Comparison of Probabilistic Classifiers:
IBM HR Analytics Employee Attrition & Performance

Yizi Huang YiziwinnieH@hotmail.com
Jiayu Qi qijiayu1994@gmail.com

Abstract

Classification is a data mining technique used to predict group label for data points in a given dataset. For binary classification, techniques like k-nearest neighbor, support vector machine and decision tree provide non-probabilistic results such as yes or no. On the other hand, Naive Bayes classification technique applies Bayes' theorem and assumes class conditional dependency, provides probabilities for each class. The paper focuses on the comparisons of the probabilities among different classification techniques; by converting non-probabilistic classifiers to probabilistic classifiers, we are able to evaluate each classifier on sensitivity, specificity, accuracy, AUC, and threshold. Specifically, we conduct our research on the dataset on IBM employee attrition, which is a binary class problem. The class distribution is unbalanced where we apply different preprocessing methods and to compare such as oversampling, undersampling, normalization, feature extraction and feature selection among Naive Bayes, KNN and SVM. After preprocessing, all three classifiers improved the prediction performance over unpreprocessed data. The results indicate that the support vector machine combined with oversampling and normalization achieves the best classification performance. Application of these models has the potential to help reduce employee attrition.

Introduction

In machine learning, a probabilistic classifier is a classifier that is able to predict, given an observation of an input, a probability distribution over a set of classes, rather than only outputting the most likely class that the observation should belong to. However, there are non-probabilistic classifiers that the uncertainty can't be "quantified".

In this research project, we are interested in comparing non-probabilistic classifiers such as k-nearest neighbor (KNN) and Support Vector Machines (SVM) to probabilistic ones and compare with probabilistic classifier Naive Bayes; therefore, to gain the best probabilistic classifier quantifying the uncertainty of the case labeled into the certain class. Moreover, we are interested in exploring the effect of preprocessing steps on the overall performance of classifiers. The dataset we are using is IBM HR Analytics Attrition & Performance. Attrition in human resources refers to the gradual loss of employees over time. The goal is to find the best probabilistic classifier to predict the attrition of valuable employees.

Related Work

The paper written by Jadhav and H.P. in 2016 shows the comparative study of KNN, Naive Bayes and Decision Tree classification techniques. They ran each classifier separately and compare results on accuracy, effectiveness on data size and speed. None of the algorithms can satisfy all constraints and criterions. The limitation is that there are more model evaluation metrics that could be applied such specificity and sensitivity. That is why in our research paper, we will convert the non-probabilistic classifiers to probabilistic ones for comparisons on different metrics. Another research paper that is really similar to our study is *The case of predicting freshman student attrition* (Thammasirion & Monikal, 2014). They compared oversampling and undersampling among different algorithms such as SVM, logistic regression and neural networks. The result shows that oversampling with SVM has the highest overall performance. Nevertheless, they conducted the oversampling after normalizing the data where we want to explore what the effect would be if we normalize before sampling.

In *Support Vector Machines as Probabilistic Model,* the outputs of Linear Support Vector Machine decision function could be converted to probabilistic outputs by computing posterior probability (Franc & Zien & Scholkopf, 2011).

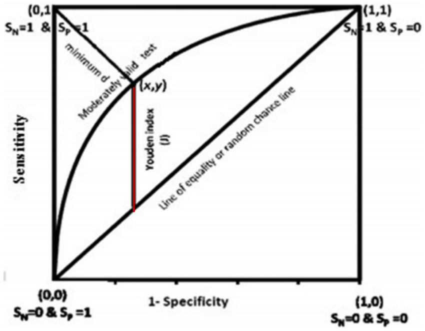
Easy-to-compute posterior probability:

$$p(y \mid \boldsymbol{x}; \tau, \omega, \boldsymbol{u}) = \frac{\exp(-\omega^{y} \ell(y\langle \tau\boldsymbol{u}, \boldsymbol{x}\rangle))}{\sum_{y\in\{1,-1\}} \exp(-\omega^{y} \ell(y\langle \tau\boldsymbol{u}, \boldsymbol{x}\rangle))} \qquad (\text{or } \boldsymbol{w} = \tau\boldsymbol{u})$$

This formula provides a convenient alternative to the commonly used Platt's approach which is based on retrospectively fitting the sigmoid function to the SVM outputs. The shape of SVM posterior is very similar to the Logistic model (sigmoid) used by Plat.

