A blend of Supervised and UnSupervised Learning from Imbalanced Datasets

April 28, 2022

Registration number: 2101194
Project: Imbalanced datasets
Link to GitHub: https://github.com/ahmadraza346/Learning-From-Imbalanced-DataSets-CE888-

Executive summary (max. 250 words)	245
Introduction (max. 600 words)	597
Data (max. 300 words/dataset)	176, 234, 196
Methodology (max. 600 words)	561
Results and Discussion (max. 1000 words combined)	852
Conclusions(max. 500 words)	223
Total word count	3084

Contents

1 Introduction **3**

2 Data **4**
2.1 Mushroom Data Set . . . 4
2.2 Gender Data Set . . . 4
2.3 Heart Data Set . . . 4

3 Methodology **5**
3.1 Creation of Surrogate Data Sets from Original Data Sets . . . 5
3.1.1 Mushroom Data Surrogate Data Sets . . . 5
3.1.2 Gender Data Surrogate Data Sets . . . 5
3.1.3 Heart Data Surrogate Data Sets . . . 6
3.2 Ensemble Model Proposed in the Project . . . 6
3.2.1 Stratified Cross-Validation on Random Forest as Base Model . . . 7
3.2.2 Creation of Stratified folds for Proposed Model . . . 7
3.2.3 Creating Clusters with the help of K Mean Clustering . . . 7
3.2.4 Training Random Forest Classifier on Clusters . . . 7
3.2.5 Testing the Model . . . 7
3.2.6 Model Results . . . 7

4 Results **8**
4.1 Gender Dataset Results . . . 8
4.1.1 Results for balanced Gender Dataset . . . 8
4.1.2 Results for Low Imbalanced Gender Dataset . . . 9
4.1.3 Results for Medium Imbalanced Gender Dataset . . . 9
4.1.4 Results for High Imbalanced Gender DataSet . . . 9
4.2 Heart Dataset Results . . . 9
4.2.1 Results for Balanced Heart Dataset . . . 9
4.2.2 Results for Low Imbalanced Heart Dataset . . . 10
4.2.3 Results for Medium Imbalanced Heart Dataset . . . 10
4.2.4 Results for High Imbalanced Heart Dataset . . . 10
4.3 Mushrooms Dataset Results . . . 10

5 Discussion **10**

6 Conclusions **11**

Abstract

A very common difficulty that classification models come across is the imbalanced datasets. As the classes in these datasets doesn't have equal representation, therefore the models trained on such data are compromised because minority class is ignored while training process. Many solutions have been proposed to this problem in past but most popular practice is to re-sample the data where either the majority class is down-sampled or minority class is up-sampled. This paper proposed a new model to learn from imbalanced datasets by combining the supervised and unsupervised learning. For this project, three initially balanced publicly available datasets were chosen, cleaned, pre-processsed and their surrogates were created with low, medium and high imbalance. During deployment, random forest classifier was used to perform stratified cross-validation on all three datasets and their surrogates as a baseline. Proposed model also used a method similar to stratified cross-validation but using K-mean clustering along with Random Forest Classifier, where RFC was used to train the clusters identified by K-Mean that had samples from more than one classes. To compare results obtained from both models, box plots were drawn and permutation tests were performed on accuracy and F1 scores obtained for all datasets and their surrogates. It was observed that our model outperformed the baseline model only in the case of very high data imbalance. Also, as the data imbalance increased, the performance of our model kept getting better. As a conclusion, higher the data imbalance, higher the proposed model performance.

