CSI 5155 – Machine learning Project Report

Name: Abhishek Chandar

Student number: 300222152

Introduction:

Semi-supervised learning (SSL) is a machine learning approach that is implemented in cases where the class labels are not known. The goal of this approach is to leverage the unlabelled data during training along with small set of labelled data. Previous research has proven that semi-supervised techniques provide results as good as a completely supervised learning approach.

In this project, three different semi-supervised approaches namely Self Training, Semi-supervised ensemble and Unsupervised Pretraining are implemented on three datasets from different domains. The implementation of the algorithms in terms of the choice of algorithm, parameter settings are discussed in this report. Additionally, a comparative analysis is conducted among the semi-supervised approaches and the best performing algorithm for each dataset is discussed. The results are supported with evaluation metrics such as F1-score, ROC curve and runtime. Statistical significance testing is also performed to empirically evaluate different semi-supervised techniques implemented in this work.

The contributions from this report are as follows:

1. Identifying the impact of varying the number of labels on model construction and results
2. Investigating the impact of class imbalance on model construction and results

Background:

The datasets are used in this work to perform binary classification are as follows:

1. Online customer intention:
    a. The target class predicts whether a customer will proceed to complete the shopping or not.
2. Marketing Campaign
    a. The target class predicts if the number of Teens is 0 or 1.
3. Heart Disease
    a. This dataset was used to predict whether a patient has heart disease or not

The algorithms implemented on the above-mentioned datasets are discussed further below:

Self-Training:

This approach implements a supervised classifier to repeatedly re-train on its most confident predictions. More specifically, in each iteration, the model is trained on labelled data and makes a prediction on the unlabelled data. The unlabelled data is soft labelled (ie. Pseudo-labelled) and the most confident instances from the unlabelled data is appended to the labelled data and trained further. The stopping criteria of this iteration is when the number of unlabelled data reaches 0 or there are no highly confident instances in the unlabelled data.

Semi-Supervised Ensemble – SEMIBOOST[2]:

In this approach, the concept of pseudo-labelling the data is applied to Boosting methods. SSMBoost is one of the early attempts to implement pseudo-labelling data to Boosting methods. In this approach, a loss function is defined for unlabelled data and the model is trained with the goal of minimizing the loss. In the case of SSMBoost, the algorithm does not assign pseudo labels to unlabelled data points. On the contrary, ASSEMBLE which was developed to overcome the disadvantages of SSMBoost pseudo labelled the unlabelled data after each iteration. Therefore, unlike SSMBoost, there was no need for a semi-supervised base learner. But the disadvantage of ASSEMBLE is that the unlabelled data points are passed on to the next iteration using Random selection specifically by bootstrapping with replacement. To overcome this advantage, SemiBoost was proposed.

Instead of random selection of data points, SemiBoost leverages the concept of manifold assumption and concepts from graph-based methods. Each unlabelled data point is assigned a pseudo-label. The confidence for each unlabelled data point is calculated based on a similarity graph that calculates similarity between data points. Therefore, the similarity between the data points will find the nearest labelled data point and assign the confidence to the unlabelled data points with respect to that labelled data class.

Unsupervised Pretraining using Autoencoders:

Autoencoders are deep learning models that are used to reconstruct the input data. It has two components namely encoders and decoders. As the name suggests, the encoders encode the data into a low dimensional space and the decoders decode the output of the encoder back to the original dimension. The encoder and decoder are connected by a bottleneck layer that uses the encoder output as input. In case of large number of unlabelled data, stacked autoencoders can be used to gain information about the data (both labelled and unlabelled). This step is called pretraining. The autoencoder's encoded output without the unlabelled data can further be used as an input to a supervised classifier to perform classification [3]. By this method, the classifier can learn from labelled as well as unlabelled data and use that information to further perform a supervised classification using the labelled data only. This method ensures that the unlabelled data is utilized in the best way possible.

Experimentation Setup:

Different semi-supervised algorithms are applied to three datasets used in this project. Furthermore, to understand the performance of the semi-supervised algorithms at a holistic level, each dataset was preprocessed to have different level of unlabelled data. These levels range from 0%, 10%, 20%, 50%, 90% and 95% where 0% level is considered as the baseline model for semi-supervised algorithm and in this case, there is no unlabelled data. Apart from the different level of unlabelled data, the first dataset has three version based on the skewness of the data. Finally, each semi-supervised algorithm was experimented using six different base algorithms namely Decision Tree, Support Vector Machines, K-Nearest Neighbors, Random Forest, Multilayer Perceptron and Gradient Boosting algorithm.