It discussed in *Receiver Operating Characteristics (ROC) Curve for Medical Researchers* that roughly speaking, existing criterions to find optimal threshold point from ROC curve can be characterized into points on curve closest to the (0, 1), Youden index, and minimize cost criterion (Kumar, 2011). The first two criterions give equal weight to sensitivity and specificity. The third criterion includes cost which mainly considers financial cost for correct and false diagnosis, and cost of further investigation when needed.

Figure1: Three methods find the best cut-off from the ROC curve

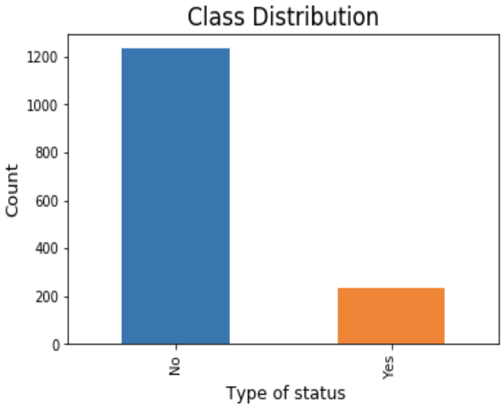


Methodology

There are four parts in the methodology section: data overview, data preprocessing, pattern discovery and pattern evaluation.

1. Dataset Overview

The IBM HR analytics on employee attrition and performance dataset, which was downloaded from Kaggle, has in total 1470 cases. The dataset includes features like Age, Employee Role, Daily Rate, Job Satisfaction, Years At Company, Years In Current Role etc. There are 34 features including both numerical and categorical features; 1 class feature attrition of 'yes' and 'no'. As we can see from the *Class Distribution* plot, the class is unbalanced.



2. Data Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format that could improve the performance of the classifier in some way.

- Data cleaning: No missing values are filled into the dataset. Three features named EmployeeCount, Over18, and StandardHours are removed directly since those features only have one value respectively which don't have practical meaning on model building.
- Data transformation: The numeric features are normalized by using StandardScaler(). It will normalize the features so that each feature on a training set will have mean = 0 and standard deviation = 1 so as to be able later reapply the same transformation on the testing set.
- Data reduction: There are two different dimensionality reduction methods are used.
  - 1. Feature selection: Eight features are selected based on their scores. Their scores are sorted by descending and are plotted on graph, The error rate vs. Number of features. When the error rate line approaches to the point, the number of features is equal to 8 resulting in the smallest error. The selected features are Monthly Income, Monthly Rate, Daily Rate, Total Working Years, Years at Company, Years In Current Role, Year with Current Manager, and Age
  - 2. Feature extraction: Principle Component Analysis (PCA), one of dimensionality reduction methods, facilities us extracting new features from original high-dimensional dataset. Two principal components are extracted which capture 99% of the variance for our training set.

- Imbalanced data: Two balanced methods are applied in this project.
    - 1. Undersampling: The random undersampling is used, RandomUnderSampler . It undersamples the majority class (No Attrition) to 189 by random selecting samples in training set and makes the number of majority class (No Attrition) equal to the number of minority class (Attrition).
    - 2. Oversampling: The random oversampling is used, RandomOverSampler. It oversamples the minority class (Attrition) to 987 by re-sampling samples in training set and makes the number of minority class (Attrition) equal to the number of majority class (No Attrition).

At first, the normalization, undersampling, oversampling, feature selection, feature extraction will be implemented individually on raw dataset. Two or three previous preprocessing methods will be implemented together on raw dataset afterwards. At last, the order in each combination of preprocessing methods will be changed and the new combination will be implemented on raw dataset again. To sum up, the way of preprocessing data results 20 kinds of combination (including individual one), generating 20 different datasets that could be tested performances of three types of probabilistic classifiers including Naive Bayes, KNN (after converting), and Linear Support Vector Machine (after converting).