1 Introduction

Now a days, real world applications and important decision making processes heavily rely on the tons of data coming from various sources. During past few years, the flow of this incoming data has increased drastically due to the rapid pace of advancement in almost every field especially the fields of Science and Technology. But with the increased inflow of data, data related problems have also arose, such as ML models suffering from imbalanced datasets.[4] An imbalanced dataset is the one that contains very less samples of one class whereas majority of samples belong to the rest of class/classes. If dataset is biased towards specific class/classes, results generated by ML techniques are biased towards the class to which majority of the samples belong. ML algorithms perform well when the data is unbiased. [3] Among many other factors, one important factor that these imbalanced datasets exist is the rare existence of the examples of minority class in real world scenarios. For instance, bank datasets have very few samples of fraud transactions because in real world normal transactions happen every minute but the fraud transactions are committed only once in a while.

When it comes to the learning of classifiers from imbalanced datasets, accuracy metric for the class with majority instances produce unrealistically excellent results where as in case of the minority class, results are quite poor.[2] Classification techniques make an assumption that all classes in dataset are balanced and the cost of errors produced by each class is same therefore they try to minimize the combined error. Due to this accuracy driven wrong approach, these algorithms produce very poor results for imbalanced datasets.[9] In actual, there is a possibility that we bear a lot more cost for incorrectly classifying imbalanced datasets. For instance in case of cancer diagnosis, if a person who has cancer is mis-classified as not having cancer, it can cost his life whereas if a person who doesn't have cancer is diagnosed for cancer, costs are not going to be as drastic as in first case.[2]

To tackle the problem of data imbalance, many efforts have been made in the past including the modification of existing classifiers and modification of actual datasets. Another alias for the modification of dataset is re-sampling of data. In re-sampling of datasets, either the class with majority instances is down-sampled or minority class is up-sampled. If down-sampled, the examples from majority class are deleted to balance both classes. If over-sampled, minority class is fed with new examples to create balance. Many pros and cons are associated with data modification. Such as, in case of down-sampling, we can lose important information while deleting rows whereas in opposite case our learning time increases and a model can suffer from over-fitting.[6]

This paper has blended supervised learning with unsupervised learning. The motivation was to build an ensemble model that can tackle the problem of data imbalance without the manipulation of actual datasets. We performed a methodology similar to stratified cross-validation using a blend of K-mean clustering along with Random Forest Classifier. F1 score was used as main evaluation metric along with the accuracy because accuracy tends to over-look the minority class in case of high data imbalance as ML models are accuracy-driven. [5] Boxplots and Permutation tests were used to see the improvement in proposed model from baseline model. Boxplots help in comparing results of both models. Further, those results were validated by performing permutation tests as

they gives the probability that the difference observed between the results of both models could have been by chance as datasets are usually only a sample of larger population.[7]

2 Data

We chose three balanced datasets for our project that are publicly available on kaggle. Reason behind the selection of these datasets was their balanced ratio for both classes as we needed balanced datasets initially so that we could create imbalanced surrogates from these datasets as we progressed in our project to perform different experiments.

2.1 Mushroom Data Set

This dataset is taken from kaggle and is known as Mushroom data set. It is a binary data set that originally classifies mushrooms into two classes. I.e. poisonous(p) and edible(e). Before any pre-processing steps, the data set contains 8124 rows and 23 columns out of which 22 columns are the feature columns and one column is the target column. Data set is balanced and contains 3916(48.2 percent) examples of class poisonous(p) and 4208(51.8 percent) examples of class edible(e). All columns belonging to the data set have categorical values. I.e. all features as well as the target column contains categorical data. Data set contains no duplicate or null data.

During the pre-processing of the data, categorical features were encoded using one hot encoding instead of nominal encoding as our features have nominal values therefore we cannot use ordinal encoding as it will rank them. Furthermore, target column was encoded using label encoding. Posterior to pre-processing, feature vector size had increased to 117. Further insights gathered during data exploration are available in the code uploaded on git hub.

2.2 Gender Data Set

This data set is a small data set available on kaggle and is known as Gender Classification data set. The data is binary in nature and was collected in 2015 from university students of 21 nationalities studying different majors in different countries to predict the gender as Male(M) or Female(F) based on personal preferences by performing binary classification. Before any pre-processing steps, the data set contained 66 rows and 5 columns out of which 4 columns are the feature columns and one column is the target column. Data set is perfectly balanced with 50 Percent(33) instances from both male and female classes. All columns belonging to the data set have categorical values. I.e. all features as well as the target column contains categorical data. Data set contains 4 duplicate examples but no null data.