In summary, the number of dataset versions sums up to 30 with 6 levels of unlabelled data along with 5 different datasets namely Dataset 1, Oversampled Dataset 1, Under sampled Dataset 1, Dataset 2 and finally Dataset 3.

This experimental setup is used to conduct experiments so that the following questions can be answered:

- Why is a particular SSL algorithm chosen?
- What is the best performing semi-supervised algorithm for each version of dataset?
- What is the impact of the number of labelled data points on the SSL algorithms?
- How can the impact of skewness in data affect the SSL algorithms?

The above-mentioned questions are answered in the next section with supporting results from the experimentation conducted in this project.

Experimentation:

Based on the results from the experimentation performed, the following questions are answered.

1. **Why is a particular SSL algorithm chosen?**

This question has been partially answered in the Background section where the semi-supervised learning algorithm used in this project is briefly discussed. Self training algorithm was experimented with different base learners and the best performing base learner is considered for the SSL algorithm for each dataset. In case of SemiBoost, it was implemented because of the meaningful approach that the algorithm takes to calculate the confidence of the unlabelled data using neighboring data points. Finally, autoencoders are used to learn the pattern in the input data by training an autoencoder to represent an input data and thereby leveraging the unlabelled data as well. The application of autoencoder to construct the input data and passing the generated input data as the input data for a supervised classifier improves the performance of the supervised classifier.

2. **What is the best performing semi-supervised algorithm for each version of dataset?**

In order to identify the best performing SSL algorithm in each dataset version, Friedman test was performed on the F1 score of each algorithm with respect to 30 different versions of dataset. The null hypothesis that was made are as follows:

All the SSL models considered perform similar to each other over different datasets

Thus, the alternate hypothesis was as follows:

All the SSL models perform significantly different from each other

Below is the table that represents the rank of each algorithm for a given version of dataset which was used to perform Friedman test. Based on the results from Friedman at level of significance 95% and 99%, it was concluded that the null hypothesis was NOT rejected because the statistic value = 28 which is less than critical value = 43.77.

Further, the Friedman test table is divided into 5 representing each version of dataset and the best performing algorithm is explained. For each dataset, the total number of rank 1 is calculated which is referred to as the “Best count” because rank 1 means that the algorithm has performed the best among

the other algorithms for a given version of dataset. The tables below show the ranking of each algorithm on different levels of unlabelled data for each version of dataset.

Table I: Rank algorithms for Imbalance data

Dataset_unlabelled	Self Training	SemiBoost	Unsupervised Pretraining
Imbalanced_0%	1	3	2
Imbalanced _10%	2	3	1
Imbalanced _20%	2	3	1
Imbalanced _50%	2	3	1
Imbalanced _90%	2	3	1
Imbalanced _95%	2	3	1
Best Count	1	1	**5**

Table II: Rank algorithms for Oversampled data

Dataset_unlabelled	Self Training	SemiBoost	Unsupervised Pretraining
Oversampled_0%	1	2	3
Oversampled_10%	3	2	1
Oversampled_20%	1	1	3
Oversampled_50%	1	3	2
Oversampled_90%	1	3	2
Oversampled_95%	1	3	2
Best Count	**5**	1	1

Table III: Rank algorithms for Imbalance data

Dataset_unlabelled	Self Training	SemiBoost	Unsupervised pretraining
Undersampled_0%	2	1	3
Undersampled _10%	1	3	2
Undersampled _20%	1	3	2
Undersampled _50%	1	3	2
Undersampled _90%	1	3	2
Undersampled _95%	1	3	2
Best Count	**5**	1	0

Dataset_unlabelled	Self Training	SemiBoost	Unsupervised Pretraining
Dataset2_0%	1	3	2
Dataset2_10%	2	1	3
Dataset2_20%	1	2	3
Dataset2_50%	1	3	2
Dataset2_90%	1	3	2
Dataset2_95%	1	3	2
Best Count	**5**	1	0

Dataset_unlabelled	Self Training	SemiBoost	Unsupervised Pretraining
Dataset3_0%	1	3	2
Dataset3_10%	2	1	3
Dataset3_20%	1	2	3
Dataset3_50%	1	3	2
Dataset3_90%	1	3	2
Dataset3_95%	1	3	2
Best Count	**5**	1	0

