Automatic detection of atrial fibrillation episodes

University of Twente, Data Science
Primary Topic: DM, Secondary Topic: TS
Course: 2019-1B – Group: 58 – Submission Date: 2019-02-02

Nils Rublein, s1864432
University of Twente
n.rublein@student.utwente.nl

Vera Dierx, s1444271
University of Twente
v.m.l.dierx@student.utwente.nl

ABSTRACT

Automatic detection of Atrial Fibrillation (AF) is important since manual detection can be hard and time consuming. AF can cause serious complications when missed. Therefore the question 'To what extend can one automatically detect episodes of AF?' is of importance. AF can be detected by irregularities in RR-intervals from ECG. A data-set based on this principle is used, from which additional features are extracted and selected. With these feature six models were tested. The random forest tree performed the best with an accuracy, recall, precision and F1, of respectively 0.962, 0.934, 0.913, 0.924. This is a relatively good result, however future recommendations to test more ECG characteristics, features, parameters and models are made. The random forest tree performed the best with an accuracy, recall, precision and F1, of respectively 0.962, 0.934, 0.913, 0.924. This is a good result, however future recommendations to test more ECG characteristics, features, parameters and models are made.

KEYWORDS

Atrial Fibrillation (AF), RR-Intervals, ECG, feature extraction, classification algorithms

1 INTRODUCTION

Atrial fibrillation (AF) is a common complication occurring postoperatively from cardiac surgery. For patients undergoing coronary artery bypass surgery the incidence of AF occurring afterwards is 30%. This incidence is only expected to rise in the future, given that patients undergoing surgery are getting older with age being a major risk factor for AF[3]. AF is associated with a higher incidence of postoperative complications such as thromboembolic disease and stroke, haemodynamic compromise and ventricular dysrhythmias[3, 4]. Therefore also increasing the length of hospital stay with approximately 5 days and subsequently increasing the costs with about $ 10.000 in the United States[3, 4]. Since there are several treatment options that offer benefits but also can have serious risks it is important to accurately detect AF[9, 16]. AF is often asymptomatic and occurs periodically which makes it hard to be detected. The ECG monitoring necessary for this detection is done manually which is time-consuming and therefore also costly. For those reasons automatic detection of AF has major benefits[16]. This is leading to the following research question for this paper: To what extend can one automatically detect episodes of AF?

In order to answer this question it is important to know what characterizes AF. It is dened as a period of at least 30 seconds in which an irregular ventricular rate and absent P-waves[12]. Therefore it can be characterized by two ECG features, namely by noticing irregularities in the RR intervals and the absence of P-waves[16]. Since the R-peaks are a prominent feature in an ECG and can be easily detected, they will be used to answer the research question. The provided data will be evaluated based on this ECG feature. Different features and models are selected to answer the research question, as is described in section 3. The results will be evaluated in section 4. Eventually the performed experiments will be discussed and compared with literature in section 5. This will lead to an answer of the research question in the conclusion section (section 6) of this paper. Additional figures can be found in the appendixes.

2 BACKGROUND AND RELATED WORK

2.1 Balancing data

Data is called imbalanced if it contains many more samples in a class then in other classes. This causes problems if it occurs in training data, since the accuracy on the majority class can be good while being poor in the minority classes. This is due to the influence the majority class has on the training criteria, since most classification methods try to minimize the overall error rate. These methods treat every miss-classification error equally. In a lot of applications this assumption is incorrect which is also the case for the classification of AF. Missing an AF episode is more serious then wrongly classifying non-AF as AF. Since imbalanced data tends to have a bias towards the majority class and no AF occurs more frequently than AF in ECG data, this will cause a problem. Solutions for this can be at data and algorithmic levels. For the data level this is mostly done by re-sampling methods and for algorithmic level solutions include to discriminate between the costs of different classes. The literature does not point to one technique being superior to another in general. Since this seems to be dependent on the exact data-set and predictive method used, however re-sampling seems to be a commonly used technique.[2, 13]. In section 3.1 four different re-sampling methods will be discussed.

2.2 RMSSD

The root mean square of successive differences (RMSSD) is a time domain feature to assess variability in heart rate using RR intervals. AF causes a higher variability than a regular heart rhythm, therefore this can be a useful feature in the automatic detection of it. It is calculated by equation 1, where $a(j)$ represents a given segment of RR intervals of a length $l(ms)$. [10, 26]

$$RMSSD = \sqrt{\frac{1}{(l-1} \cdot \sum_{j=1}^{l-1} (a(j+1) - a(j))^2} \qquad (1)$$

2.3 Feature selection

Feature selection is a method of selecting a subset of relevant features and is used to reduce over-fitting and training time, improve accuracy and lead to a better understanding of the features as well as their relationship to the response variables [5, 14]. Typical feature selection methods include wrappers, filters, embedded methods. Wrappers methods utilize predictive models to evaluate feature subsets, where the error rate is used to assign a score to each feature. Wrapper methods are computationally intensive but generally select the best features for a particular data set [18]. Filter methods select subsets of features based on proxy measures instead of error rates. Filter methods are generally less computationally intensive than wrapper methods and are not configured to a specific type model, which makes them more universal but also gives them a reduced prediction performance in comparison to wrapper methods [30]. Embedded methods combine wrapper methods and filters by performing feature selection and classification simultaneously. Embedded methods gravitate towards the middle of wrapper and filter methods in terms of computational time and predictive performance[18].

2.4 Related Work

There is already research done on different methods for detecting AF episodes. The article of Larburu[16] compares different algorithms tested in earlier studies but in this case on the same database. The conclusion from this study is that a choice of algorithm should be made on the application in which it is used. Their results vary from a recall of 91 % to 98%. When measuring in real time, a small window length is preferred. Here the framework combination method did well with a signal length of 10 seconds, still resulting in a high sensitivity. Other algorithms needed a length of about 1 minute to perform well.

The research paper of Dash [9] considers different features using logistic regression with the combination of a Turning Point Ratio, RMSSD, the shannon Entropy and the RR-interval time. This was tested on two databases with a recall of 90% and 94 %..

Another article by Bruser [6] compares different models for detecting AF. Although here a BCG system is used instead of a standard ECG, the R peaks can still be detected and used for processing. The article uses 17 different features to which feature selection is implemented. F1 scores for a linear regression, Support vector machine (SVM), Nave Bayes and random forest tree are, respectively, 0.749, 0.881, 0.890 and 0.936, from which they conclude the random forest tree would be the most suitable algorithm to use.

The last report to be discussed here is from Martis [20]. This report summarizes the results other studies testing automatic detection of AF possibilities. The results from these studies vary quite a lot from a recall of about 70 % to even a recall of 100 % from its own experiment. For the study a mixed model was used, the paper did not elaborate on the methods used in the compared results.