During the pre-processing of the data, duplicate examples were deleted and categorical features were encoded using one hot encoding. Furthermore, target column was encoded using label encoding. Posterior to encoding, feature vector size had increased to 20 but we dropped one column for each feature because while performing one hot encoding dummy variables are created resulting in strong correlation among at least two variables. This issue is known as dummy variable trap. After performing all pre-processing steps, we have 16 variables and 62 instances of data. Further insights gathered during data exploration are available in the code uploaded on git hub.

2.3 Heart Data Set

This data set is a small data set available on kaggle and is known as Heart Attack Classification data set. It is a medium sized binary data set that originally predicts the chances of a heart attack in person. I.e. 1(more chances of heart attack) and 0(less chances of heart attack). Before any pre-processing steps, the data set contains 303 rows and 14 columns out of which 13 columns are the feature columns and one column is the target column. Data set is almost balanced and contains 165(54.5 percent) examples of class 1 and 138(45.5 percent) examples of class 0. Apparently, all columns belonging to the data set have numerical values. I.e. all features as well as the target column contains numerical data. Data set contains 1 duplicate example but no null data.

During the pre-processing of the data, duplicate example was removed. Features were not scaled as both baseline and proposed method uses Random Forest Classifier for training and Random Forest Classifier doesn't necessarily need any standardization. After performing all pre-processing steps, we have 13 feature columns and 302 instances of data. Further insights gathered during data exploration are available in the code uploaded on git hub.

3 Methodology

3.1 Creation of Surrogate Data Sets from Original Data Sets

In order to observe the effect of different ratios of imbalance on different data sets, we selected three initially balanced datasets and after performing exploratory and pre-processing steps on them, three surrogates where created for each dataset where first surrogate was created with low ratio of imbalance(65:35), second with medium ratio of imbalance(75:25) and third with highest ratio of imbalance(90:10). These surrogates were achieved by randomly sub-sampling from the minority class. Now, for deployment phase, we have three datasets with 9 surrogates in total.

3.1.1 Mushroom Data Surrogate Data Sets

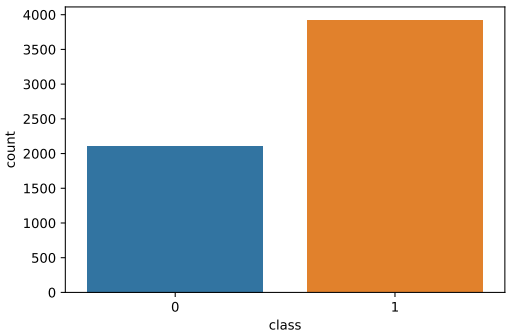

Figure 1: Mushroom Data Low Imbalance Surrogate Data Set.

- Low imbalance Surrogate for this dataset contains 3916 and 2109 rows for majority and minority class respectively. Figure1
- Medium imbalance Surrogate for this dataset contains 3916 and 1307 rows for majority and minority class respectively.
- High imbalance Surrogate for this dataset contains 3916 and 436 rows for majority and minority class respectively.

3.1.2 Gender Data Surrogate Data Sets

- Low imbalance Surrogate for this dataset contains 30 and 16 rows for majority and minority class respectively.

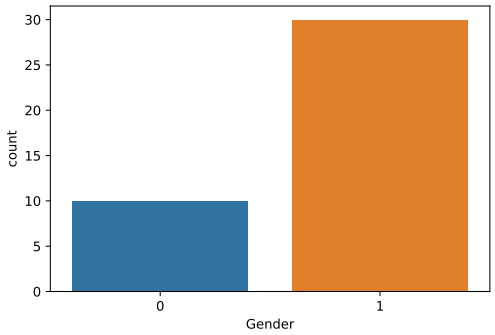

Figure 2: Gender Data Medium Imbalance Surrogate Data Set.

