CSI 5155 Machine Learning Project Report

All code was done using Python in an interactive notebook file. Results are available as inline output of each cell. ROC Curve images for each model are attached.

Project Group 7

John Joseph Zagariah Daniel (300382174)
Rowan Mohamed Amin Hefny Hussein (300381055)

Semi-supervised learning, label scarcity, and class imbalance

Introduction:

This report investigates the effectiveness of semi-supervised learning techniques in addressing label scarcity and class imbalance. We explore various inductive semi-supervised algorithms, including self-training, co-training, semi-supervised ensembles, and an approach leveraging unsupervised pre-training. Our experiments evaluate the effectiveness of these techniques on a dataset with class imbalance. We assess model performance using a comprehensive set of metrics, including accuracy, F1-score, precision, recall, and ROC curve plots. This report delves into the detailed procedures, results, and the overall impact of semi-supervised learning in mitigating label scarcity and class imbalance.

1. Initial supervised learning

Procedure:

- Conducted feature engineering on the dataset.
- Established a baseline model using a Support Vector Machine (SVM) with optimal hyperparameters (C=1, gamma=0.1, kernel='rbf', etc.).
- Trained the SVM on the original imbalanced dataset.
- Applied two resampling techniques:
  - NearMiss: Removed majority class samples to balance the class distribution.
  - SMOTE: Generated synthetic samples for the minority class.
- Retrained the SVM model on the resampled datasets.
- Evaluated the performance of all models using accuracy, precision, recall, and F1-score.

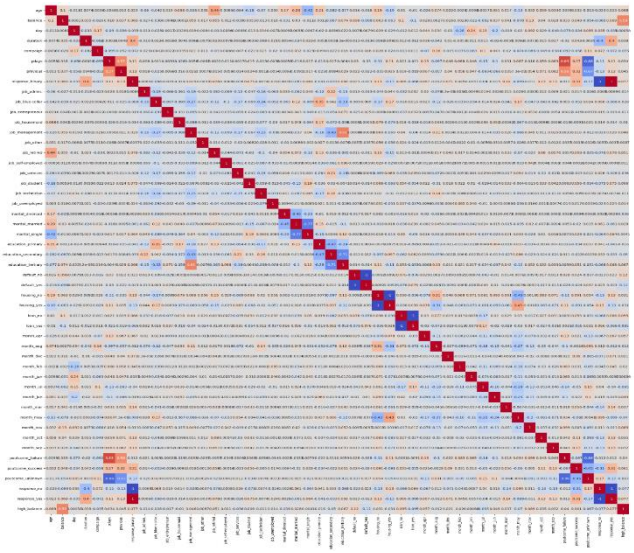
Model	Accuracy	Precision	Recall	F1-score
Baseline (Imbalanced)	0.8999	0.6885	0.2758	0.3938
NearMiss Under sampling	0.7084	0.7082	0.745	0.7146
SMOTE Oversampling	0.9344	0.9769	0.8903	0.9325

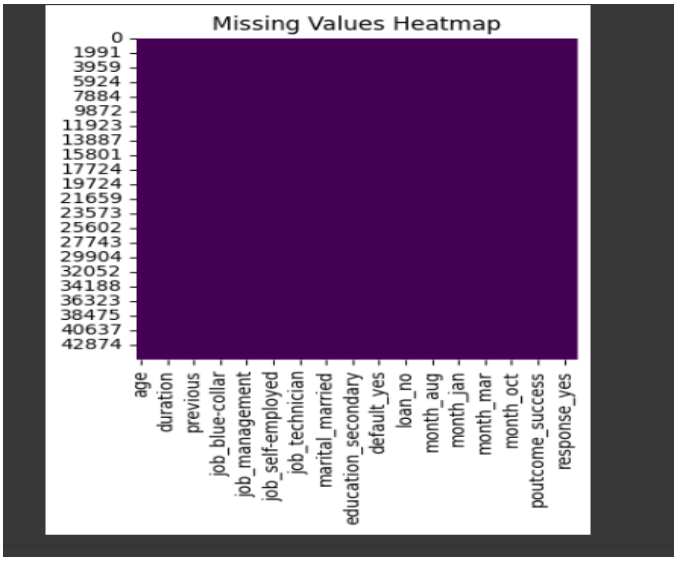
Results:

The baseline model achieved high overall accuracy (89.99%) but exhibited a low recall (27.58%) for the minority class. NearMiss improved recall for the minority class (74.50%) but lowered overall accuracy (70.84%). SMOTE emerged as the most effective technique, achieving both high accuracy (93.44%) and improved recall for the minority class (89.03%). This demonstrates the potential of resampling techniques to address class imbalance and improve model performance for the minority class.