3. Pattern Discovery

• Naive Bayes: A Probabilistic Classifier

Naive Bayes is a probabilistic classifier therefore no conversion is needed, which makes classification using posteriori decision rule. The probability of event H give evidence E is P(H/E) = P(E/H) * P(H)/P(E).

• KNN: Use Distance Weighted KNN algorithm converting to probabilities

The method to convert K-nearest neighbor algorithm to probabilistic classifier is using distance weighted KNN algorithm (Gou, 2012). First, given a query point, the closer the points to the query, the greater the weight w. The distance measure is Euclidean Distance. Thus, the weight w is: $w_i = \frac{1}{\mathrm{d}(x_q, x_i)^2}$

Then we normalize the weights into between 0 and 1 and regard them as probabilities.

• SVM: Convert Linear Support Vector Machine to probabilistic classifier

$$\min F(z) = -\sum_{i=1}^{l} l(tilog(pi) + (1-ti)\log(1-pi)), for\ pi = P\ A, B(fi),\ \ and\ ti = \frac{\frac{N++1}{N++2}}{\frac{1}{N-+2}}\ \ \frac{if\ yi = +1}{if\ yi = -1}, i = 1, \ldots .\mathrm{l}.$$

$$1 - pi = 1 - \frac{1}{1 + \exp\,(\mathrm{A}fi + \mathrm{B})}$$

The notes of converting Linear Support Vector Machine to probabilistic classifier are provided by LibSVM authors (Franc & Zien & Scholkopf, 2011). The probabilistic outputs are generated by the formula as above. However, there is a problem when the *pi* is close to 1. To figure out this problem, the authors came up with a way to solve this problem by implementing the formula as If $\mathrm{A}fi + \mathrm{B} \geq 0$ then use $\frac{1}{1+\exp(-\mathrm{A}fi-\mathrm{B})}$; Else use $\frac{\exp(\mathrm{A}fi+\mathrm{B})}{1+\exp(\mathrm{A}fi+\mathrm{B})}$.

Finally, the probabilistic outputs could be gained in by using predict_proba() for building our Linear Support Vector Machine probabilistic classifier.

• Methods to find the "optimal" threshold point
The Youden index is used in our project to find the optimal threshold point. Because this method can reflect our intention to maximize the correct classification rate and gives the same weight to sensitivity and specificity which have the same importance for us. The goal of Youden index is to maximize the difference between TPR (sn) and FPR (1 – sp) and yields J = max[sn+sp]. Once the J is yielded, the optimal point on ROC curved could be found and the value of this point could be identified by corresponding value on right Y-axis named threshold, ranging from 0 to 1 (Kumar, 2011). The case could be assigned to certain class based on the comparison output between its probability and the optimal point. If the value of its probability is larger than that of optimal point, it will be assigned to the Class 1 (Attrition), otherwise Class 0 (No attrition).

4. Pattern Evaluation

There are three evaluation metrics for checking our probabilistic classifiers' performances. Those metrics are accuracy, sensitivity and specificity, and ROC (Receiver Operating Characteristics) AUC (Area Under the Curve) curve.
•Accuracy
Accuracy simply treats all cases the same and reports a percentage of correct classification. The higher the accuracy, the better the performance of the classifier. The accuracy may can't be a good way to evaluate the performance of a classifier without any other evaluation metrics since our dataset is imbalanced. The majority class may yield a very high accuracy while this accuracy could not present the accuracy of minority class which is the class we mainly focus on. However, the accuracy can give us general idea that how does the performance of each classifier improve and make the comparison of performances easier.