- Medium imbalance Surrogate for this dataset contains 30 and 10 rows for majority and minority class respectively. Figure 2
- High imbalance Surrogate for this dataset contains 27 and 3 rows for majority and minority class respectively.

3.1.3 Heart Data Surrogate Data Sets

- Low imbalance Surrogate for this dataset contains 164 and 88 rows for majority and minority class respectively.
- Medium imbalance Surrogate for this dataset contains 164 and 54 rows for majority and minority class respectively.

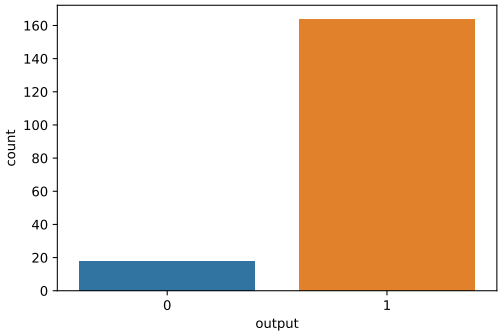

Figure 3: Heart Data High Imbalance Surrogate Data Set.

- High imbalance Surrogate for this dataset contains 164 and 18 rows for majority and minority class respectively. Figure 3

3.2 Ensemble Model Proposed in the Project

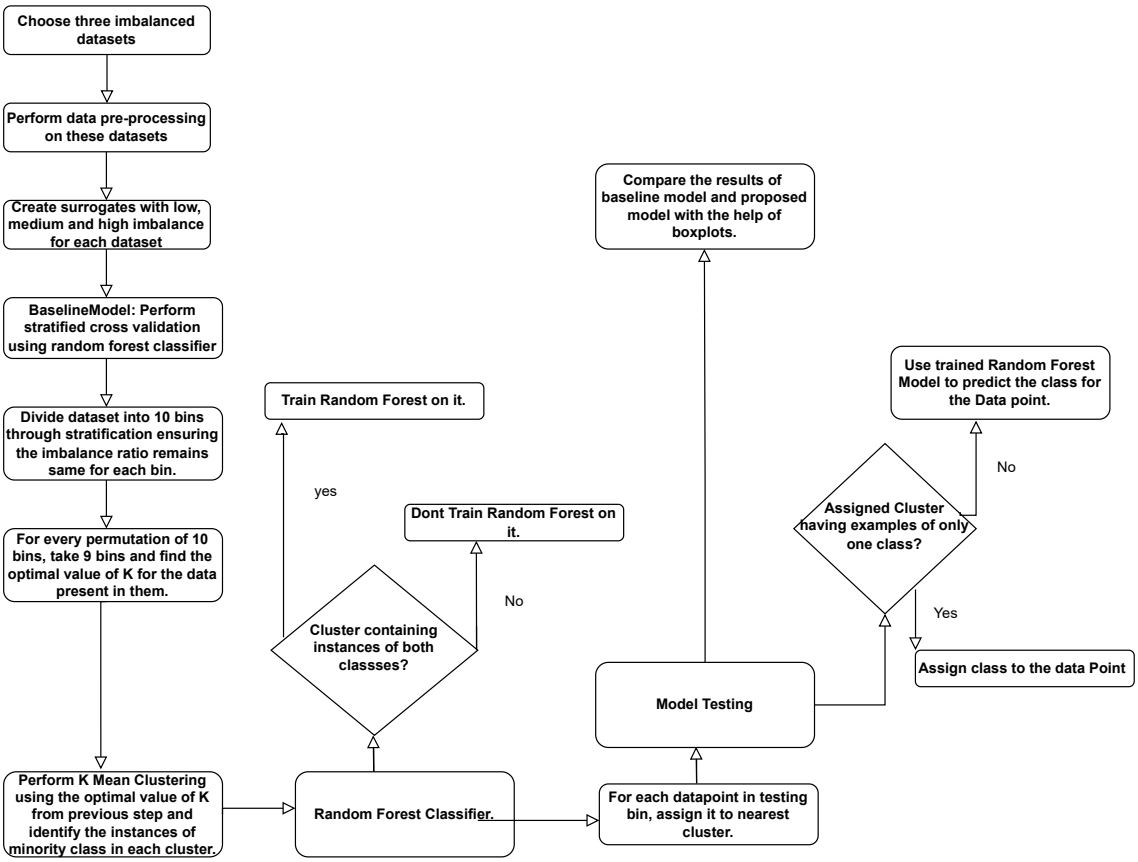

Figure 4: Sequence of events elaborating the flow of the project

3.2.1 Stratified Cross-Validation on Random Forest as Base Model

To build our base model, we performed stratified cross-validation on data sets and their respective surrogate data sets using Random Forest Classifier. Stratification ensured that data imbalance remained same in every fold created during cross-validation. As a result, accuracy and F1 scores were obtained and saved to compare with proposed model.

3.2.2 Creation of Stratified folds for Proposed Model

To ensure that the imbalance ratio is not disturbed, data was again divided into 10 stratified folds this time for the proposed model. This was done for every data set and its respective surrogate data sets.

3.2.3 Creating Clusters with the help of K Mean Clustering

