PREDICTING EMPLOYEE ATTRITION USING VARIOUS ML ALGORITHMS

P. D. Sai Vardhan | A. Sai Kaushik | Balagopal. T. S | Dr. Sairabanu J | SCOPE

Introduction

Employee attrition is the loss of employees due to personal reasons, low job satisfaction, low salary, and/or a bad business environment. Aim of our project is to develop a software that predicts employee attrition using various parameters utilizing various machine learning algorithms and also compare the results.

Motivation

One of the major issues facing business leaders within companies is the loss of talented employees. This project studies employee job satisfaction using machine learning models.

Scope of the Project

In the base paper the authors have used ADASYN as the oversampling technique and then fused it with other machine learning algorithms to predict whether the employee is going to leave the company or not. In our paper we are implementing SMOTE as the oversampling technique and then test it against other oversampling techniques. We use the database which is openly available from IBM's repository and then refine it to remove any anomalies that might skew our results later on. So in order to refine the database we use oversampling techniques as stated above and then make the refined database undergo prediction based machine learning algorithms to get results.

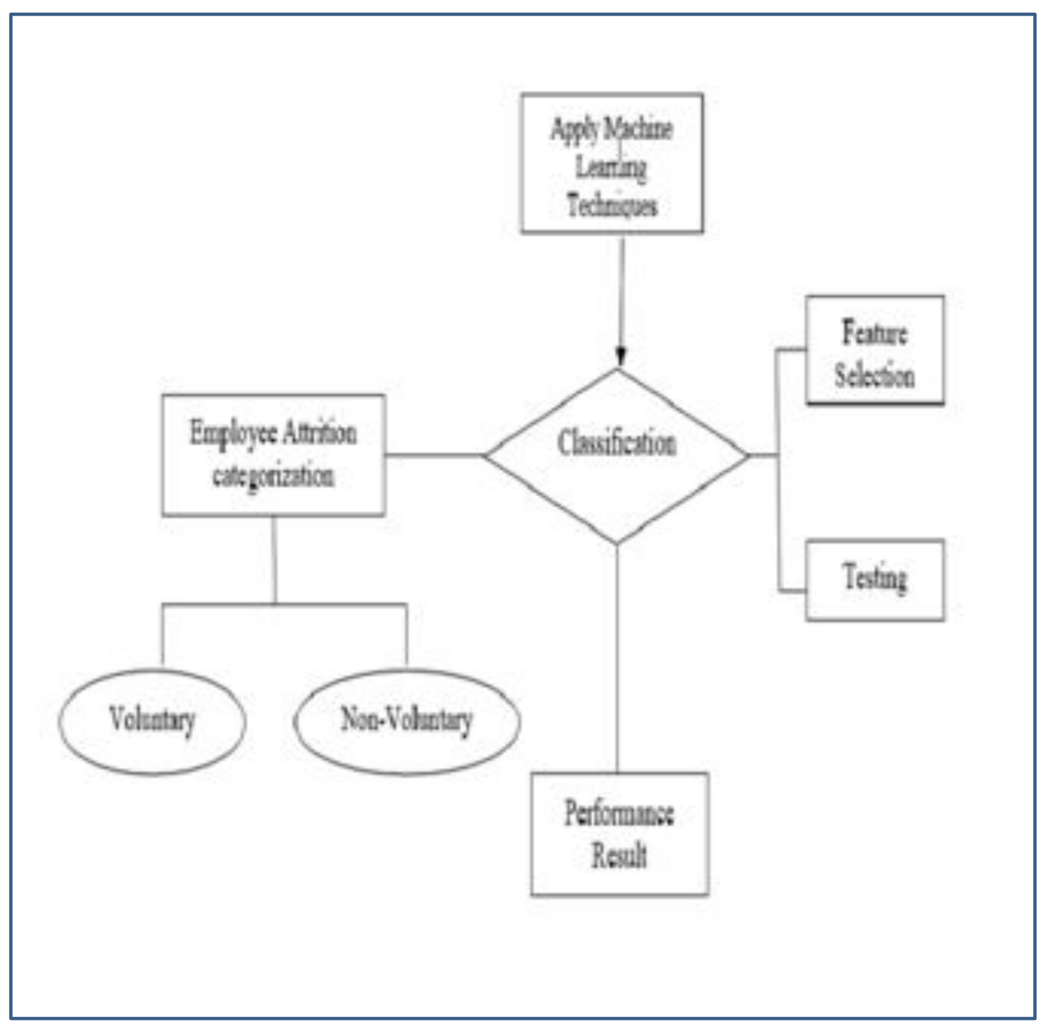
Methodology

I. Related concepts

In this project the model is prepared in two steps. First the imbalanced data is converted to balanced data using SMOTE and Random Sampling. Then the data is trained using the Classifier models: KNN, Random Forest classifier, SVM and Artificial Neural Network(ANN).Then their performance is compared in terms of accuracy and precision. The existing paper uses ADASYN for sampling but we are using random sample and SMOTE instead. We also intend to compare the results thus obtained and infer which machine learning algorithm is best suitable for predicting the results.