•Sensitivity & Specificity
If a staff is intended to leave a company, how often he or she will be predicted to leave this company (true positive rate) in advance? *Sensitivity= true positives/ (true positive + false negative)*. If a staff is intended to leave a company, how often he or she will be predicted to leave this company (true positive rate) in advance? *Specificity= true negative/ (true negative + false positive)*. The Youden index aims to find the point that maximizes the sensitivity and specificity, making accurate prediction as high as possible and saving human resources cost as much as possible. In other word, the higher sensitivity and specificity, the better performance of the classifier.

- AUC - ROC curve

The AUC - ROC curve visualizes the tradeoff between sensitivity and specificity. The classifier that gives curve closer to the top-left corner indicates a better performance of the classifier. Random classifier is expected to give points lying along the diagonal as a baseline. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test. It means that the curve which is far away from the baseline stands for a better performance of the classifier. In our probabilistic classifiers, ROC gives a probability that reflects the degree to which an instance(staff) belongs to one class rather than another. We created a curve by varying the threshold for the probability. AUC is the total area under ROC curve for measuring the performance of classifier. The larger the AUC, the better is overall performance of the classifier correctly identifying the attrition of staff or not. The value of AUC will be displayed on the right bottom of figure.

- T-test

Once three types of probabilistic classifiers have been tested on the 20 datasets gained by different combination of data preprocessing. Their performances including accuracy, sensitivity, specificity, and AUC will be compared using paired T test. Let's take accuracy as an example.

Ha: $\boldsymbol{\mu}1acc - \boldsymbol{\mu}2acc < 0$ Ha: $\boldsymbol{\mu}1auc - \boldsymbol{\mu}2auc < 0$

Given Classifier 1 with $n1$ cases and $\boldsymbol{\mu}1auc$ and Classifier 2 with $n2$ cases and $\boldsymbol{\mu}2auc$, d is the difference of $\boldsymbol{\mu}1auc - \boldsymbol{\mu}2auc$ (2-sided test). Thus, $\sigma\, d = \boldsymbol{\mu}1auc\ (1 - \boldsymbol{\mu}1auc)/\ n1 + \boldsymbol{\mu}2auc\ (1 - \boldsymbol{\mu}2auc)/n2$. At 95% confidence level, $Z\alpha/2 = 1.96$: $dt = d \pm 1.96 \cdot \sigma\, d$. If the interval contains 0, the difference may not be statistically significant. It means that the null hypothesis could not be rejected, and Classifier 2 doesn't have a better performance with higher accuracy than that of Classifier 1. If the both values of interval are negative, the difference is statistically significant. It means that the null hypothesis could be rejected, and Classifier 2 do have a better performance with higher accuracy than that of Classifier 1. This is one of comparisons on accuracy. More comparisons will be made on sensitivity, specificity, and AUC following the same way. The $\boldsymbol{\mu}1se$, $\boldsymbol{\mu}1sp$ and $\boldsymbol{\mu}1auc$ stand for the mean sensitivity, mean specificity and mean AUC of Classifier 1. The best classifier under each type of classifier will be selected and those selected best classifiers will be paired and compared again to select the best classifier. Since we have three types of classifiers, Naive Bayes, KNN, and SVM, there will be three best classifiers selected from 20 classifiers and the best classifier will be selected among those three based on the significance of comparison of evaluation metric.

Results

- Parameter

First, we want to cover the parameters we selected for each classifier. For Naive Bayes, we use Gaussian NB, which a normal distribution is assumed. As we calculate the probabilities for each class using the frequency, we also calculate the mean and standard deviation of points for each class to summarize the distribution. Additionally, Gaussian NB works great with continuous values. For KNN, to select the best number of neighbors k, we calculate the errors for k equals to 1, 3, 5, 7, 0, 11, 13, 15. Based on our *KNN error rate plot* in appendix, we select when k = 5. For SVM, to select the best parameter c, soft margin cost function, controls the influence of each individual support vector. We select c=1 based on our *SVM error rate plot* in appendix.