Data in 9 folds was dedicated for training data where as 1 fold was allotted to testing data. Graphs were drawn using elbow and silhouette method [1], [8] to determine the optimal value of K. Values obtained from these graphs were used to define the upper and lower limits. K Mean algorithm was trained using values within these limits and the value of K was selected for which F1 score was the highest. Centroid values and number of samples of minority class in each cluster were calculated and saved to be used in next steps.

3.2.4 Training Random Forest Classifier on Clusters

Random Forest classifier was trained only on the clusters having instances of at least two classes. Here the instances of minority class saved in last step helped us determine whether to train Random Forest Classifier on a respective cluster or not.

3.2.5 Testing the Model

We assigned nearest cluster to the instances of unseen fold. If the assigned cluster was trained on RFC, we predict the value using the trained RFC model otherwise if cluster contains instances of only one class, we assign that class to the instance of the test data. The above process starting from cluster creation till testing was repeated for every permutation of the 10 created folds.

3.2.6 Model Results

After all the iterations, we calculated the F1 and accuracy scores for all the permutations, found their average and standard deviations and saved them to compare with the results of the base model in the results section. Project flow is replicated in Figure 4

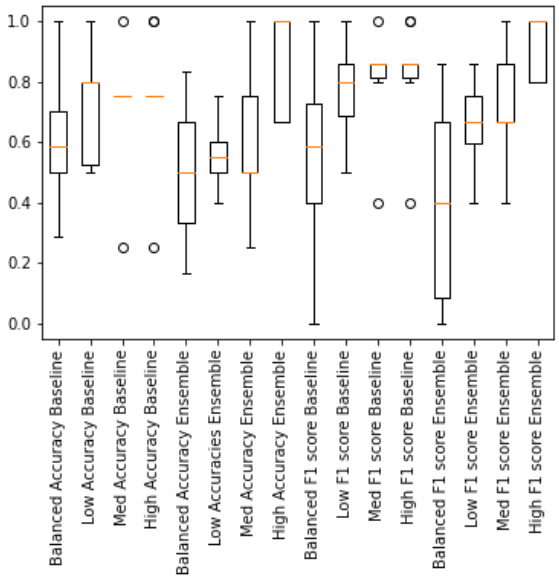

Figure 5: Boxplot for Gender Data Set Surrogate Results.

