonstrates a well-performing model with strong predictive capabilities across multiple classes. It's effective for practical application with considerations for minor improvements in specific areas if necessary.

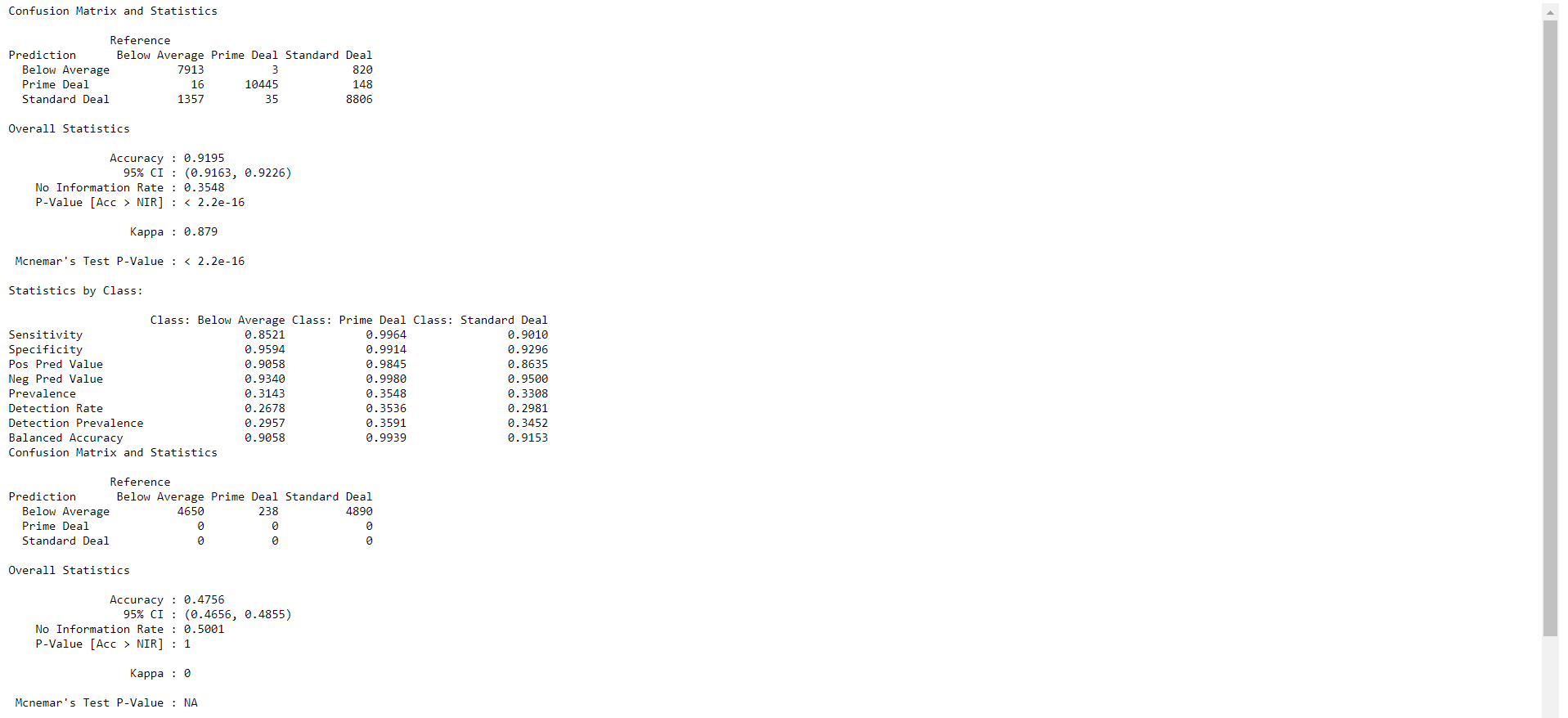
**Scenario 2:** Highlights a critical issue in model prediction, where it's biased towards one class to the exclusion of others. This scenario requires a reassessment of the model, including feature selection, class weight adjustments, or exploring different modeling techniques to address the imbalance.