- Evaluation Metrics

Next, the data preprocessing modules include 20 different combinations in different orders as shown in appendix *Results of Performance Metrics of Classifiers.* After running each classifier for the 20 different preprocessing steps, from the table below, we select the best preprocessing method with the highest overall scores. As shown in the table below, the best Naive Bayes classifier with undersampling, normalization and PCA, KNN with undersampling and normalization, SVM with oversampling and normalization. Compare to the raw data, the overall performance has increased for all classifiers. For example, the accuracy scores on KNN and SVM improve from .43 and .54 to .68 and .77. Furthermore, the classification threshold for each optimized classifier increase from raw data for our application to maximize both sensitivity and specificity, points above the threshold belongs to 1.

Classifier	Preprocessing	Sensitivity		Specificity		Accuracy		AUC		Threshold (Probability)
		Train	Test	Train	Test	Train	Test	Train	Test	
Naive Bayes	Raw Data	.66	.64	.86	.70	.83	.70	.80	.70	.26
	Undersampling + Normalization + PCA	.75	**.66**	.66	**.75**	.75	**.74**	.77	**.73**	.62
KNN	Raw Data	.11	.88	.99	.34	.85	.43	.81	.65	.11
	Undersampling + Normalization	.93	**.73**	.70	**.57**	.82	**.68**	.90	**.67**	.56
SVM (Linear)	Raw Data	0	.58	1	.62	.62	.54	.62	.61	.16
	Oversampling + Normalization	.71	**.67**	.71	**.79**	.78	**.77**	.77	**.73**	.61

- T- Test

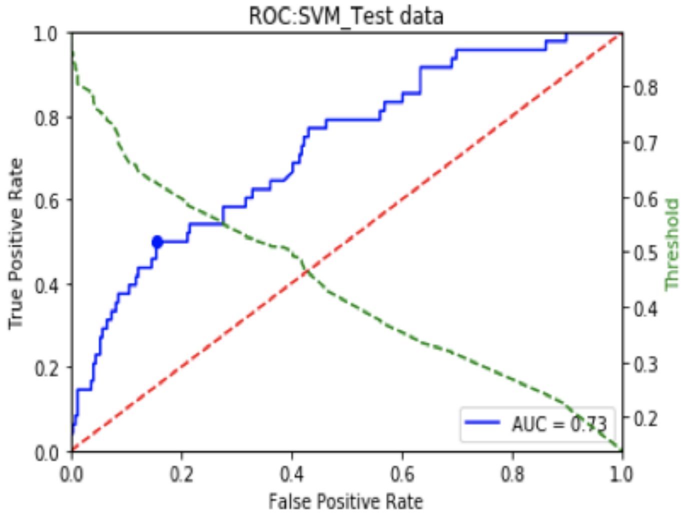
At 95% confidence level, $Z\alpha/2$ = 1.96: dt =d ± 1.96 ·σ d. If the interval contains 0, the difference may not be statistically significant. It means that the null hypothesis could not be rejected, and Classifier 2 doesn't have a better performance with higher accuracy than that of Classifier 1. Based on this method we mentioned in T-test, outputs are gained from best three probabilistic classifiers under each type of classifiers. Since the interval of sensitivity, specificity, accuracy, and AUC in the first comparison contain 0, the difference may not be statistically significant. It means that the null hypothesis could not be reflected and Classifier KNN with undersampling and normalization does not have a better performance than that of Classifier Naive Bayes. Since the interval of all evaluation metrics on the second comparison don't contain 0, the difference is statistically significant. It means that the null hypothesis could be reflected and Classifier SVM with oversampling and normalization does have a better performance than that of Classifier Naive Bayes with undersampling, normalization and PCA. Since three evaluation metrics don't contain 0, the difference is statistically significant. However, only interval of sensitivity contains 0 meaning insignificant. We make decision on combined outputs and say that Classifier SVM with oversampling and normalization does have a better performance than that of Classifier KNN with undersampling and normalization. The Classifier SVM with oversampling and normalization is the best classifier after three rounds comparisons.