4 Results

As discussed in the methodology section earlier, the proposed ensemble model also uses stratified cross validation like the base model, therefore to compare the results of both models for different surrogates, boxplots of cross-validated accuracy and F1 scores were formed. Figures 5, 6, 7 contains the boxplot for Gender, Heart and Mushrooms surrogates respectively. Table 1 generates a summary of cross-validated accuracy and F1 scores for all the datasets and their possible surrogates. After comparison, the surrogates for which proposed model performed better than the base model were subjected to permutation tests using the performance values obtained from each fold to check if it actually gives significantly better results.i.e to which extent it differs from the baseline model. Permutation tests results can be seen in table 2.

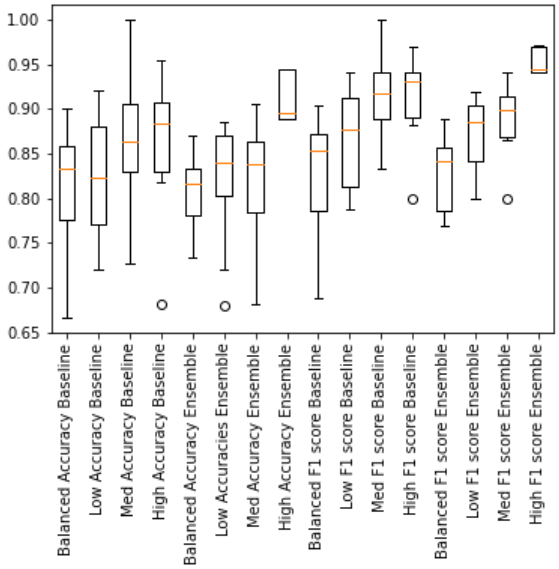

Figure 6: Boxplot for Heart Data Set Surrogate Results.

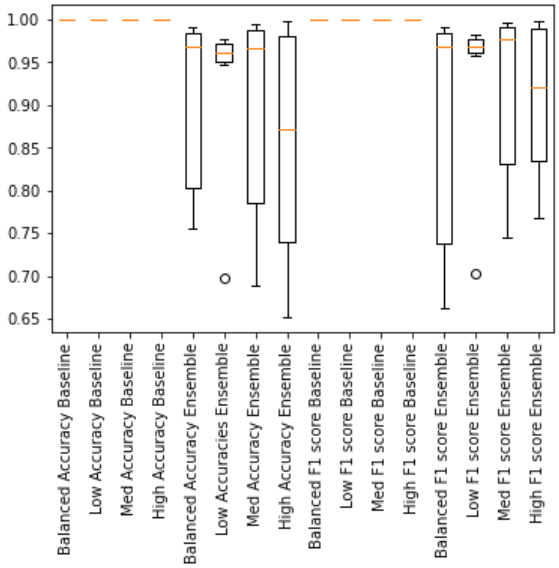

Figure 7: Boxplot for Mushrooms Data Set Surrogate Results.

4.1 Gender Dataset Results

4.1.1 Results for balanced Gender Dataset

Baseline model produced better results with balanced Gender dataset giving an accuracy and f1 score of 0.60 and 0.55 respectively as compared to proposed model giving accuracy and f1 scores

DataSet	Model	Surrogate	Accuracy	F1 Score
Gender	Baseline	Balanced	0.6	0.55
Gender	Baseline	Low Imbalanced	0.71	0.78
Gender	Baseline	Medium Imbalanced	0.72	0.81
Gender	Baseline	High Imbalanced	0.75	0.83
Gender	Ensembled	Balanced	0.50	0.41
Gender	Ensembled	Low Imbalanced	0.55	0.66
Gender	Ensembled	Medium Imbalanced	0.62	0.73
Gender	Ensembled	High Imbalanced	0.87	0.92
Heart	Baseline	Balanced	0.80	0.82
Heart	Baseline	Low Imbalanced	0.83	0.87
Heart	Baseline	Medium Imbalanced	0.87	0.92
Heart	Baseline	High Imbalanced	0.87	0.92
Heart	Ensembled	Balanced	0.81	0.83
Heart	Ensembled	Low Imbalanced	0.82	0.87
Heart	Ensembled	Medium Imbalanced	0.82	0.89
Heart	Ensembled	High Imbalanced	0.91	0.95
Mushroom	Baseline	Balanced	1.0	1.0
Mushroom	Baseline	Low Imbalanced	1.0	1.0
Mushroom	Baseline	Medium Imbalanced	1.0	1.0
Mushroom	Baseline	High Imbalanced	1.0	1.0
Mushroom	Ensembled	Balanced	0.91	0.88
Mushroom	Ensembled	Low Imbalanced	0.94	0.94
Mushroom	Ensembled	Medium Imbalanced	0.89	0.92
Mushroom	Ensembled	High Imbalanced	0.85	0.90