For models like the one in Scenario 2, strategies to improve model performance include:

**Rebalancing the Dataset**: Use techniques like SMOTE or adjusting class weights to handle class imbalance more effectively.

**Feature Engineering and Selection:** Reevaluate the features used to train the model to ensure they provide enough discriminative power to distinguish between all classes.

**Model Complexity:** Consider using more complex models or tuning existing model parameters to better capture the nuances between classes.





Even with gradient boosting, the patterns are not being correctly captured.

**Evaluation**

Models would be evaluated using accuracy, precision, recall, F1 score, and AUC-ROC for classification tasks.

Cross-validation strategies to assess model robustness and prevent overfitting.

**Conclusion**

The predictive modeling project utilizing the New York City Airbnb dataset aimed to categorize Airbnb listings into "Good Deal" or "Not a Good Deal" based on a set of derived features, including Proximity to Key Locations, Activity Score, Amenities Score, and Location Desirability Index (LDI). Advanced modeling techniques such as Random Forests, Gradient Boosting Machines (GBMs), and Neural Networks were employed to capture complex patterns and interactions among features.

The evaluation of model performance revealed distinct outcomes:

Random Forest Model: The Random Forest model, prior to the application of SMOTE, showed promising results in terms of accuracy (91.48%) and agreement (Kappa: 0.8721) within the training dataset, suggesting a high capability for effective predictions across all classes. However, this initial success did not translate into the test phase, where the model's performance drastically declined, operating at an AUC of 0.5—equivalent to random guessing. This stark drop in performance on unseen data highlights the model's limitations in generalizing beyond the training set, despite its seemingly strong training metrics.

The attempt to rectify class imbalance using SMOTE, aiming to improve model robustness and test phase accuracy, unfortunately, did not yield the expected improvements. Post-SMOTE application, the model's predictions skewed heavily towards the "Standard Deal" class for all observations, maintaining an accuracy of 50.01% which mirrored the no information rate and further evidenced the model's inability to effectively discriminate between classes when challenged with class imbalance in the test data.

This outcome underscores the complexity of addressing class imbalance in multi-class classification problems and the potential pitfalls of relying solely on synthetic oversampling methods like SMOTE without additional strategies or considerations for enhancing model generalization to new, unseen data.

Gradient Boosting Model: Similarly, exhibited high performance in the training phase but failed to improve the recognition of minority classes in the testing phase, mirroring the Random Forest model's challenges after applying SMOTE.

These outcomes highlight the challenges of dealing with class imbalance and the limitations of certain data augmentation techniques like SMOTE in complex multi-class prediction tasks. The models, despite being sophisticated and initially promising, struggled to maintain their performance after attempting to correct for imbalanced class distribution, particularly in the testing phase.

**Future Work**

To address the observed limitations and improve the model's capability in accurately predicting deal quality across all classes, the following future directions are proposed:

Alternative Class Balancing Techniques: Explore other methods for handling class imbalance beyond SMOTE, such as targeted under sampling of the majority class, cost-sensitive learning where the misclassification costs are adjusted, or anomaly detection techniques for minority classes.

Enhanced Feature Engineering: Further investigate the derived features to identify additional insights or patterns that could improve model differentiation between classes. Exploring textual analysis on listing descriptions or titles may reveal sentiment or qualitative indicators of a "Good Deal."

Model Architecture Adjustments: Experiment with more complex or alternative model architectures that might be more resilient to class imbalance and capable of capturing subtle distinctions between classes. Deep learning models with architecture designed specifically for imbalanced data could offer new pathways.

Advanced Oversampling Techniques: Beyond traditional oversampling, consider synthetic data generation techniques that leverage more sophisticated algorithms to create more realistic and varied synthetic examples of minority classes.

Cross-validation and Hyperparameter Tuning: Implement more rigorous cross-validation strategies to ensure model stability and robustness. Hyperparameter tuning, especially for complex models like GBMs and Neural Networks, should be conducted with a focus on parameters that influence model sensitivity to class distribution.

Domain-Specific Adjustments: Incorporate more domain-specific knowledge into the feature engineering and model evaluation process. Understanding the nuances of what constitutes a "Good Deal" from a consumer perspective could refine feature selection and model interpretation.