3 APPROACH

The data for this report is already processed as is described in section 4.1. First, the data will be split into a training and a testing set with a ratio of respectively 75% and 25%. Then the data will be balanced and several features will be extracted and selected. Finally, various classification models will be generated and compared. All steps are being made using Python.

3.1 Re-sampling

As shown in section 4.1, the data needs balancing. The methods that will be evaluated for this paper are under-sampling, over-sampling, over-sampling using Synthetic Minority Oversampling Technique (SMOTE) and a combination of oversampling (SMOTE) and under-sampling via Tomek Link Removal (TLR).

Under-sampling is a re-sampling strategy that only uses a subset of the majority class [17]. The advantage of under-sampling is that the training process becomes faster due to the smaller sample size, however, a disadvantage is that potentially useful information contained in the excluded data is neglected. Oversampling is a widely used method that copies existing data points randomly and adds them to the training set until a full balance is reached [11]. A disadvantage of oversampling is that it can over-fit the data. SMOTE creates synthetic elements for the minority class, based on those that already exist, then randomly picks a point from this minority class and computes the k-nearest neighbors for this point [8, 23]. Finally, the synthetic points are then added between the chosen point and its neighbors. Tomek links are pairs of data points that belong to different classes and are each other's nearest neighbors [23, 28]. Removing Tomek links creates more differences between the two classes and is a variation of under-sampling. First under-sampling the synthetically created samples by SMOTE with TLR will remove unwanted noise and improve the quality of the data [23, 29].

To evaluate which method would be preferable, a simple logistic regression, ROC curves and precision-recall plots will be compared for all four methods.

3.2 Additional features and feature selection

The pre-processed data already has 30 features, which are the bins made in the pre-processing as explained in section 4.1. For each period some additional features are created, namely:

(1) Mean
(2) Standard Deviation
(3) Maximum
(4) Minimum
(5) Median Absolute Value
(6) RMSSD

A selection from the extracted and the original features is made using a random forest, recursive feature elimination and stability selection, where the recursive feature elimination and stability selection are both wrapper methods and random forest is an embedded method [1].

Recursive feature elimination is based on the concept of repeatedly building a model and choosing either the best or worst performing feature. This feature is then excluded for the next iteration of the algorithm until the list of features is exhausted. Finally, features are ranked based on the best performing. Stability selection applies feature selection algorithms on different subsets of data with different subsets of features. After repeating this process multiple times, the results are being accumulated by inspecting how many times a feature has been considered as relevant or irrelevant. Highly relevant features will be picked close to 100% whereas irrelevant features will be picked close to 0%. Random forests have an embedded feature selection method that is based on the mean decrease of impurity. Impurity is the measure which chooses the (locally) optimal condition for splitting the nodes based on the features in the decision trees of the random forest. Computing the decrease in the weighted impurity for each tree is then used to rank the features.

After executing these three methods, the overlap of relevant features will be considered for further analysis of the data.

3.3 Classification algorithms

In order to answer the research question different classification algorithms are tested and compared. The choice was made to use some basic data mining and time series algorithms in combination with algorithms found in existing literature.

(1) Decision Tree
(2) Random Forest Tree
(3) Linear Regression
(4) Gaussian Naïve Bayes Classifier (GNB)
(5) K-Nearest Neighbours (KNN) - Dynamic Time Warping (DTW)
(6) Support Vector Machine (SVM)

Decision trees break down data sets into smaller and smaller subsets based on the entropy of the features where the best predictor in a decision tree is called root node and is at the top of the tree [7].

Random forest trees combine multiple decision trees. They pick certain subsets of the data set and build a decision tree for each of these nodes. The process is repeated N times, where the node with the highest number of votes gets selected. For this data set, after testing multiple iterations, N has been set 100.

For the Linear regression a logistic regression function (2) is used. The solver for training the data was used at default setting with the exception of the solver, which is based on the F1 score[15].

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}} \quad (2) \qquad p(y|X) = \frac{P(X|y)P(y)}{P(X)} \quad (3)$$

The GNB classifier based on equation 3, $with\ X = (x_1, x_2, ..., x_n)$, was used in default settings based on the assumption that $P(X_i|y)$ follows a normal distribution [27].

The KNN algorithm is a classification model that stores all training data and classifies new data based on a similarity measure [24]. A data point is classified via majority voting by its neighbors, where it is assigned to the class most common amongst its KNN calculated by a distance function. Typical distance measures include Euclidean, Manhattan or Minkowski distance. For this data set, DTW has been chosen as distance measure with K being equal to 1 based on the meta analysis by Mitsa [21]. DTW computes the best alignment between two sequences and is useful for classifying sequences that have different frequencies or have different phase shifts, which makes it an effective method to classify time series data [19].

The SVM uses a separating hyper-plane. The method allows for transformations that are called kernels in order to draw the hyper-plane in different dimensions allowing different distributions of data being processed effectively. For this data a polynomial kernel is used. Since there is an overlap in the distribution of classes, a soft-margin SVM is chosen meaning two tuning parameters are implemented. The first parameter is the regularization value C which determines to what extend misclassification should be avoided in training. Secondly, the gamma parameter determines how far the influence of a single training examples reaches. This means for a high gamma value that training points close to the decision boundary carry more weight than points far away and vice versa for a low gamma value [14, 25]. To determine the optimal values for these two parameters a grid-search function with a five time cross-validation is used.

3.4 Evaluation Metrics

The models will be evaluated after the trained model is tested in the test set. For this evaluation four performance criteria are calculated, as shown in equations 4, 5, 6 and 7, with TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative. Thereby ROC and precision-recall curves are made for each model and each data set to visualize the above mentioned metrics.

$$Accuracy = \frac{TP+TN}{all\ samples} \quad (4) \qquad Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (5) \qquad F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (7)$$

4 EXPERIMENTS

4.1 Data

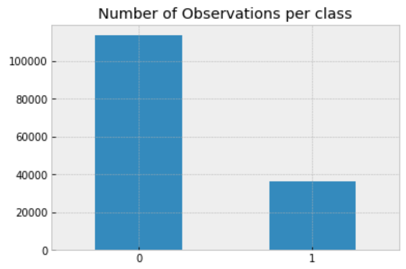
The data used is preprocessed ECG data from a 12 lead ECG recording, with annotated RR peaks. The preprocessed data consists of 31 columns and 15000 rows. Column 1 to 30 are bins created from RR intervals of 200 ms up to 1700 ms, with steps of 50 ms. The last column is the classification where 0 mean no AF was reported and 1 means AF was reported by a physician. The rows represent periods of half a minute recorded from different patients. Data that had nonphysical or missing values was excluded. Looking at the classes the data set is unbalanced with 113463 non AF and 36537 AF periods as visualized in figure 1a, so only 32.2% are AF episodes.