1. Initial supervised learning Version 2

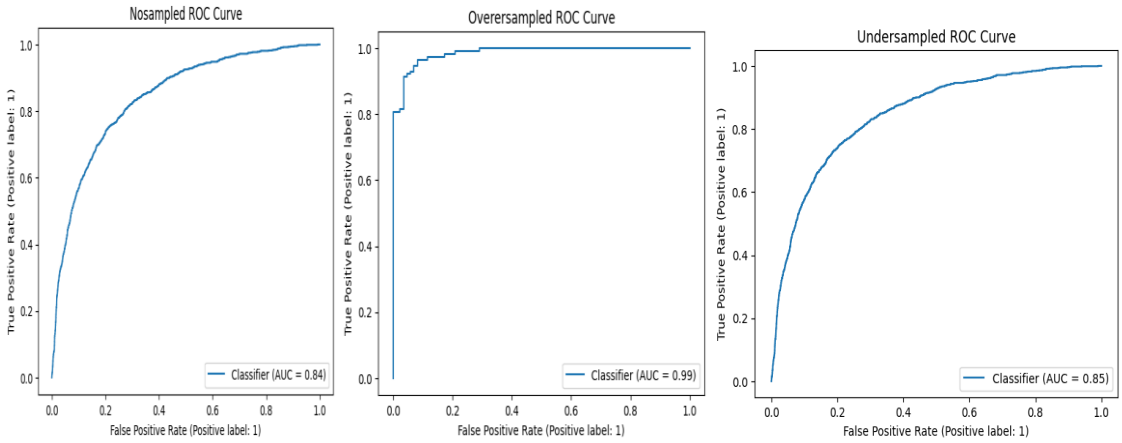
In this project, we delve into the challenges of semi-supervised learning, label scarcity, and class imbalance using a dataset from a Portuguese bank's direct marketing campaign. The initial step involved preprocessing the data to ensure a robust foundation for model training. This included stratifying the data into training and test sets with an 80-20 split and employing one-hot encoding to convert categorical variables into a machine-readable format. Numerical features were standardized using a StandardScaler to normalize the data distribution. Additionally, we introduced a novel binary feature, 'high_balance', to distinguish clients based on the median balance, which could provide further insight into the customer's likelihood to subscribe to a term deposit also with a heatmap to assert the absence of missing values, confirming the dataset's readiness for the subsequent stages. Furthermore, we constructed a detailed correlation matrix heatmap to uncover the intricate relationships between features. This visualization revealed the degrees of correlation, highlighting potential predictors and areas where multicollinearity might be a concern





Results:

- **Without Sampling**: The model's performance without any sampling technique yielded an using 'LogisticRegressionCV 'the Accuracy: 0.8849, F1 Score: 0.2645, ROC AUC: 0.8423, suggesting a decent ability to distinguish between the classes but with room for improvement.
- **Oversampling**: Applying Synthetic Minority Over-sampling Technique (SMOTE) to balance the dataset improved the classifier's Roc AUC dramatically to 0.99, Accuracy: 0.9400, F1 Score: 0.9483, indicating a nearly perfect classification by the model.
- **Undersampling**: The use of RandomUnderSampler resulted in an Roc AUC of 0.85, Accuracy: 0.8057, F1 Score: 0.4761,which is slightly better than the model without sampling but not as high as with oversampling.
- The findings show that resampling techniques, especially oversampling with SMOTE, significantly improve the model's ability to predict the minority class in an imbalanced dataset. This underlines the importance of addressing class imbalance in predictive modeling to enhance performance metrics and achieve more reliable and fair predictions.



2. Semi-Supervised Learning

2.1 Self Training Algorithm

Self-training is a semi-supervised learning technique that iteratively improves a model by leveraging both labeled and unlabeled data. It starts by training a model on a small set of labeled data. Then, the model predicts labels (pseudo-labels) for the unlabeled data, with only the most confident predictions added to the labeled set. The model is then retrained on this expanded dataset, allowing it to potentially learn from the unlabeled data and enhance its performance over multiple iterations. This approach is particularly useful when labeled data is limited but a wealth of unlabeled data exists.