At 95% confidence level:

No	**Comparison**	**Sensitivity_µ**se	**Specificity_µ**sp	**Accuracy_µ**acc	**AUC_µ**auc
1	NB>KNN	[-0.03, -0.01]	[0.13,0.15]	[0.02,0.11]	[0.09,0.14]
2	NB>**SVM**	[-0.053, -0.27]	[-0.12, -0.11]	[-0.04, -0.18]	[-0.07, -0.15]
3	KNN>**SVM**	[-0.009,0.01]	[-0.147, - 0.087]	[-0.18, -0.06]	[-0.21, -0.107]

From above, the classifier with the best overall performance is linear SVM with oversampling and normalization based on criterions of sensitivity, specificity, accuracy, AUC and threshold. The *Figure 2* is gained from the SVM classifier on test data. The y-axis is true positive rate on left, the y-axis is true positive rate on left, the y-axis is threshold value on right, and the x-axis is false positive rate. The baseline is marked in diagonal red line that can be obtained with a random classifier. The further our ROC curve is above the line, the better of our classifier. The best cut-off from the ROC curve is marked in blue point gained by using Youden index. For example, the blue point corresponds to FPR and TPR from the ROC curve and corresponds to threshold 0.61 from the threshold curve.

Figure 2: Youden index finds the best cut-off from the ROC curve



Conclusion

In this paper, we discussed our methods to preprocess our imbalanced dataset, convert non-probabilistic classifiers KNN and SVM into probabilistic classifiers, to evaluate each classifier based on the performance metrics we chose. At the end, we are able to quantify uncertainty of classification and conclude with a best classifier. The results of the study show that, linear SVM with oversampling has the highest overall performance (i.e. sensitivity, specificity, accuracy and AUC) compared to Naive Bayes and KNN, accuracy is 77% compared to 68% and 74%. The sensitivity, specificity and AUC are 67%, 79% and 73% with threshold value of 0.61. The t-test further demonstrated the performance of SVM on metrics over the other classifiers with over 95% significance. Another finding of this study is that the performances classifiers varies with different preprocessing steps. When we look at SVM model, the accuracy of normalization before oversampling is 67%, compared with normalization after oversampling is 77%. It also applies to undersampling. The accuracy of normalization before undersampling is 67%, compared with normalization after undersampling is 75%. Thus, the classifier provides better performance results if we normalize the data after oversampling or undersampling.

Future Work

Potential future directions of our study include:

1) Extend the predictive methods to conduct ensembles such as bagging and boosting, rather than a single model, to combine several models to improve predicted performance.
2) Test on larger datasets (type, size, no. of features): one of the limitation of our study is the data size is not large. To improve the overall performance, we would need larger datasets and more features related to attrition.
3) Apply other balanced methods to see their effects on probabilistic classifier: the oversampling technique we apply in this study is random oversampling. In the future, we could try another technique such as SMOTE to see if the performance increases.

Reference

Franc, Vojtech, Alexander Zien, and Bernhard Schölkopf. "Support Vector Machines as Probabilistic Models." *ICML*. 2011.

Gou, Jianping, et al. "A new distance-weighted k-nearest neighbor classifier." *J. Inf. Comput. Sci* 9.6 (2012): 1429-1436.

Jadhav, Sayali D., and H. P. Channe. "Comparative study of KNN, naive Bayes and decision tree classification techniques." *International Journal of Science and Research* 5.1 (2016).

Kumar, Rajeev, and Abhaya Indrayan. "Receiver operating characteristic (ROC) curve for medical researchers." *Indian pediatrics* 48.4 (2011): 277-287.

Lin, Hsuan-Tien, Chih-Jen Lin, and Ruby C. Weng. "A note on Platt's probabilistic outputs for support vector machines." *Machine learning* 68.3 (2007): 267-276.

Noel Cressie and Christopher K. Wikle, "Expecting Too Much Certainty." *Statistics for Spatio-Temporal Data*, Wiley, 2011, p. 4

Rüping, Stefan. *A simple method for estimating conditional probabilities for svms*. No. 2004, 56. Technical Report/Universität Dortmund, SFB 475 Komplexitätsreduktion in Multivariaten Datenstrukturen, 2004.