4.2 Re-sampling

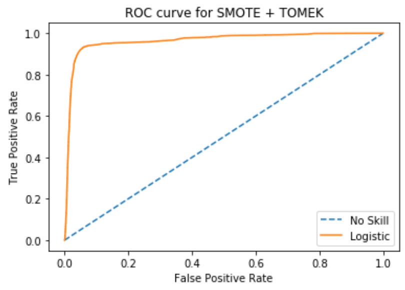
First a test classifier has been implemented that selects no AF by default as baseline to compare with the re-sampling methods. Implementing under-sampling, oversampling, SMOTE and SMOTE followed by TML improves accuracy, recall and the F1 score but lowers the precision, as depicted in table 1. Comparing the different re-sampling methods, it can be seen that all have the same accuracy and almost the same F1 score. Based on these results, it has been decided to use SMOTE and TML to re-sample the training data; a ROC curve and precision-recall curve can be found below in figures 1b and 1c respectively. ROC and precision-recall curves for the other re-sampling methods can be found in appendix A.

Table 1: Re-sampling results using logistic regression.

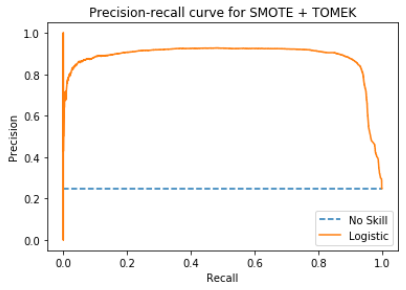
Re-sampling	Accuracy	Recall	Precision	F1
None	0.933	0.807	0.911	0.856
Under-sampling	0.944	0.892	0.883	0.887
Over-sampling	0.944	0.898	0.880	0.889
SMOTE	0.944	0.898	0.880	0.889
SMOTE + TLR	0.944	0.898	0.880	0.889



(a) Visualisation of the number of observations in each class.



(b) ROC curve for SMOTE followed TML.



(c) Precision-recall curve for SMOTE followed TML

Figure 1: Unbalanced data (a) and the results of balancing by Re-sampling (b,c))

4.3 Feature Selection

The embedded feature selection of the random forest classifies the last ten data bins as well as the mean as irrelevant by giving it a zero value. A graph visualising this is shown in appendix B. The second feature selection algorithm that has been implemented was the recursive feature elimination. After executing the algorithm and ranking each feature, data bins 1 and 17 to 30 as well as the mean have been eliminated. Lastly, the stability selection selected all extracted features 100% of the times while it gave scores less than 100% for data bins 8 to 11 and 23 to 28, where the minimum percentage was 49% for data bin 9. This graph is also shown in appendix B.

4.4 Classification

Based on the results of the feature selection, the mean and the last 10 data bins have been excluded. To test whether the feature extraction and selection improves the predictive performance of the models, three data sets have been created: First, a data set that includes the original and the all of the extracted features, labeled as 'all'. Second, a data set that only includes the selected features, labeled as 'selected'. Third, a data set that excludes the extracted features and only contains the original features, labeled as 'original'. The results of all classification models for all three data sets can be found below in table 2. ROC and precision-recall curves for all data sets and models can be found in appendix C.

Table 2: Metric results for classification models

Model	Data-features	Accuracy	Recall	Precision	F1
Decision Tree	Selected	0.944	0.878	0.893	0.885
	All	0.944	0.878	0.893	0.885
	Original	0.942	0.872	0.892	0.882
Logistic Regression	Selected	0.950	0.901	0.895	0.898
	All	0.952	0.907	0.898	0.903
	Original	0.945	0.899	0.880	0.889
GNB	Selected	0.902	0.888	0.757	0.818
	All	0.689	0.925	0.438	0.594
	Original	0.612	0.922	0.381	0.539
KNN with DTW	Selected	0.899	0.830	0.815	0.822
	All	0.896	0.868	0.786	0.825
	Original	0.880	0.849	0.756	0.800
Random Forest	Selected	0.960	0.932	0.909	0.920
	All	0.961	0.932	0.910	0.921
	Original	0.962	0.934	0.913	0.924
SVM	Selected	0.921	0.912	0.797	0.851
	All	0.921	0.913	0.798	0.851
	Original	0.912	0.927	0.764	0.838

5 DISCUSSION

First, balancing the training data will be discussed. In literature when speaking of unbalanced data, ratios of 99 to 1 percent or smaller are often mentioned in this context [22]. Compared to this the data in this report would not be categorized as unbalanced. So one can say balancing the data is unnecessary, only causes loss of data and taking up extra time. However when testing the balanced data it gives slightly better results, and since the eventual method is a combination of over- and under-sampling, the amount of lost data is minimized. The time lost will not cause a problem since balancing is only necessary for the training data.

To interpret the result the practical meaning of the evaluation metrics has to be taken into account. The accuracy is an use-full measure since it gives information about not only the true positives but also about the true negatives, which is interesting since a high amount of true negatives is preferable in medical cases as this one. Looking at the results stated above it can be seen that the accuracy is in most cases the highest performing metric. However one should take into account that the test data is still unbalanced which could cause a bias towards the true negatives. This could be partly solved with applying a standard error to the accuracy, which would be a recommendation when further research is done. Looking at other metrics also solves this problem. Especially the recall is of importance, since this is a measure of the number of AF episodes that are wrongly classified as non AF, causing a health risk for the patient. From this perspective the GNB for all and the original features would give one of the top results. However looking further at the precision, the performance in this case is low. In practice this means that there is a low amount of missed AF episodes but a lot of patients get false positives meaning that a physician has to check on them. This is time consuming and therefore costly, counteracting the advantage of automatic detection being less time consuming and expensive then manual ECG monitoring. The F1 score combines recall and precision, making it a very interesting metric for AF classification. Judging on the F1 score the two best performing models are the logistic regression and random forest tree. Combining this with the accuracy score as a secondary measure, the random forest tree model performs the best. This is in line with some of the research mentioned in 2.4 where also a random forest tree performed the best. The results there where a bit better, which could be due to the use of more extensive features. However the obtained results seem to be in line with a lot of the previous research.

When inspecting the results of the three separate data-sets in terms of the used metrics, there seems to be no significant differences between with the exception for the GNB, as this classifier treats all features with the same importance, it is sensitive to features that result in a lot of false classified results. Also a lot of rejected features are not expected to have a normal distribution, which is assumed in this model. The insignificant differences between the data-sets for the other models may be caused as a consequence of wrapper methods that depend on the specific classification model (e.g. SVM, random forest) that is used to rank the features and should therefore not be generalized for other kinds of models. When testing several models, either 1) a specific wrapper or embedded method for each model should be used, 2) only filter methods should be used that can be generalized for several kinds of models.