Procedure:

- Constructed a SelfTrainingClassifier from scikit-learn Base estimator: SVM.
- For each unlabeled data percentage (0%, 50%, 75%, 90%, 95%, 99%):
  - Split the dataset into labeled and unlabeled portions as specified by the percentage.
  - Train the SelfTrainingClassifier on the labeled data.

- Evaluate model performance on the held-out test set.

➢ Collect accuracy, precision, recall, and F1-score for each unlabeled data percentage scenario.
➢ Performed the same steps using both under sampled and over sampled dataset.

Dataset	Unlabeled Data	Accuracy	Precision	Recall	F1-Score	ROC Curve Area
Imbalanced	0.00%	0.8999	0.6281	0.3544	0.4531	0.66
	50.00%	0.8977	0.5887	0.4168	0.4881	0.69
	75.00%	0.8968	0.5912	0.3827	0.4647	0.67
	90.00%	0.896	0.5821	0.395	0.4707	0.68
	95.00%	0.8949	0.5818	0.3629	0.447	0.66
	99.00%	0.8809	0.4816	0.2362	0.317	0.6
NearMiss	0.00%	0.7173	0.7031	0.7523	0.7269	0.72
	50.00%	0.6947	0.6886	0.7107	0.6995	0.69
	75.00%	0.6862	0.6927	0.6691	0.6807	0.69
	90.00%	0.6677	0.6632	0.6814	0.6722	0.67
	95.00%	0.6517	0.6547	0.6417	0.6482	0.65
	99.00%	0.5047	0.5045	0.5293	0.5166	0.5
Smote	0.00%	0.9373	0.9627	0.9098	0.9355	0.94
	50.00%	0.9363	0.9623	0.9082	0.9344	0.94
	75.00%	0.9363	0.9551	0.9157	0.935	0.94
	90.00%	0.936	0.9559	0.914	0.9345	0.94
	95.00%	0.9302	0.9394	0.9197	0.9295	0.93
	99.00%	0.9048	0.9196	0.8872	0.9031	0.9

Results:
In the imbalanced dataset, models maintained high accuracy but struggled with precision, recall, and F1-score, especially with increasing percentages of unlabeled data. NearMiss sampling improved precision, recall, and F1-score compared to the imbalanced dataset, but performance slightly decreased as the percentage of unlabeled data increased. Conversely, SMOTE sampling significantly enhanced precision, recall, and F1-score across all scenarios, emerging as the most effective technique for addressing class imbalance. As the percentage of unlabeled data increased, the Self-Training algorithm demonstrated adaptability by leveraging a larger pool of unlabeled examples to refine its decision boundaries and capture underlying patterns. However, excessively high percentages of unlabeled data posed challenges, leading to diminished performance metrics. Balancing the distribution of labeled and unlabeled data is crucial, and optimization strategies are necessary to effectively leverage unlabeled data for improved model performance and accurate predictions in real-world applications.

2.2 A Co-Training Algorithm

we've leveraged the co-training approach using CTClassifier to predict client subscription outcomes in a Portuguese bank's marketing campaign. The CTClassifier was initially applied with two different estimators, RandomForest and a LogisticRegression, to handle multiple views of the dataset, aiming to capitalize on their distinct learning strengths.

In a comparative approach, the CTClassifier was later trained using two LogisticRegression models for both views. The sampling strategies tested were no sampling, SMOTE, and

RandomUnderSampler, to tackle the prevalent class imbalance. The evaluation focused on ROC-AUC, accuracy, and F1 scores to assess performance and a t-test for statistical significance regarding model training runtime.