Thammasiri, Dech, et al. "A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition." *Expert Systems with Applications* 41.2 (2014): 321-330.

Yigit, Halil. "A weighting approach for KNN classifier." *Electronics, Computer and Computation (ICECCO), 2013 International Conference on*. IEEE, 2013.

Appendix

Peer Evaluation (0-10 scale) : Jiayu Qi.

1	Obtain the results **10**	She is very supportive to combine the outputs after I provided the part of codes for KNN and Naïve Bayes. She provided some suggestions on improving the performances of classifiers when she looked the result. The table for displaying results is provided by her. It looks very concise and organized holding many results.
2	Final presentation **9**	She drafted the template of the ppt slides and is mainly responsible for adjusting the content and format.
3	Final report **10**	She finished almost 45% of final report, including introduction, abstraction, part of related work, KNN, NB, conclusion, and feature work. She is very support to finish this project.

Results of Performance Metrics of Classifiers (Training)

ML **Data**	**Training Data** **Naïve Bayes / KNN / SVM(Linear)**			
	Sensitivity	Specificity	Accuracy	AUC
Raw data	.66/.11/0	.86/.99/1	.83/.85/.62	.80/.81/.62
Undersampling	.78/.55/.69	.62/.73/.51	.40/.65/.60	.81/.75/.64
Oversampling	.53/.44/.55	.80/.90/.55	.60/.65/.55	.81/.87/.58
Norm w/ raw data	.19/.22/.12	.91/.14/.98	.99/.64/.85	.75/.75/.16
Norm + Under	.39/.70/.71	.87/.68/.71	.83/.63/.72	.76/.78/.78
Norm + Over	.47/.93/.71	.85/.72/.70	.66/.83/.71	.75/.90/.78
Under + Norm	.33/.93/.71	.85/.70/.70	.62/.82/.71	.74/.90/.77
Over + Norm	.46/.93/.71	.83/.70/.71	.65/.82/.78	.73/.90/.77
PCA	.0/.11/0	1/.99/1	.84/.85/.84	.64/.79/.57
PCA + Under	.82/.73/.73	.26/.64/.47	.54/.69/.6	.64/.75/.64
PCA + Over	.52/.66/.72	.76/.54/.44	.72/.56/.58	.64/.61/.66
Under + PCA	.65/.54/.72	.62/.65/.43	.62/.63/.58	.63/.61/.66
Over + PCA	.81/.91/.72	.29/.65/.44	.56/.78/.58	.64/.86/.66
Under + Norm + PCA	.75/.74/.72	.66/.64/.7	.75/.69/.71	.77/.76/.77
Over + Norm + PCA	.81/.91/.68	.29/.65/.47	.56/.79/.58	.64/.86/.61
Feature selection	.74/.70/0	.66/.71/1	.70/.71/.84	.77/.78/.61

Under + F selection	.76/.74/.71	.50/.67/.48	.63/.71/.60	.71/.76/.66
Over + F selection	.75/.91/.54	.49/.65/.53	.63/.79/.54	.69/.87/.55
Under + Norm + F selection	.68/.65/.67	.57/.66/.45	.63/.66/.56	.68/.71/.61
Over + Norm + F selection	.67/.91/.61	.54/.60/.51	.63/.76/.57	.69/.78/.63

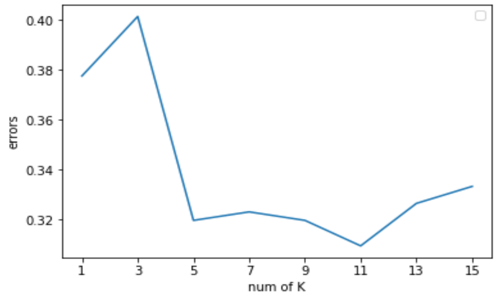
Results of Performance Metrics of Classifiers (Testing)