In order to fully asses to what extent AF can be automatically detected one would also need to validate the pre-processing of the data. The models are tested on pre-processed data assuming this data is reliable. However when R-peaks are wrongly detected this could cause a change in the frequency an interval occurs in a bin. Therefore also changing the outcome if data is classified as AF or not. This validation is currently not done. Another drawback of the used data-set is the limitations to RR-intervals. Another ECG characteristic for detection of AF is the absence of a P-wave. Implementing this would come with more difficulties since automatic detection of P-waves is more intricate. However research has already been done using more ECG characteristics, so there are possibilities to implement this. Unfortunately with the available data there was no possibility to look at this option.

A more general drawback in this experiment is the selection of parameters for the used models. Especially for the decision tree and random forest tree the set parameters were chosen by manually performing some iterations and choosing the best option. For these models a grid search as implemented with the SVM would be an improvement in reliability of the models and possibly in the performance. For the other models mostly default setting are used. Tuning the parameters for the models could improve the results and would be recommended to implement if further research is done.

A fourth drawback is that the SVM and KNN models are trained on only a proportion of the data. For the KNN is this was also done for the test data. This was due to lack of computational power and time, since these models take a long time to train. However the results on a small part of the data-set are already quite positive.

In order to use automatic detection of AF in a hospital setting, real time detection of AF is a prerequisite. This is not expected to cause a problem, however this is not tested in this research. For real time detection, speed is important, but only the training of the models is time consuming, where the classification is often quickly executed. Further given an AF episode, immediate action is not required as would be when detecting for instance cardiac arrest. In practice a physician would want to know if and how often in time AF occurred. Therefore, a model that takes some time and has some error margin could still be implemented real time.

6 CONCLUSIONS

This study was set out to explore possibilities of automatically detecting the occurrence of AF episodes. Since AF can cause serious complications, detection is important. However AF episodes can be easily missed and the manual ECG monitoring necessary to detect AF is time-consuming and therefore costly. This makes the exploration to the performance of automatic AF detection interesting leading to the research question 'to what extent can one automatically detect episodes of AF'?

This question was answered by using balancing techniques on the provided ECG data. After feature extraction and selection, six models were tested on their classification abilities. The majority of these models performed well, with the random forest tree with original features having the highest scores in accuracy, recall, precision and F1, being respectively 0.962, 0.934, 0.913, 0.924. Although the feature selection did not influence the results much, they are in the same line as other research done on this topic. Thereby real-time implementation seems to be realistic. This leads to answering the research question that AF can already be predicted to a reasonable good extend, but better performance seems to be possible.

Important to mention is that this study limits itself to evaluation of classification models on preprocessed data that is assumed to be reliable. Research to the reliability of the prepossessing would be recommended, thereby testing models on different data would test its reliability more extensively. Secondly, the search for parameters was limited and could be expanded. Thirdly, for the SVM and KNN a test over the whole data-set would be recommended when having enough computational power or time, if not re-coding in order to decrease the training time could be considered. Finally looking if more complex methods increase the ability to correctly predict AF,would be recommended. This would include using more complex ECG characteristics as the P-wave as well as using other classification methods. The same goes for the used features, in literature one comes across a lot of different features used for AF detection, for example the entropy. Before implementing automatic detection of AF in a hospital setting it would be recommended to look at these proposed extension.

However the random forest tree classification model as used in this experiment can already satisfactorily detect episodes of AF based on the used data-set.

REFERENCES

[1] A. Saabas. 2014. Selecting good features – Part IV: stability selection, RFE and everything side by side. https://blog.datadive.net/selecting-good-features-part-iv-stability-selection-rfe-and-everything-side-by-side/ [Online; accessed 28-January-2020].

[2] Aida Ali and et al. 2015. Classification with class imbalance problem: A review. 7 (01 2015), 176–204.

[3] Sary F. Aranki and et al. 1996. Predictors of Atrial Fibrillation After Coronary Artery Surgery. *Circulation* 94, 3 (1996), 390–397. https://doi.org/10.1161/01.CIR.94.3.390

[4] Awad. 2010. Atrial fibrillation post cardiac surgery trends toward management. *Heart Views* 11, 2 (2010), 57–63. https://doi.org/10.4103/1995-705X.73212

[5] M. Bermingham and et al. 2015. Application of high-dimensional feature selection: evaluation for genomic prediction in man. *Sci Rep* 5 (2015). https://doi.org/:10.1038/srep10312

[6] C. Bruser, J. Diesel, M. D. H. Zink, S. Winter, P. Schauerte, and S. Leonhardt. 2013. Automatic Detection of Atrial Fibrillation in Cardiac Vibration Signals. *IEEE Journal of Biomedical and Health Informatics* 17, 1 (Jan 2013), 162–171. https://doi.org/10.1109/TITB.2012.2225067

[7] C. Sehra. 2018. Decision Trees Explained Easily. https://medium.com/@chiragsehra42/decision-trees-explained-easily-28f23241248 [Online; accessed 28-January-2020].

[8] N.V. Chawla and et al. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357. https://www.scopus.com/inward/record.uri?eid=2-s2.0-0346586663&partnerID=40&md5=dfb419b8460388447758f9c7f8c2a103

[9] S. Dash and et al. 2009. Automatic Real Time Detection of Atrial Fibrillation. *Annals of Biomedical Engineering* 37, 9 (01 Sep 2009), 1701–1709.

[10] S. Dash and et al. 2009. Automatic Real Time Detection of Atrial Fibrillation. *Annals of Biomedical Engineering* 37, 9 (01 Sep 2009), 1701–1709. https://doi.org/10.1007/s10439-009-9740-z

[11] A. Estabrooks and et al. 2004. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence* 20, 1 (2004), 18–36. https://doi.org/10.1111/j.0824-7935.2004.t01-1-00228.x cited By 497.

[12] V. Fuster and et al. 2001. ACC/AHA/ESC Guidelines for the Management of Patients With Atrial Fibrillation: Executive Summary A Report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines and the European Society of Cardiology Committee for Practice Guidelines and Policy Conferences (Committee to Develop Guidelines for the Management of Patients With Atrial Fibrillation). *Circulation* 104, 17 (2001), 2118–2150. https://doi.org/10.1161/circ.104.17.2118 arXiv:https://www.ahajournals.org/doi/pdf/10.1161/circ.104.17.2118

[13] Vaishali Ganganwar. 2012. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering* 2 (01 2012), 42–47.

[14] G. James and et al. [n. d.]. *An Introduction to Statistical Learning: with Applications in R.* Chapter 9: Support Vector Machines.