The results revealed an interesting outcome: while the AUC scores improved with both oversampling and undersampling, the best F1 score was achieved with SMOTE, highlighting its effectiveness in improving the classifier's recall without severely compromising precision. Furthermore, the runtime results were statistically significant, indicating that the efficiency gains with SMOTE and RandomUnderSampler were not due to random chance.

we learned several valuable lessons:

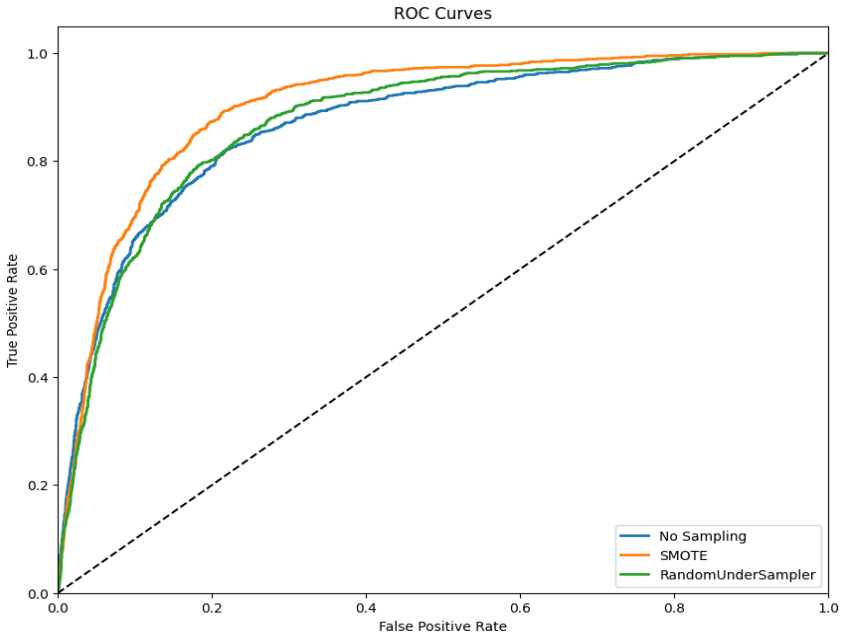
- **Class Imbalance Impact**: The performance improvements with SMOTE and RandomUnderSampler underline the critical impact of class imbalance on model performance and the importance of using appropriate sampling techniques.
- **Estimator Choice**: The differences in performance between using RandomForest and LogisticRegression estimators within CTClassifier emphasize the importance of choosing the right estimator for the task at hand.
- **Efficiency and Effectiveness**: The runtime analysis showed that resampling methods could be efficient without compromising performance, as evidenced by the comparable runtime between SMOTE and RandomUnderSampler and their effectiveness in improving the ROC-AUC and F1 scores.
- **Estimator Consistency**: Using the same estimator for both views didn't substantially degrade performance, suggesting that in scenarios where computational resources or model interpretability is a concern, a more uniform approach could still yield beneficial results.
- **Balance Between Metrics**: A high ROC-AUC does not necessarily mean a high F1 score, highlighting the importance of considering multiple metrics to get a holistic view of model performance.

Comparing the results obtained using different estimators within the CTClassifier illustrates that while diverse models can capitalize on distinct patterns in the data, consistency in estimator choice can also be effective, especially when dealing with a class imbalance. This insight stresses the value of experimenting with different models and resampling techniques to find an optimal balance between predictive performance and computational efficiency.

Results:

Co with under and over sampling two different estimator

Training with No Sampling ROC-AUC: 0.869, Accuracy: 0.886, F1-Score: 0.212, Runtime: 1.986sTraining with SMOTE: ROC-AUC: 0.903, Accuracy: 0.894, F1-Score: 0.521, Runtime: 7.055s Training with RandomUnderSampler: ROC-AUC: 0.874, Accuracy: 0.8106, F1-Score: 0.5077, Runtime: 1.41 S