ML **Data**	**Testing Data** **Naïve Bayes / KNN / SVM(Linear)**				
	Sensitivity	Specificity	Accuracy	AUC	Threshold
Raw data	.64/.88/.58	.70/.34/.62	.70/.43/.54	.70/.65/.61	.26/.11/.16
Undersampling	.72/.52/.67	.67/.67/.61	.68/.61/.62	.74/.61/.66	.61/.56/.51
Oversampling	.71/.75/.93	.65/.58/.23	.67/.61/.34	.74/.66/.57	.55/.44/.47
Norm w/ raw data	.62/.63/.67	.74/.67/.70	.72/.66/.70	.71/.71/.74	.07/.22/.16
Norm + Under	.60/.46/.77	.74/.80/.60	.72/.75/.63	.69/.69/.73	.22/.67/.45
Norm + Over	.58/.71/.70	.79/.64/.66	.76/.65/.67	.70/.70/.74	.34/.56/.46
Over + Norm	.77/.67/.67	.59/.68/.79	.62/.68/.77	.69/.72/.73	.12/.44/.61
Under + Norm	.65/.73/.65	.71/.57/.80	.70/.68/.77	.70/.67/.74	.29/.56/.62
PCA	.52/.75/.87	.74/.37/.28	.71/.43/.37	.64/.56/.56	.22/.11/.13
PCA + Under	.65/.54/.54	.61/.65/.72	.62/.63/.69	.63/.61/.65	.60/.56/.52
PCA + Over	.52/.66/.90	.76/.54/.28	.72/.56/.38	.64/.61/.58	.60/.44/.45
Under + PCA	.65/.54/.52	.62/.65/.75	.62/.63/.71	.63/.61/.66	.60/.56/.55

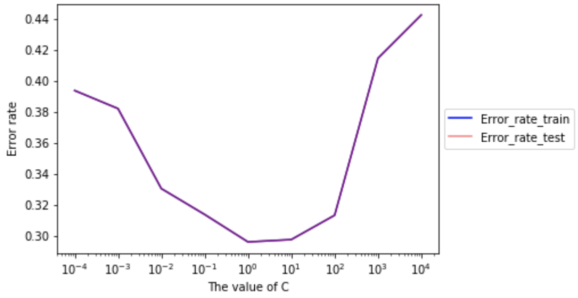
Over + PCA	.52/.67/.65	.75/.54/.61	.72/.56/.62	.64/.61/.66	.60/.44/.52
Under + Norm + PCA	.66/.69/.63	.75/.69/.80	.74/.69/.77	.73/.70/.60	.62/.56/.74
Over + Norm + PCA	.90/.81/.88	.32/.36/.43	.42/.44/.43	.60/.59/.59	.43/.33/.43
Feature selection	.58/.79/.81	.67/.39/.41	.66/.45/.48	.66/.59/.64	.28/.11/.14
Under + F selection	.50/.72/.69	.74/.46/.61	.70/.51/.63	.66/.61/.66	.78/.44/.51
Over + F selection	.58/.62/.60	.67/.60/.60	.66/.60/.52	.66/.64/.52	.66/.44/.50
Under + norm + F selection	.46/.75/.79	.80/.43/.48	.74/.48/.53	.65/.59/.48	.62/.44/.64
Over + Norm +F selection	.60/.27/.46	.65/.87/.80	.65/.78/.74	.64/.57/.64	.52/.78/.57

Select parameter

KNN Error Rate Plot



SVM - Error Rate Plot



Feature Selection

```
                  Feat_names     F_Scores
10             MonthlyIncome  94903.809453
11               MonthlyRate   2357.165804
1                  DailyRate    652.141075
17         TotalWorkingYears    202.459576
20            YearsAtCompany    120.183570
21        YearsInCurrentRole     93.907474
23      YearsWithCurrManager     92.625091
0                        Age     70.068292
2           DistanceFromHome     61.436285
4             EmployeeNumber     19.929315
8                   JobLevel     19.069412
16          StockOptionLevel     16.248151
30                  OverTime     11.519305
6                 HourlyRate     10.346676
```