[15] G. James and et al. [n. d.]. *An Introduction to Statistical Learning: with Applications in R.* Chapter 4: Classification.

[16] N. Larburu and et al. 2011. Comparative study of algorithms for Atrial Fibrillation detection. (Sep. 2011), 265–268.

[17] X. Liu and et al. 2009. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39, 2 (April 2009), 539–550. https://doi.org/10.1109/TSMCB.2008.2007853

[18] M. Lu. 2019. Embedded feature selection accounting for unknown data heterogeneity. *Expert Systems with Applications* 119 (2019), 350 – 361. https://doi.org/10.1016/j.eswa.2018.11.006

[19] M. Regan. 2017. Timeseries Classification: KNN DTW. https://nbviewer.jupyter.org/github/markdregan/K-Nearest-Neighbors-with-Dynamic-Time-Warping/blob/master/K_Nearest_Neighbor_Dynamic_Time_Warping.ipynb [Online; accessed 29-January-2020].

[20] Roshan Joy Martis, U.Rajendra Acharya, Hari Prasad, Chua Kuang Chua, and Choo Min Lim. 2013. Automated detection of atrial fibrillation using Bayesian paradigm. *Knowledge-Based Systems* 54 (2013), 269 – 275. https://doi.org/10.1016/j.knosys.2013.09.016

[21] T. Mitsa. [n. d.]. *Temporal Data Mining.* Chapter 3: Temporal Classification.

[22] Ronaldo Prati and et al. 2009. Data mining with imbalanced class distributions: Concepts and methods. *Paper presented at the IICAI*, 359–376.

[23] R. Alencar. 2018. Resampling strategies for imbalanced datasets. https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets [Online; accessed 28-January-2020].

[24] s. Sayad. 2018. K Nearest Neighbors - Classification. https://www.saedsayad.com/k_nearest_neighbors.htm [Online; accessed 29-January-2020].

[25] Savan Patel. 2017. Chapter 2 : SVM (Support Vector Machine) — Theory. https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72 [Online; accessed 23-January-2020].

[26] Fred Shaffer and J. P. Ginsberg. 2017. An Overview of Heart Rate Variability Metrics and Norms. *Frontiers in Public Health* 5 (2017), 258. https://doi.org/10.3389/fpubh.2017.00258

[27] Daniele Soria and et all. 2011. A 'non-parametric' version of the naive Bayes classifier. *Knowledge-Based Systems* 24, 6 (2011), 775 – 784. https://doi.org/10.1016/j.knosys.2011.02.014

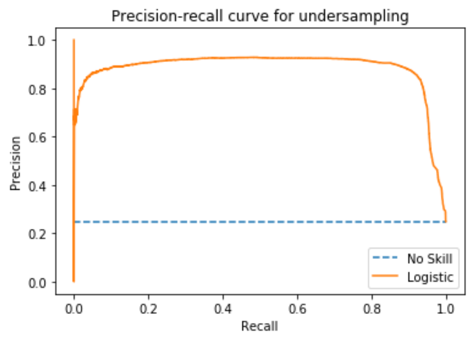
[28] Ivan Tomek. 1976. Two Modifications of CNN.

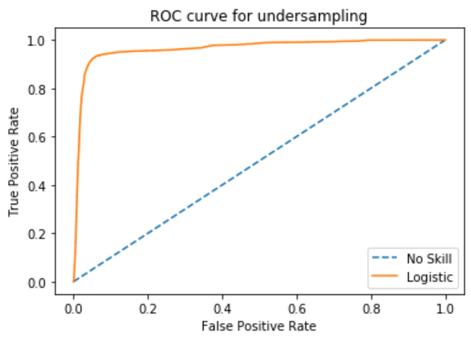
[29] M. Zeng and et al. 2016. Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data. In *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS).* 225–228. https://doi.org/10.1109/ICOACS.2016.7563084

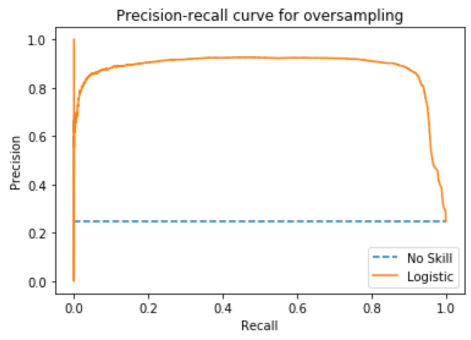
[30] Y. Zhang and et al. 2013. Divergence-based feature selection for separate classes. *Neurocomputing* 101 (2013), 32 – 42. https://doi.org/10.1016/j.neucom.2012.06.036

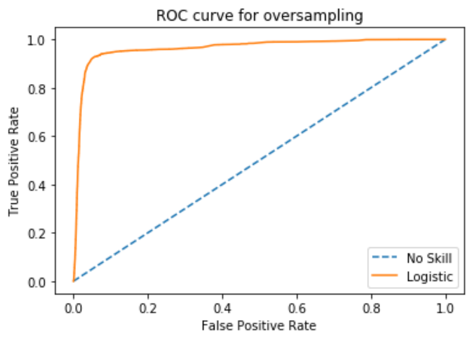
A RE-SAMPLING METHODS

The remaining ROC and precision-recall curves for the re-sampling are represented below.

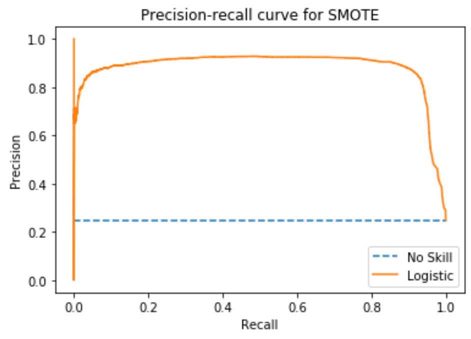


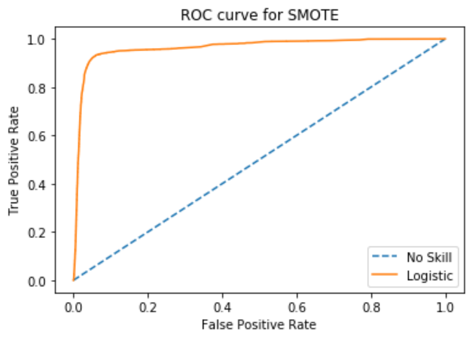


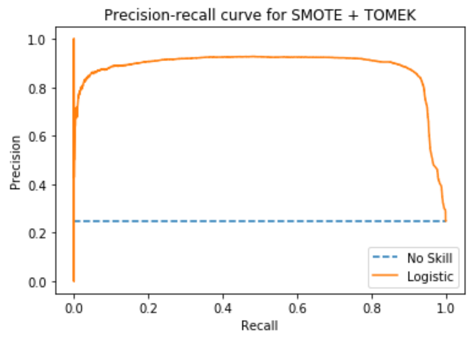




(a) ROC curve for over-sampling.