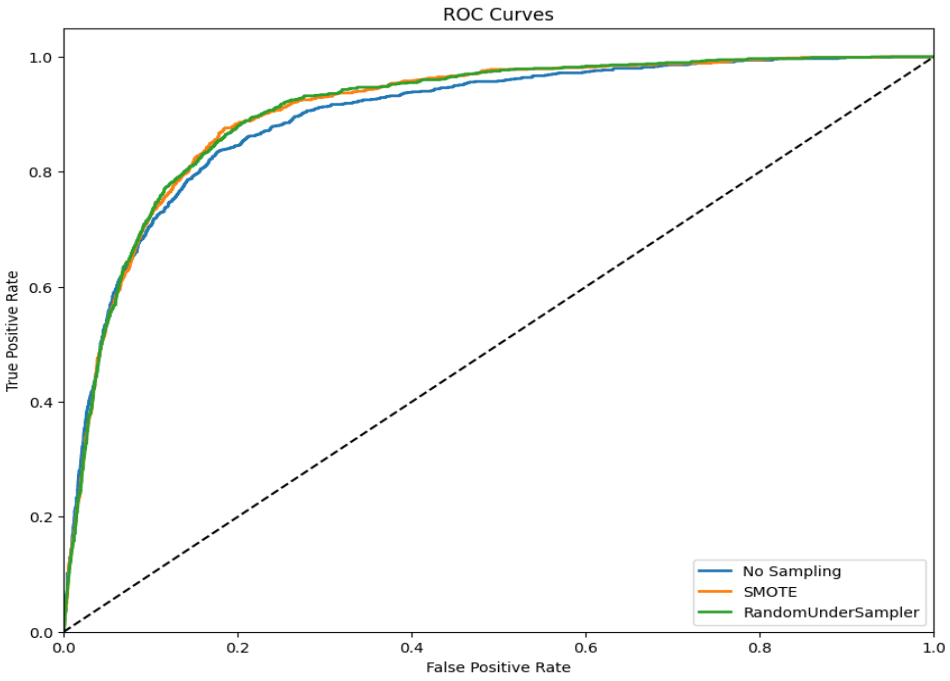


Same estimators

Training with No Sampling: ROC-AUC: 0.896, Accuracy: 0.885, F1-Score: 0.186, Runtime: 1.764s

Training with SMOTE ROC-AUC: 0.906, Accuracy: 0.849, F1-Score: 0.5683, Runtime: 1.255s

Training with RandomUnderSampler ROC-AUC: 0.907, Accuracy: 0.844, F1-Score: 0.562, Runtime: 1.1329



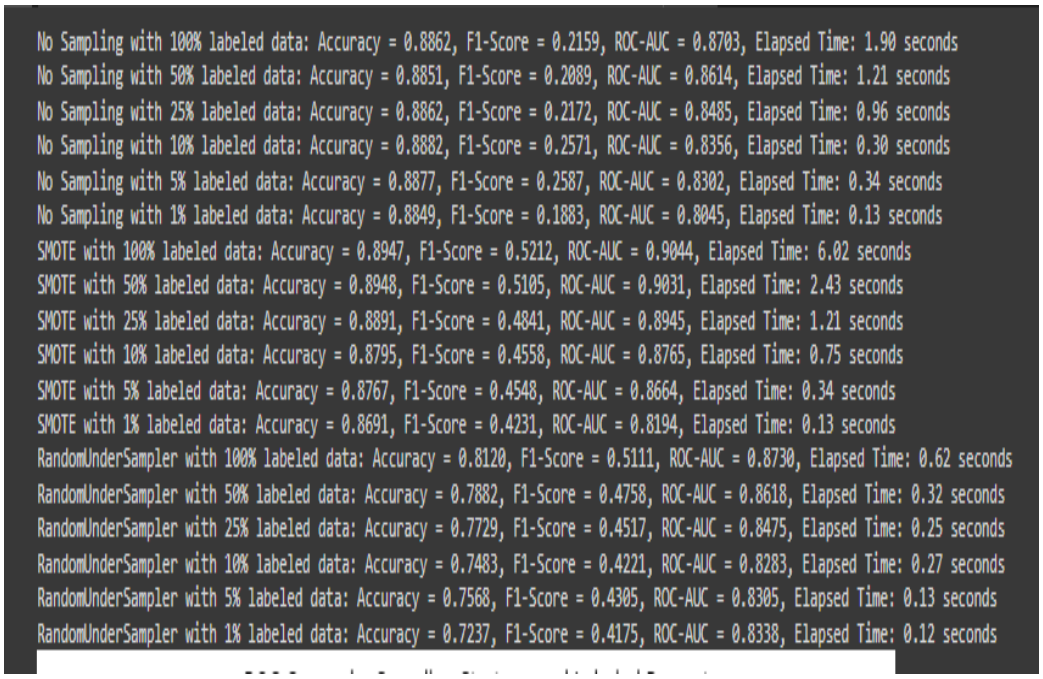
- **None Sampler** strategy proves to be surprisingly resilient, with accuracy and F1-scores remaining relatively stable even when labeled data is reduced to 50%. This suggests that semi-supervised learning techniques can effectively leverage large amounts of unlabeled data. However, as we venture into the higher brackets of unlabeled data, the ROC-AUC metric