Table 1: Summary of results for all datasets/surrogates for baseline and proposed model.

of 0.50 and 0.41 respectively. Table 1

4.1.2 Results for Low Imbalanced Gender Dataset

For low imbalance surrogate of Gender dataset, baseline model produced better results giving an accuracy and f1 score of 0.71 and 0.78 respectively as compared to proposed model giving accuracy and f1 scores of 0.55 and 0.66 respectively. Table 1

4.1.3 Results for Medium Imbalanced Gender Dataset

For medium imbalance surrogate of Gender dataset, baseline model again produced better results giving an accuracy and f1 score of 0.72 and 0.81 respectively as compared to proposed model giving accuracy and f1 scores of 0.62 and 0.73 respectively. Table 1

4.1.4 Results for High Imbalanced Gender DataSet

Proposed model outperformed baseline model in case of highly imbalanced Gender dataset giving an accuracy and f1 score of 0.87 and 0.92 respectively as compared to base model giving accuracy and f1 scores of 0.75 and 0.83 respectively. To further strengthen our claim, permutation test performed gave p value 0.67 and 0.067 for accuracy and f1 score respectively. P value for f1 score ensures that 94 percent of the times our model outperforms the base model. Table 2

4.2 Heart Dataset Results

4.2.1 Results for Balanced Heart Dataset

Proposed model outperformed baseline model in case of balanced heart dataset giving an accuracy and f1 score of 0.81 and 0.83 respectively as compared to base model giving accuracy and f1 scores of 0.80 and 0.82 respectively. To further strengthen our claim, we performed permutation test but it gave values of p as 0.45 and 0.41 for accuracy and f1 respectively which tells us that proposed model doesn't necessarily improve the base model. Table 2

DataSet	Surrogate	p value for Accuracy	p value for F1 Score
Gender	Balanced	0.83	0.85
Gender	Low Imbalanced	0.99	0.96
Gender	Medium Imbalanced	0.78	0.85
Gender	High Imbalanced	0.67	0.066
Heart	Balanced	0.45	0.41
Heart	Low Imbalanced	0.61	0.50
Heart	Medium Imbalanced	0.93	0.90
Heart	High Imbalanced	0.05	0.009
Mushroom	Balanced	1.0	1.0
Mushroom	Low Imbalanced	1.0	1.0
Mushroom	Medium Imbalanced	1.0	1.0
Mushroom	High Imbalanced	1.0	1.0

Table 2: P values for Accuracy and F1 scores.

4.2.2 Results for Low Imbalanced Heart Dataset

For low imbalance surrogate of Heart dataset, baseline model produced better results giving an accuracy and f1 score of 0.83 and 0.87 respectively as compared to proposed model giving accuracy and f1 scores of 0.82 and 0.87 respectively. Table 1

4.2.3 Results for Medium Imbalanced Heart Dataset

For medium imbalance surrogate of Heart dataset, baseline model again produced better results giving an accuracy and f1 score of 0.87 and 0.92 respectively as compared to proposed model giving accuracy and f1 scores of 0.82 and 0.89 respectively. Table 1