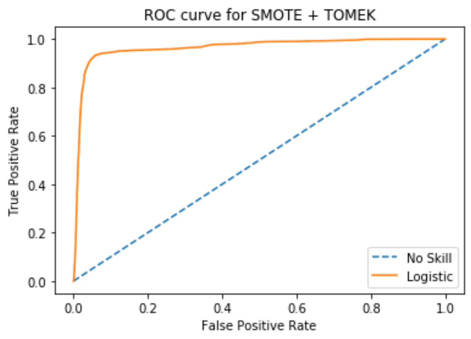


Figure A2: Precision-Recall and ROC Curves for the discarded sampling methods

B FEATURE SELECTION PLOTS

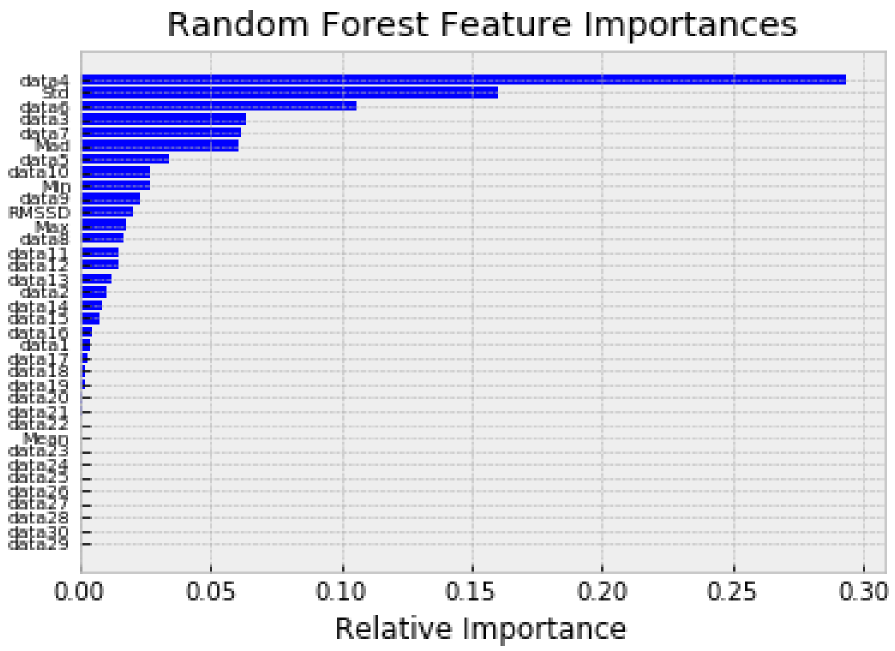


Figure A3: Feature importance of random forest.

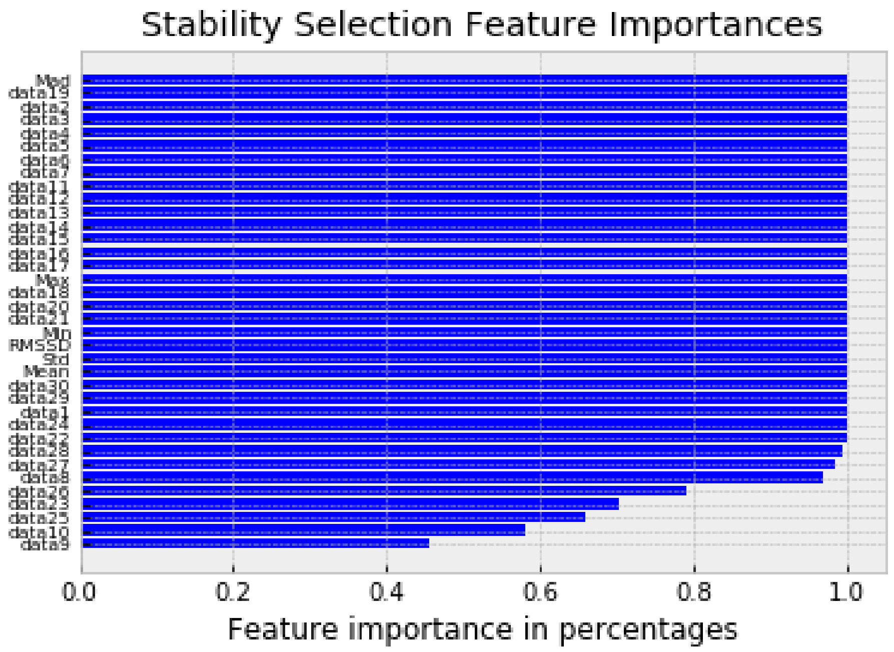
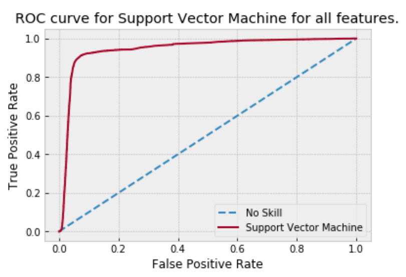


Figure A4: Results of stability selection.

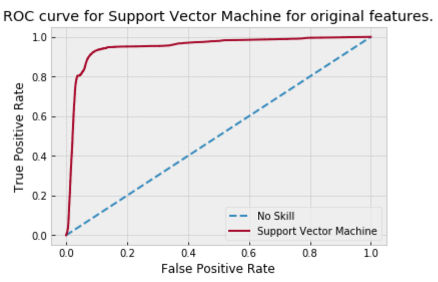
C CLASSIFICATION RESULTS

ROC and precision-recall curves belonging to the results of the different models tested on the differents datasets are presented below.

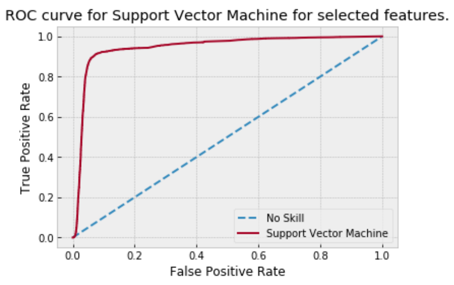
C.1 ROC Curves



(a) All features.

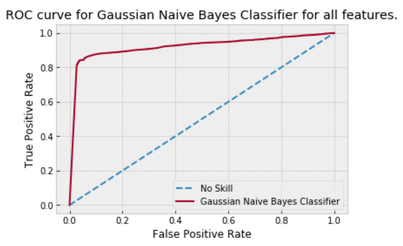


(b) Original features.