For the imbalanced dataset, the best performing algorithm is Unsupervised Pretraining with the best count value equal to 5 which is greater than the best count for Self training and SemiBoost. In this dataset, the autoencoder implemented had 2 encoders and 2 decoders stacked and the base algorithm used for supervised classification was K-Nearest Neighbors with the following hyperparameters after performing RandomSearch hyperparameter tuning. The hyperparameters are:

- n_neighbors = 9
- metric = ”Euclidean”
- weights =” Uniform”

In the case of oversampled dataset, the best count for Self training is the highest with a value of 5 when compared to SemiBoost and Unsupervised Pretraining. Therefore, this implicitly means that the model has achieved higher F1-score in more of the unlabelled cases for the oversampled dataset. After hyperparameter tuning, the base algorithm implemented in this Self training model is a Random Forest with the hyperparameters as follows:

- n_estimators = 110
- min_samples_split = 5
- max_features = “Auto”
- max_depth = 23

Finally, in the case of Under sampled, Dataset 2 as well as Dataset 3, it was found that Self Training performed better than the other two SSL algorithms because the Best count is equal to 5. The ROC plot for best performing algorithm in each dataset version is depicted below:

Figure 1 ROC plot - Self training for imbalance data

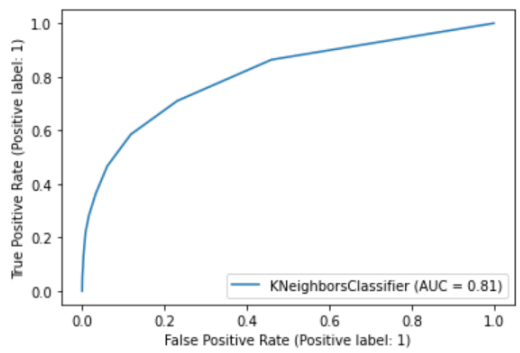


Figure 2 ROC plot - Semi Boost for oversampled data

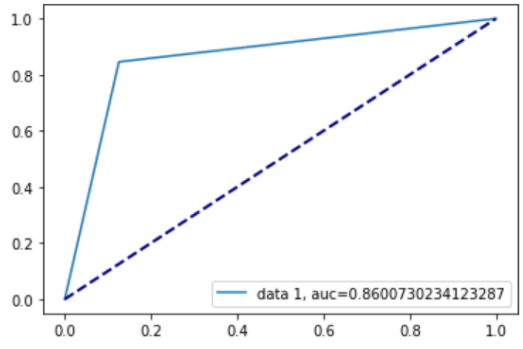


Figure 3 ROC plot - Self training for undersampled data

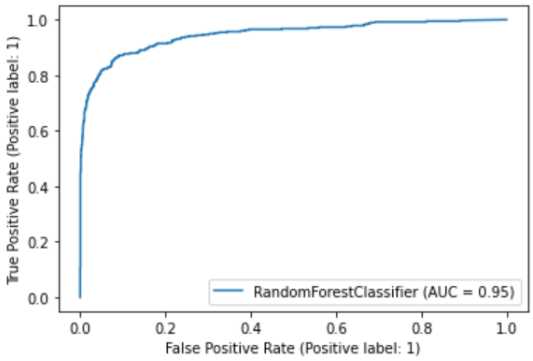


Figure 4 ROC plot - Self training for Dataset 2

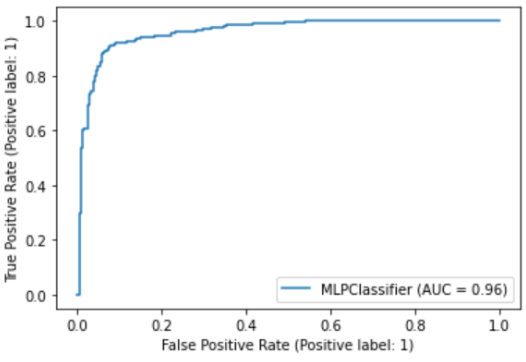
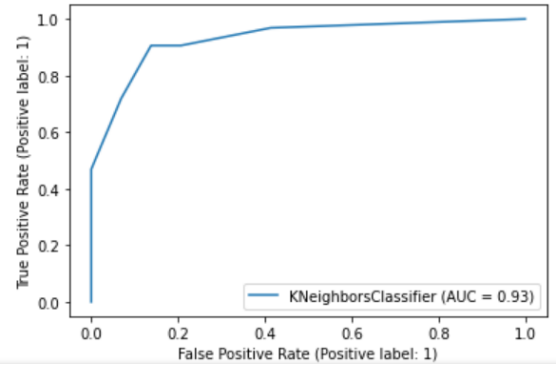