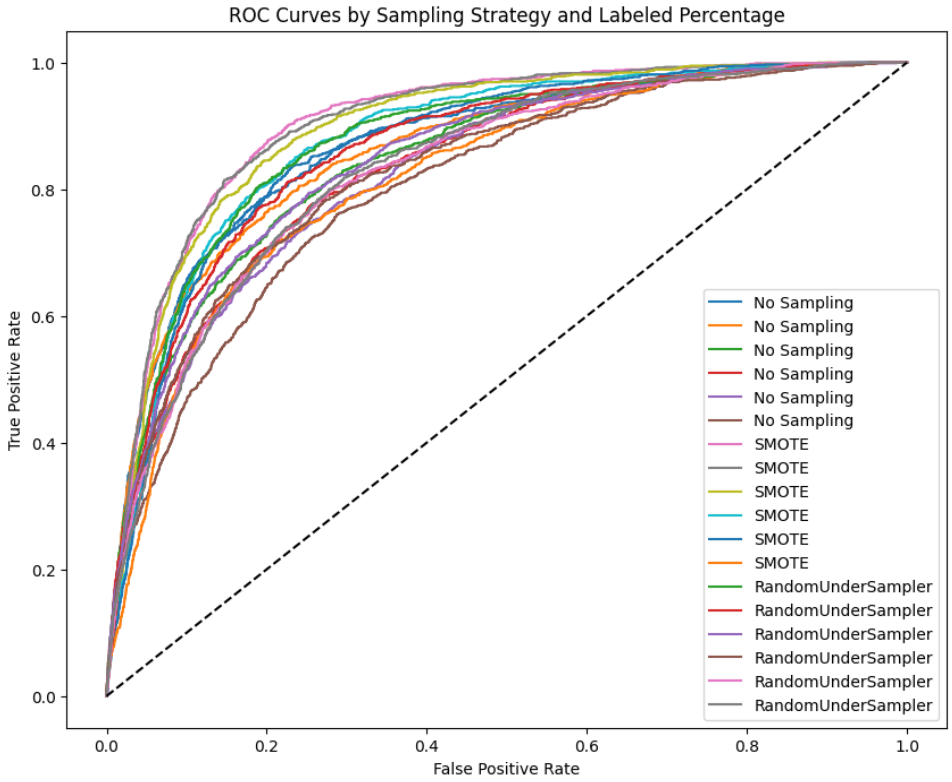
starts to decline, indicating that while the models can still make effective use of the unlabeled data, the absence of labels starts to impede the classifier's performance, especially at the extreme of 1% labeled data.

- **Oversample** by SMOTE, when applied, enhances the model's ability to predict the minority class across all levels of labeled data. Notably, even with only 1% labeled data, SMOTE ensures that the model remains functional, though with reduced effectiveness as indicated by lower accuracy and F1 scores compared to a fully labeled scenario. This reduction is less drastic than what we observe with no sampling, showcasing SMOTE's capability in mitigating the impact of label scarcity.
- **UnderSampler** presents the most significant trade-offs. It improves the recall at the cost of overall accuracy and precision, with this effect becoming more pronounced as the labeled data decreases. With only 1% labeled data, the models exhibit a remarkable ability to maintain high ROC-AUC scores, but this does not translate into high accuracy or F1 scores, implying that the model is better at identifying true positives at the risk of increasing false positives.

These findings illuminate the delicate balance between the quantity of labeled data and the choice of resampling strategies in semi-supervised learning, particularly in skewed datasets. They reinforce the notion that a well-chosen semi-supervised technique can substantially reduce the need for labeled data without severely compromising model performance, which is pivotal in applications where the acquisition of labeled data is costly or time-consuming.

.





2.3 Semi-supervised ensemble

Aimed at implementing a semi-supervised learning approach using SelfTrainingClassifier with ensemble methods like RandomForestClassifier, LogisticRegression, and GradientBoostingClassifier. An ensemble of classifiers is created using **VotingClassifier**, with each base classifier wrapped inside a **SelfTrainingClassifier**.**SelfTrainingClassifier** uses a criterion ('k_best' with a parameter k_best) to select confident pseudo-labeled samples for training.