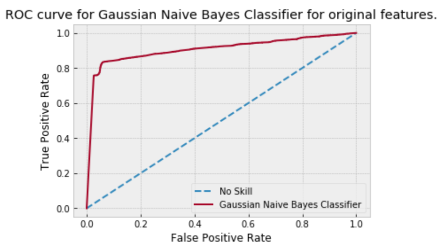


(c) Selected features.

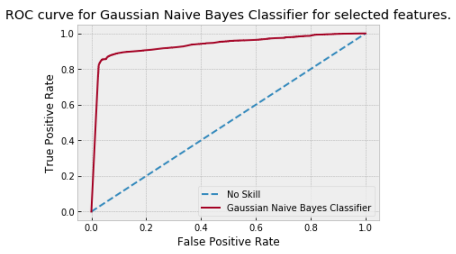
Figure A5: ROC curves for the SVM.



(a) All features.

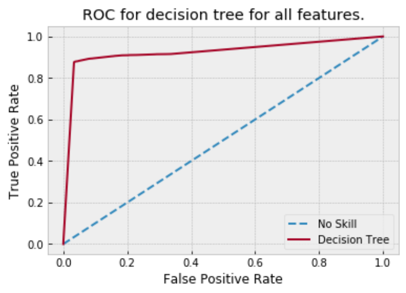


(b) Original features.

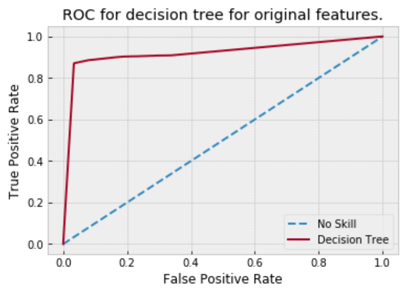


(c) Selected features.

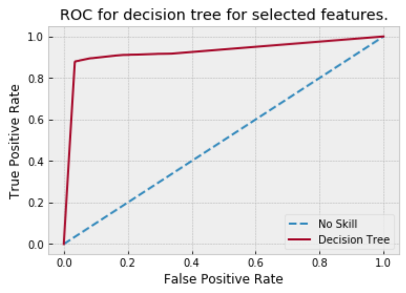
Figure A6: ROC curve for the GNB.



(a) All features.

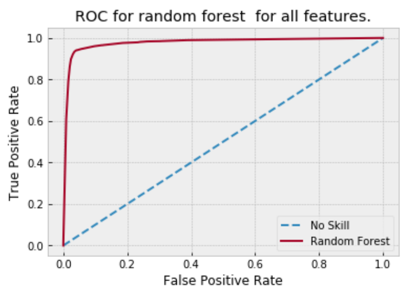


(b) Original features.

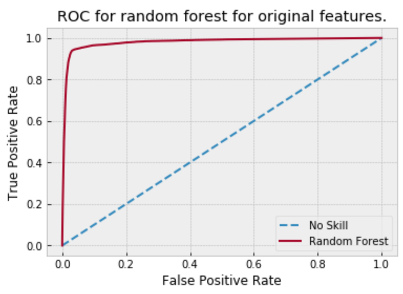


(c) Selected features.

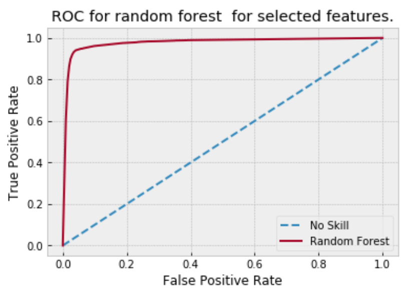
Figure A7: ROC curve for the decision tree.



(a) All features.

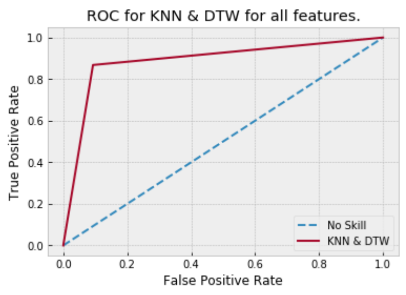


(b) Original features.

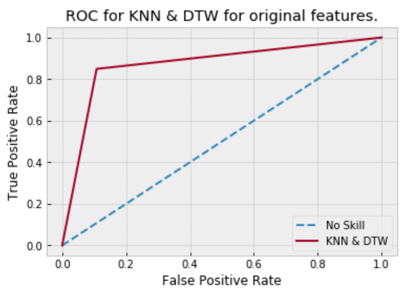


(c) Selected features.

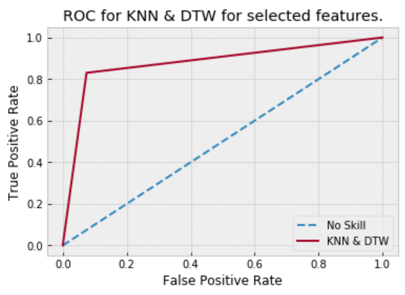
Figure A8: ROC curve for the random forest tree.



(a) All features.

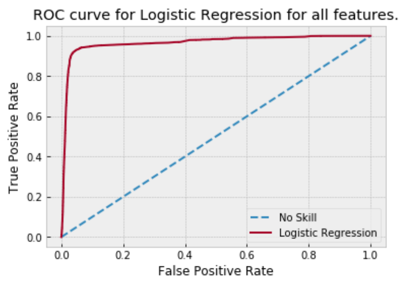


(b) Original features.

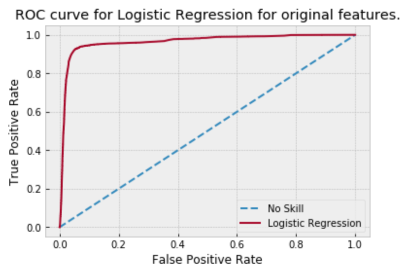


(c) Selected features.

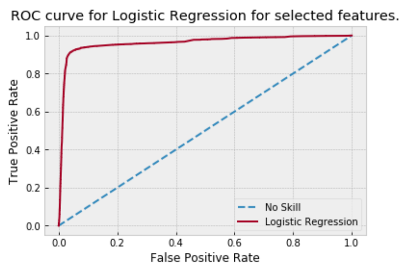
Figure A9: ROC curve for KNN & DTW.



(a) All features.



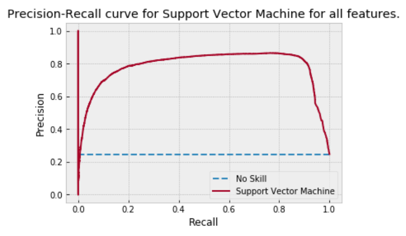
(b) Original features.