Figure 5 Self training model for Dataset 3



In summary, by calculating the best count value for the three algorithms for all datasets, we can observe that:

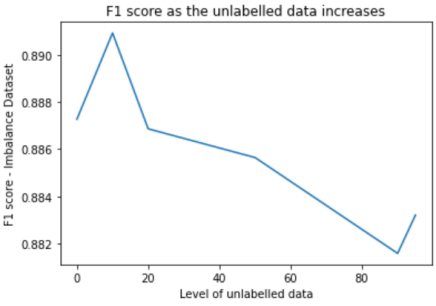
- Self training has attained rank 1 in 22 occurrences
- Semi supervised ensemble (SemiBoost) has attained rank 1 in 3 occurrences
- And finally, Unsupervised pretraining has attained rank 1 in 6 cases.

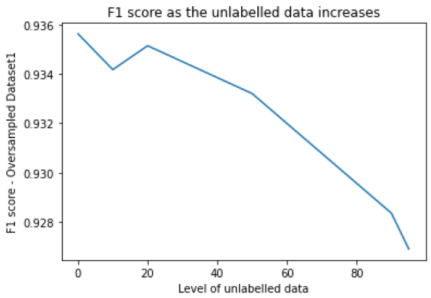
Therefore, we can conclude that Self training has performed the best overall when compared to the other two SSL algorithms.

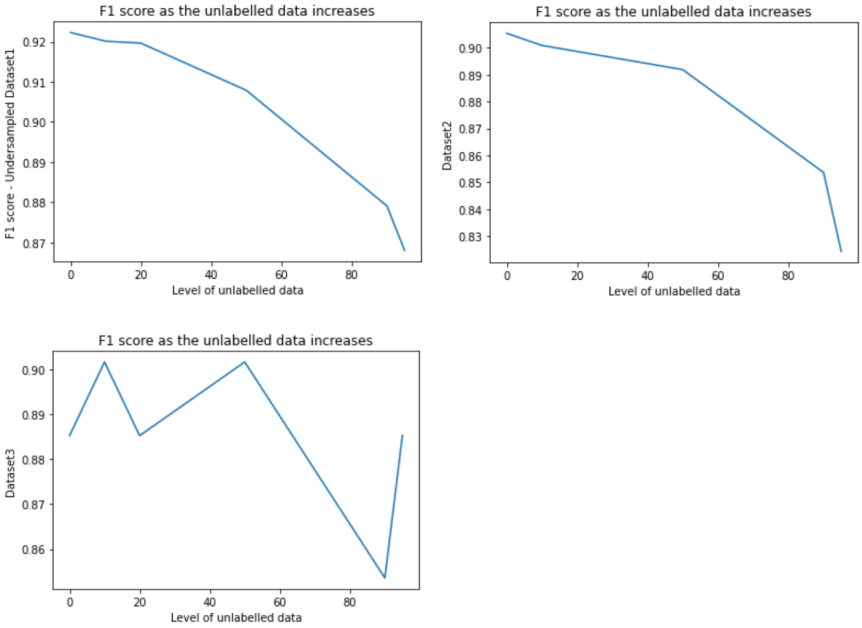
3. **What is the impact of the number of labelled data points on the SSL algorithms?**

During the experimentation, the three SSL algorithms were performed on different version of data based on the level of unlabelled data ranging from 0% to 95% to study the impact of the number of labelled data points on these algorithms. The line plots below show the F1 score for an algorithm over increasing levels of unlabelled data. These plots have been displayed for all the algorithms below.