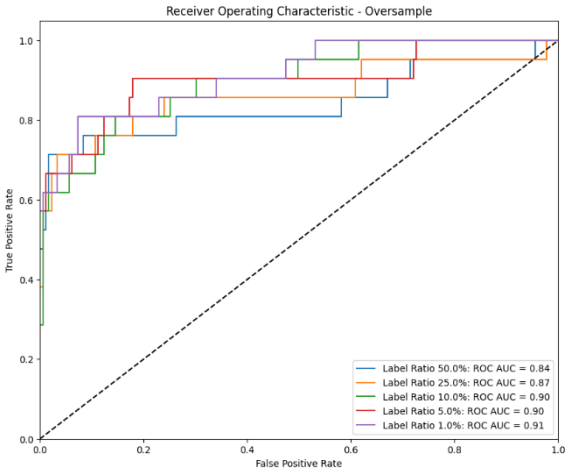
Results:

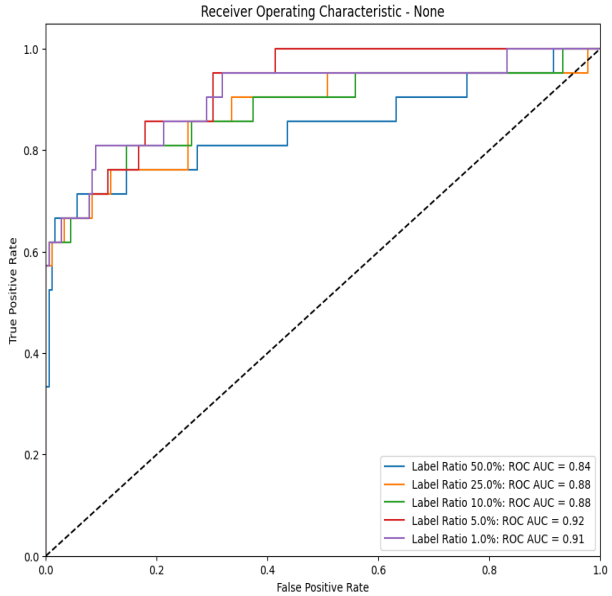
- **None sample** strategy ROC curves demonstrate an increase in AUC as the label ratio decreases, which is somewhat counterintuitive, suggesting that the semi-supervised method might be leveraging the unlabeled data effectively. Particularly, the model performance with a label ratio of 5% stands out, achieving the highest ROC AUC of 0.92, showing that even a small amount of labeled data can be sufficient for the model to learn and generalize well.
- **Oversample** by SMOTE strategy, the AUC is generally high across different label ratios, signifying that SMOTE helps in managing the imbalance even with varying amounts of labeled data. However, the highest AUC is observed at a label ratio of 1%, which might be due to the synthetic data generated by SMOTE compensating for the lack of real labeled examples.
- **Undersample** strategy presents the highest AUC at a 50% label ratio, but it experiences a drop as the label ratio decreases. This indicates that while undersampling may be effective when a moderate amount of labeled data is available, its efficacy wanes with fewer labels, as there is less data to learn from after the undersampling process.

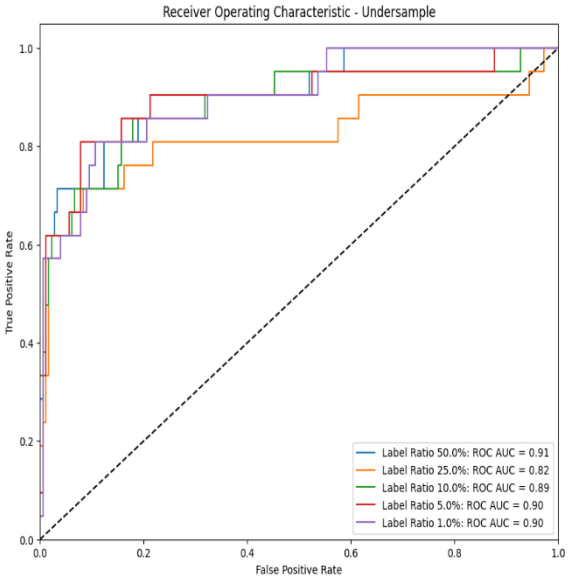
The accompanying performance table further cements these observations, with accuracy and F1 score fluctuations across strategies and label ratios. In particular, oversampling strategies tend to maintain higher F1 scores at lower label ratios compared to the no sampling strategy, showcasing their effectiveness in dealing with class imbalance.