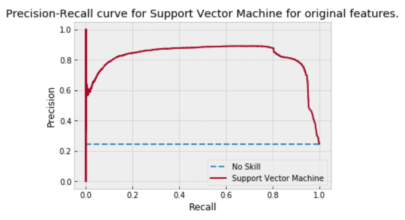
(c) Selected features.

Figure A10: ROC curve for logistic regression.

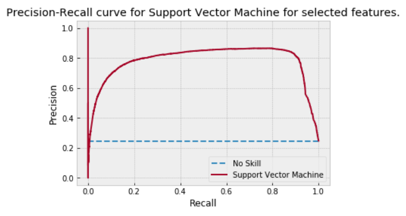
C.2 Precision-recall curves



(a) All features.

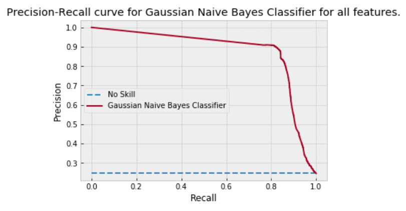


(b) Original features.

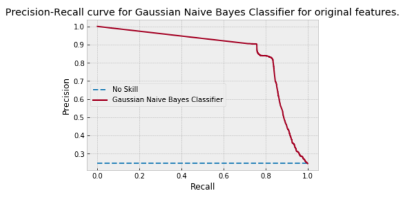


(c) Selected features.

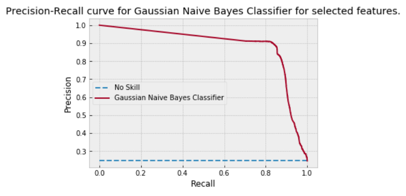
Figure A11: Precision-Recall curve for the SVM.



(a) All features.

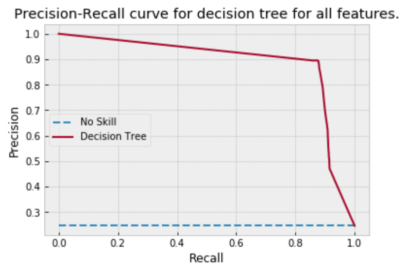


(b) Original features.

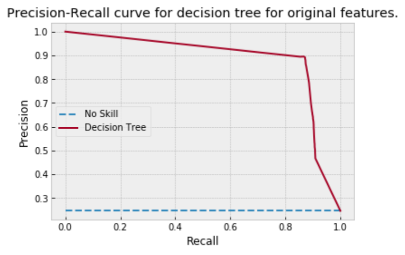


(c) Selected features.

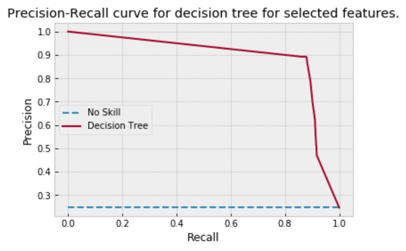
Figure A12: Precision-Recall curve for the GNB.



(a) All features.

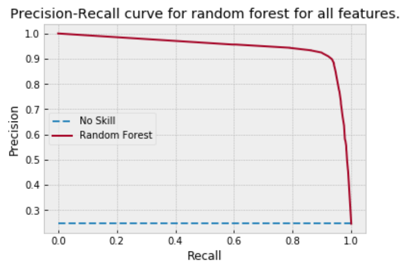


(b) Original features.

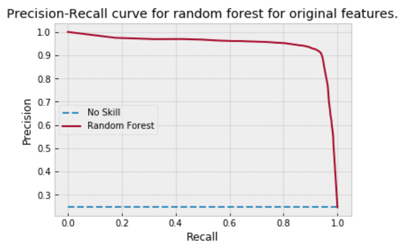


(c) Selected features.

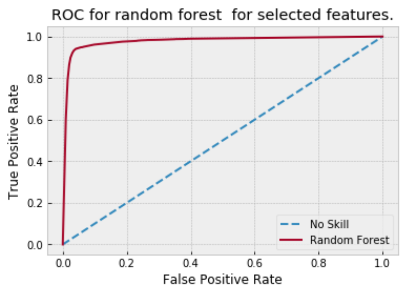
Figure A13: Precision-Recall curve for the decision tree.



(a) All features.

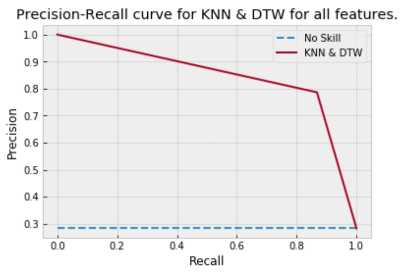


(b) Original features.

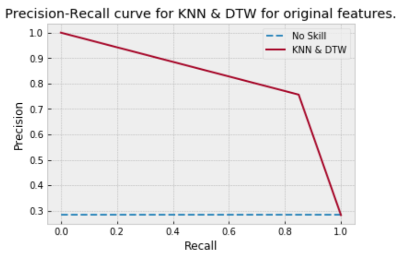


(c) Selected features.

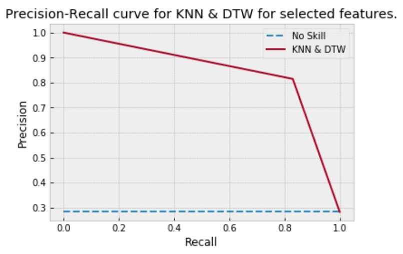
Figure A14: Precision-Recall curve for the random forest tree.



(a) All features.

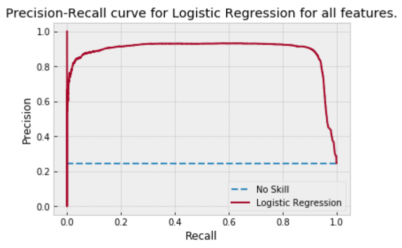


(b) Original features.

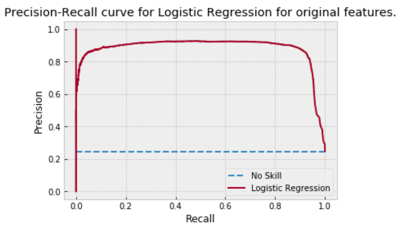


(c) Selected features.

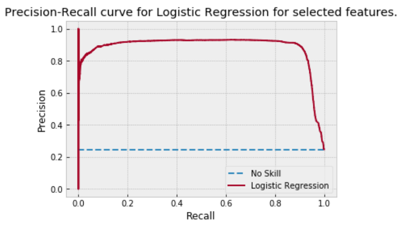
Figure A15: Precision-Recall curve for KNN & DTW.



(a) All features.



(b) Original features.



(c) Selected features.

Figure A16: Precision-Recall curve for logistic regression.