Plots for Self training for each dataset version:

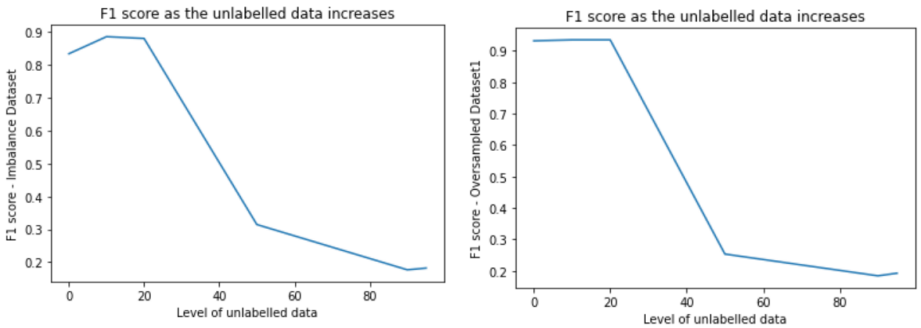


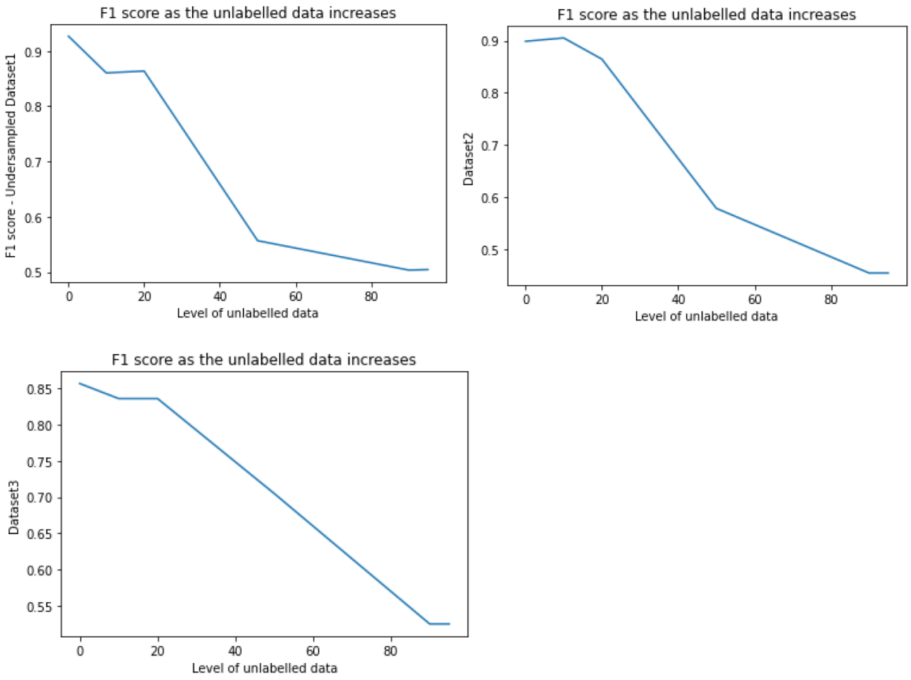




Based on the plots shown above, it can be observed that the F1-score has a decreasing trend as the percentage of unlabelled data increases except for Dataset 3. It is also important to note that the first point of the plot represents the baseline F1 score. Therefore, it can be concluded that Self training algorithm performs relatively worse than the baseline case. In this set of experimentation, dataset 3 is an outlier as Self training has performed better than the baseline when the level of unlabelled data is 50%. One reason for this could be the threshold value set for Self training that determines which pseudo-labelled data points needs to be appended to the labelled data. The threshold was set to 60% after hand-tuning.

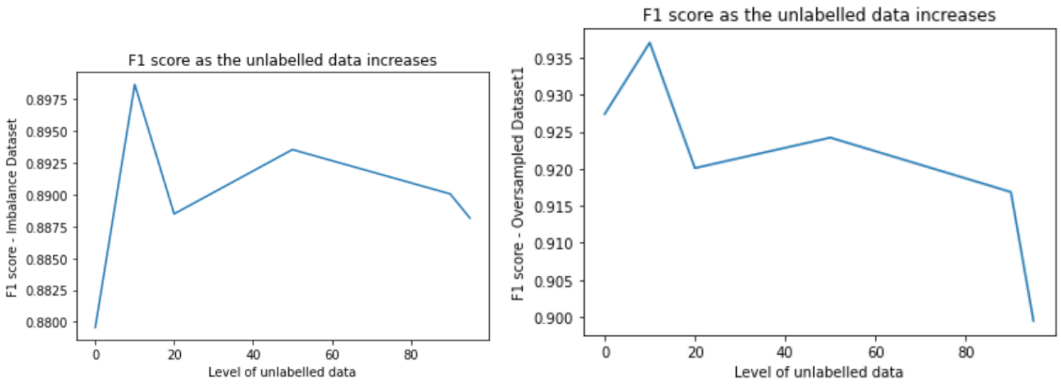
Plots for Semi supervised Ensemble for each dataset version:

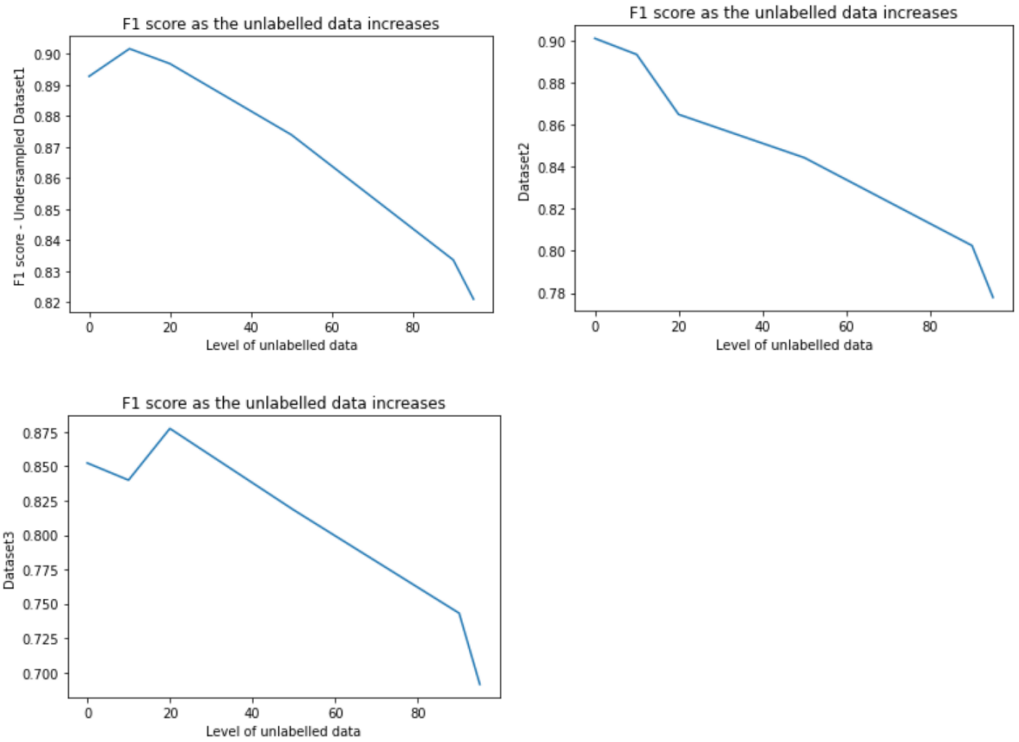




In the case of SemiBoost, we can see a similar trend in the F1 curve where the score decreases as the level of unlabelled data is increased. Unlike Self training, there are no outliers found in this case. The performance drop when the unlabelled data is increased can be attributed to the algorithm performing mistakes in calculating the similarity between the K nearest data points especially when the number of labelled data is less. In case of SemiBoost, the drop in the F1 score as the level of unlabelled data is increased is significantly higher than the other two models because of the underlying algorithm being a similarity graph-based algorithm (KNN in this case) which usually suffers in large datasets.[4]

Plots for Unsupervised pretraining for each dataset version:





Finally, Unsupervised pretraining also provides similar graphs when compared with both SemiBoost as well as Self training.

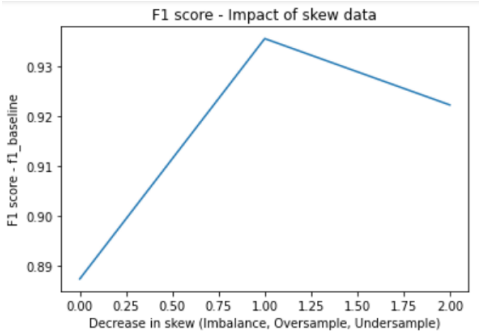
In summary, the performance of the semi supervised learning algorithms decreases as the level of unlabelled data increases.

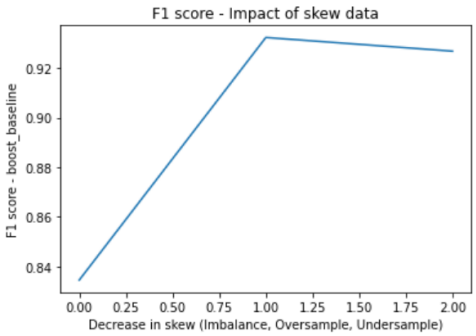
4. How can the impact of skewness in data affect the SSL algorithms?