4.2.4 Results for High Imbalanced Heart Dataset

Proposed model outperformed baseline model in case of highly imbalanced Heart dataset giving an accuracy and f1 score of 0.91 and 0.95 respectively as compared to base model giving accuracy scores of 0.87 and 0.92 respectively. To further strengthen our claim, permutation test performed gave p value 0.05 and 0.009 for accuracy and f1 score respectively. P value for f1 score ensures that more than 99 percent of the times our model outperforms the base model. Table 2

4.3 Mushrooms Dataset Results

In case of Mushroom dataset, baseline model gives perfect accuracy and f1 scores of 1 in case of balanced dataset and all imbalanced surrogates therefore, outperforming the proposed model. Table 1

5 Discussion

After discussing individual results for every dataset and its surrogates, its inferred that in case of heart and gender datasets, proposed model outperformed the baseline model when data imbalance was high whereas in case of mushrooms dataset, baseline model gave perfect scores of 1.0 for balanced data as well as all surrogates. This may be due to the baseline model failing to capture the imbalance ratio overlooking the minority class as it is accuracy driven model. The proposed model gives best results when the data imbalance is very high and these results start getting worse as the ratio of balance increases. Data imbalance and metric scores are directly proportional in case of the proposed model. Moreover, it can be observed that in case of high imbalance, the baseline model only outperformed our proposed model for the dataset having very high dimensionality. i.e. Mushroom Dataset. One reason for that could be formation of noisy clusters as features are used to find the relation between the data instances during clustering and high features vector could result in unnecessary clusters. The reason for our proposed model outperforming the base model on high imbalance could be that there is a high chance that some clusters only contain instances of a single class therefore unseen instances near to that cluster are directly assigned that class without any chance of mistake, whereas if a cluster contains examples from both classes, the class of unseen instance is predicted using the trained Random Forest Classifier which increases

the chances of error. One shortcoming of our methodology is that data was sub-sampled initially to create surrogates therefore may have resulted in losing the important instances affecting the training process.

6 Conclusions

The proposed model has advantage over the baseline model as it uses a blend of supervised and unsupervised learning ensuring that minority class is not overlooked by only training clusters having data from both classes. In addition, using multiple data sets ensures that the results produced are realistic purely based on data imbalance as we saw that in case of Mushrooms data set our model didn't perform well on the highly imbalanced data set using proposed model but it was due to the feature vector size and not the data imbalance. In future, we can study the effect of dimensionality and other factors on the results along with the data imbalance. Also for future, instead of sub-sampling, we can try other sampling techniques such as over-sampling to create the data imbalance in order to avoid losing the important data. Moreover, instead of using the Random Forest Classifier, other classification algorithms such as Support Vector Classifier, Decision Tree Classifier, etc can be used as a baseline model for comparison with the proposed model. Also, proposed model can be altered likewise using Support Vector Classifier or Decision Tree classifier instead of Random Forest Classifier for training the clusters having data from more than one class. Lastly, other metrics such as cohen's kappa score can be used in order to compare the results of two models.

References

[1] P. Bholowalia and A. Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9), 2014.

[2] V. Ganganwar. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42–47, 2012.

[3] R. Ghorbani and R. Ghousi. Comparing different resampling methods in predicting students' performance using machine learning techniques. *IEEE Access*, 8:67899–67911, 2020.

[4] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[5] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, et al. Handling imbalanced datasets: A review. *GESTS international transactions on computer science and engineering*, 30(1):25–36, 2006.

[6] A. Y.-c. Liu. *The effect of oversampling and undersampling on classifying imbalanced text datasets*. PhD thesis, Citeseer, 2004.

[7] J. Ludbrook. Advantages of permutation (randomization) tests in clinical and experimental pharmacology and physiology. *Clinical and experimental pharmacology and physiology*, 21(9):673–686, 1994.

[8] K. Matsushima. The silhouette method. In *Introduction to Computer Holography*, pages 281–308. Springer, 2020.

[9] C. Veni. On the classification of imbalanced data sets. 07 2018.