Each sampling strategy has its merits and drawbacks, which become more pronounced at different levels of label scarcity. These findings are crucial for semi-supervised learning applications, highlighting that label ratio and sampling strategy must be carefully calibrated to achieve the best possible performance, particularly when dealing with imbalanced datasets.

	Strategy	Label Ratio	Accuracy	F1 Score	ROC AUC	Runtime
0	none	0.50	0.940	0.647059	0.843841	0.013169
1	none	0.25	0.950	0.705882	0.877095	0.014856
2	none	0.10	0.950	0.705882	0.880287	0.015711
3	none	0.05	0.955	0.742857	0.924448	0.014460
4	none	0.01	0.950	0.705882	0.907688	0.016123
5	oversample	0.50	0.950	0.736842	0.841447	0.014381
6	oversample	0.25	0.940	0.666667	0.865124	0.009654
7	oversample	0.10	0.950	0.705882	0.897845	0.011802
8	oversample	0.05	0.950	0.722222	0.899175	0.014684
9	oversample	0.01	0.955	0.742857	0.913541	0.009141
10	undersample	0.50	0.835	0.507463	0.910881	0.008458
11	undersample	0.25	0.900	0.583333	0.820165	0.009372
12	undersample	0.10	0.870	0.535714	0.885874	0.010332
13	undersample	0.05	0.865	0.557377	0.897313	0.008813
14	undersample	0.01	0.865	0.557377	0.900505	0.009472







2.4 Unsupervised Pretraining

Unsupervised pre-training using autoencoders involves training a series of autoencoder neural networks without labeled data. Each autoencoder learns to compress and reconstruct the input data, initializing the network's parameters and learning meaningful representations. After pre-training, the network is fine-tuned using labeled data to optimize for a specific task such as classification or regression. This approach leverages unlabeled data to guide the learning process and improve the model's generalization performance.

Procedure:

- Build an Autoencoder model with two components encoder and decoders.
- Train the model on each level of unlabeled data.
- Fine tune the model with the available labeled data and calculate loss.
- Evaluate the model performance on the test set and calculated accuracy, precision, recall, and F1-score for each unlabeled data percentage scenario.
- Performed the same steps using both under sampled and over sampled dataset.

Dataset	Unlabeled Data	Accuracy	Precision	Recall	F1-Score	ROC Curve Area
Imbalanced	0.00%	0.8999	0.6281	0.3544	0.4531	0.66
	50.00%	0.8977	0.5887	0.4168	0.4881	0.69
	75.00%	0.8968	0.5912	0.3827	0.4647	0.67
	90.00%	0.8960	0.5821	0.3950	0.4707	0.68
	95.00%	0.8949	0.5818	0.3629	0.4470	0.66
	99.00%	0.8809	0.4816	0.2362	0.3170	0.60
NearMiss	0.00%	0.7173	0.7031	0.7523	0.7269	0.72
	50.00%	0.6947	0.6886	0.7107	0.6995	0.69
	75.00%	0.6862	0.6927	0.6691	0.6807	0.69
	90.00%	0.6677	0.6632	0.6814	0.6722	0.67
	95.00%	0.6217	0.6547	0.6417	0.6482	0.65
	99.00%	0.5047	0.5045	0.5293	0.5166	0.50
Smote	0.00%	0.9373	0.9627	0.9098	0.9355	0.94
	50.00%	0.9363	0.9623	0.9082	0.9344	0.94
	75.00%	0.9363	0.9551	0.9157	0.9350	0.94
	90.00%	0.9360	0.9559	0.9140	0.9345	0.94
	95.00%	0.9302	0.9394	0.9197	0.9295	0.93
	99.00%	0.9048	0.9196	0.8872	0.9031	0.90

The analysis of unsupervised pretraining reveals a nuanced relationship between the level of unlabeled data and model performance. Across different datasets and sampling techniques, varying percentages of unlabeled data exerted notable effects on the model's ability to generalize and make accurate predictions. Particularly, lower levels of unlabeled data tended to result in suboptimal performance, as the model lacked sufficient information to learn meaningful representations. However, as the percentage of unlabeled data increased, the model demonstrated improved performance metrics, indicating a positive correlation between the amount of unlabeled data and model performance.