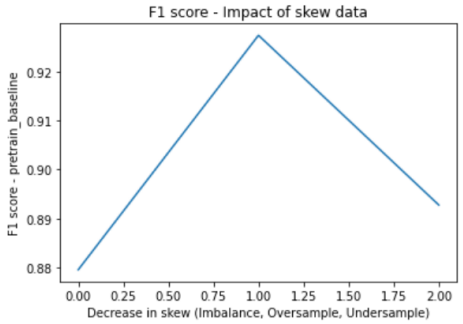
To answer this question, the F1 score of all the models were observed for the first dataset which was imbalanced in nature. Further, the dataset was rebalanced by performing oversampling using SMOTE and undersampling using Edited Nearest Neighbours (ENN). The X axis of the plots below represent order of imbalanced version, oversampled version and undersampled version. Therefore, each edge of the line shown in the graph represents the F1 score for a particular version of dataset. This plot was plotted for all the levels of unlabelled data and for all the algorithms.

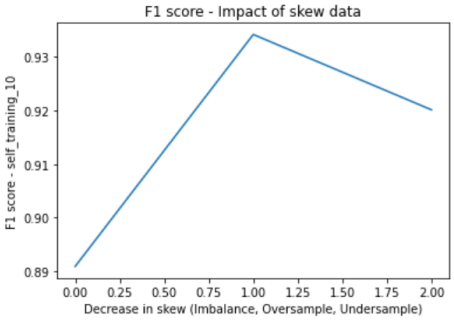
Based on the plot, the F1 score of the models increase from imbalanced data to oversampled data and drops again in the case of undersampled data. One reason is because the oversampled data represents the actual scenario when compared to undersampling. This means that the synthetic data generated did not affect the distribution of the data. In the case, undersampling, the models performed poorly since majority class data points are reduced and as a result of that, the data might not be truly representative of the actual scenario.

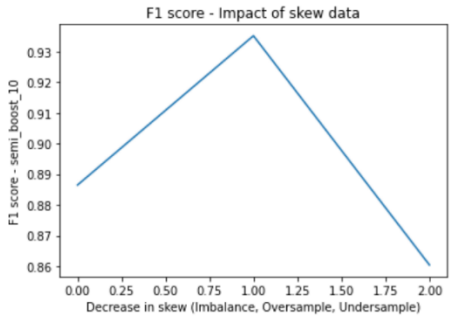
Plots showing change in F1 score as skew changes for each dataset version:

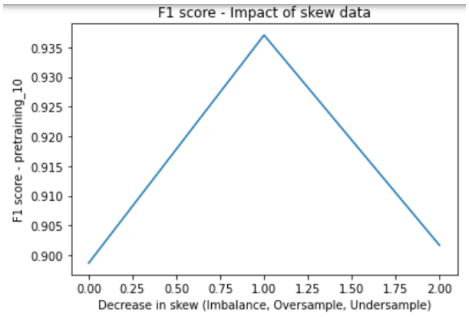


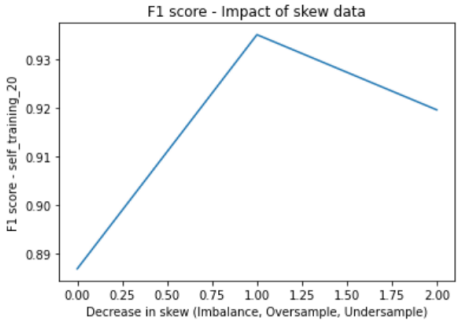


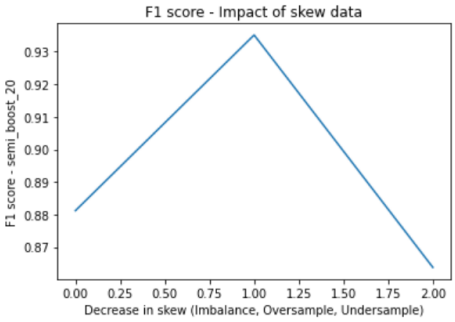


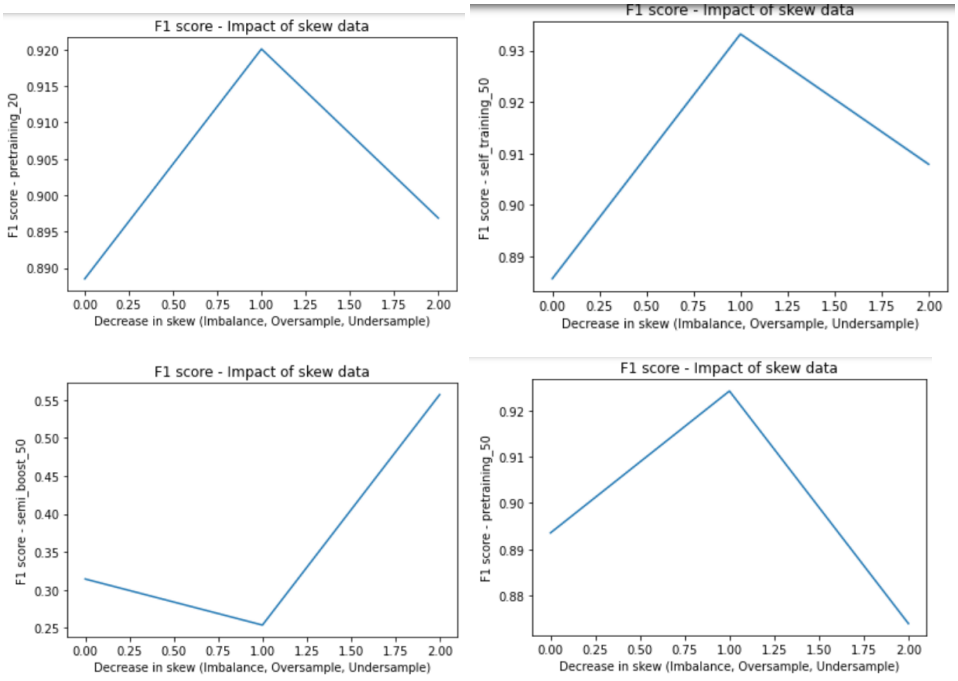












Based on the findings from this experiment, we can say that using unlabelled data in the form of pseudo labels have managed to produce similar F1-scores in the imbalanced data even in the case when level of unlabelled data is more than 50%. This can be verified by the diagram below that represents the Friedman test table that compares the F1-score of different algorithms for all possible dataset versions.

	Dataset	Unlabelled %	Self Training	Semi supervised ensemble	Unsupervised pretraining
0	Imbalanced	0	0.887267	0.834547	0.879562
1	Imbalanced	10	0.890916	0.886456	0.898683
2	Imbalanced	20	0.886861	0.881184	0.888495
3	Imbalanced	50	0.885645	0.314274	0.893552
4	Imbalanced	90	0.881590	0.176399	0.890065
5	Imbalanced	95	0.883212	0.181671	0.888166

Quick note on the impact of level of unlabelled data on training time:

As the level of unlabelled data increases, the training time for Self training is increasing. This is because there are more unlabelled data points that needs to be appended to the labelled data until the stopping criterion is met. In the case of SemiBoost, the training time decreases as the unlabelled data increases. This is because in the case of more unlabelled data, the calculation of similarity between data points is faster because of scarce number of data points. Finally, the training time for Unsupervised pretraining decreases as the number of unlabelled data increases. In this case, the time taken to train the autoencoder is going to remain the same in all cases since both the labelled and unlabelled data is considered in that

case. The time difference happens in the supervised classifier part where only the labelled data is used for training the data. Therefore, as the number of unlabelled data points is increased, the training time decreases.

Final remarks & Lessons learned:

1. SSL algorithms is a promising option when the dataset consists of large number of unlabelled data
2. The performance of SSL algorithms decreases in most cases as the level of unlabelled data increases
3. The performance of SSL algorithms increases in the case of oversampled data
4. The performance of SSL algorithms is as good as a fully supervised algorithm even when the level of unlabelled data is greater than 10%
5. Semi Boost suffers in case of large datasets because it uses KNN for similarity calculated between nearest data points

References:

1. van Engelen, J.E., Hoos, H.H. A survey on semi-supervised learning. Mach Learn 109, 373–440 (2020). https://doi.org/10.1007/s10994-019-05855-6
2. https://github.com/papabloblo/semi_boost/blob/master/src/SemiBoost.py
3. https://machinelearningmastery.com/autoencoder-for-classification/
4. https://lilianweng.github.io/lil-log/2021/12/05/semi-supervised-learning.html