II. Proposed System Methodology

Data Pre-processing:
The dataset is highly imbalanced and it is converted to balanced data by using over-sampling methods like ADASYN or SMOTE. After that the data is standardized or normalized to avoid over fitting also null values are replaced with zeros. Redundant and irrelevant columns are also removed in this step.



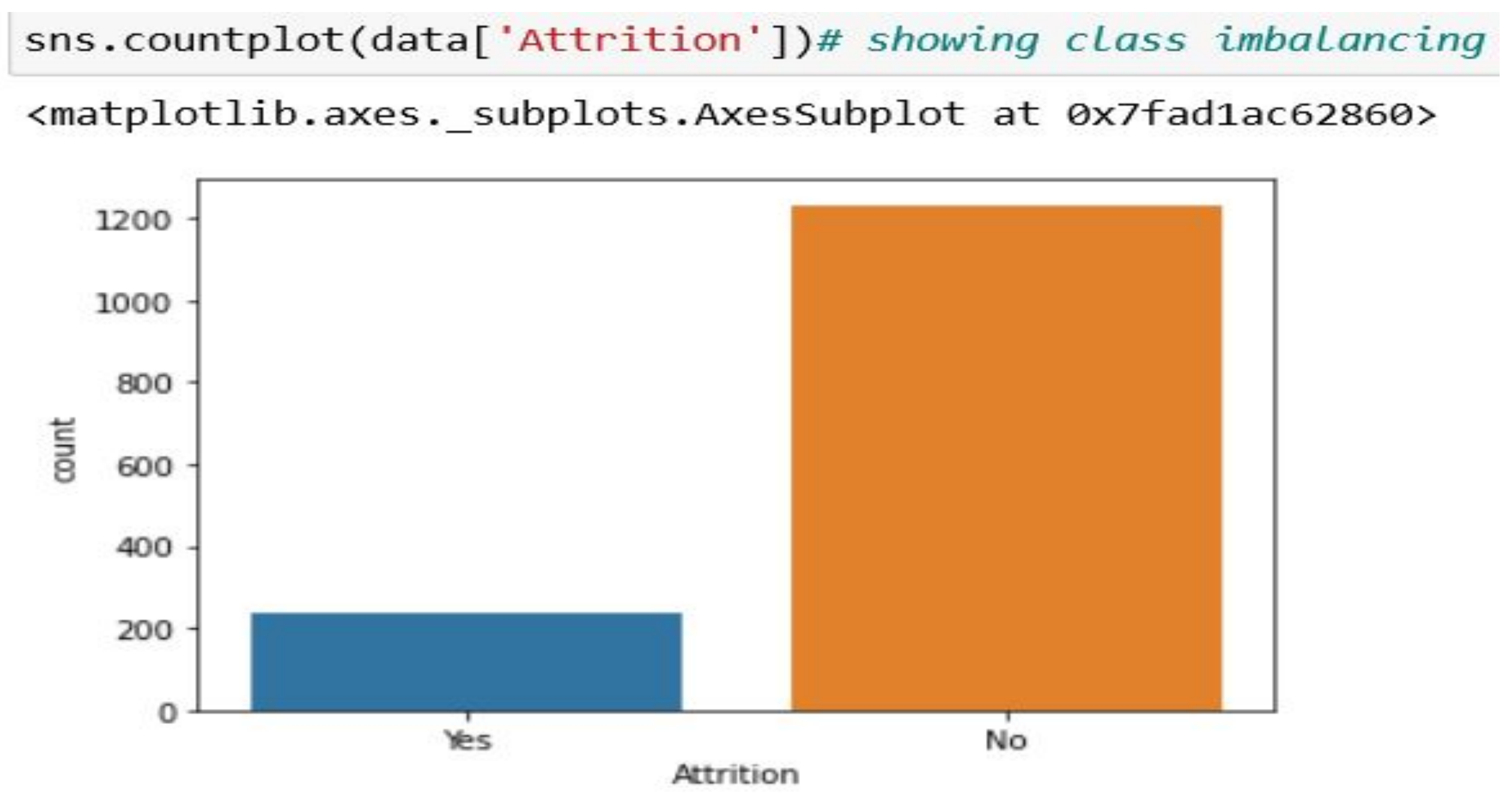
Classification flowchart

Data splitting:
Now 70 % of the data is used for training and 30% is used for testing, maintaining the class ratio. Training the model: As different models are to be used so k-fold cross validation is used for proper model selection. Then we fit our data to the models to evaluate the performance. Also we use different parameters for choosing the best model.

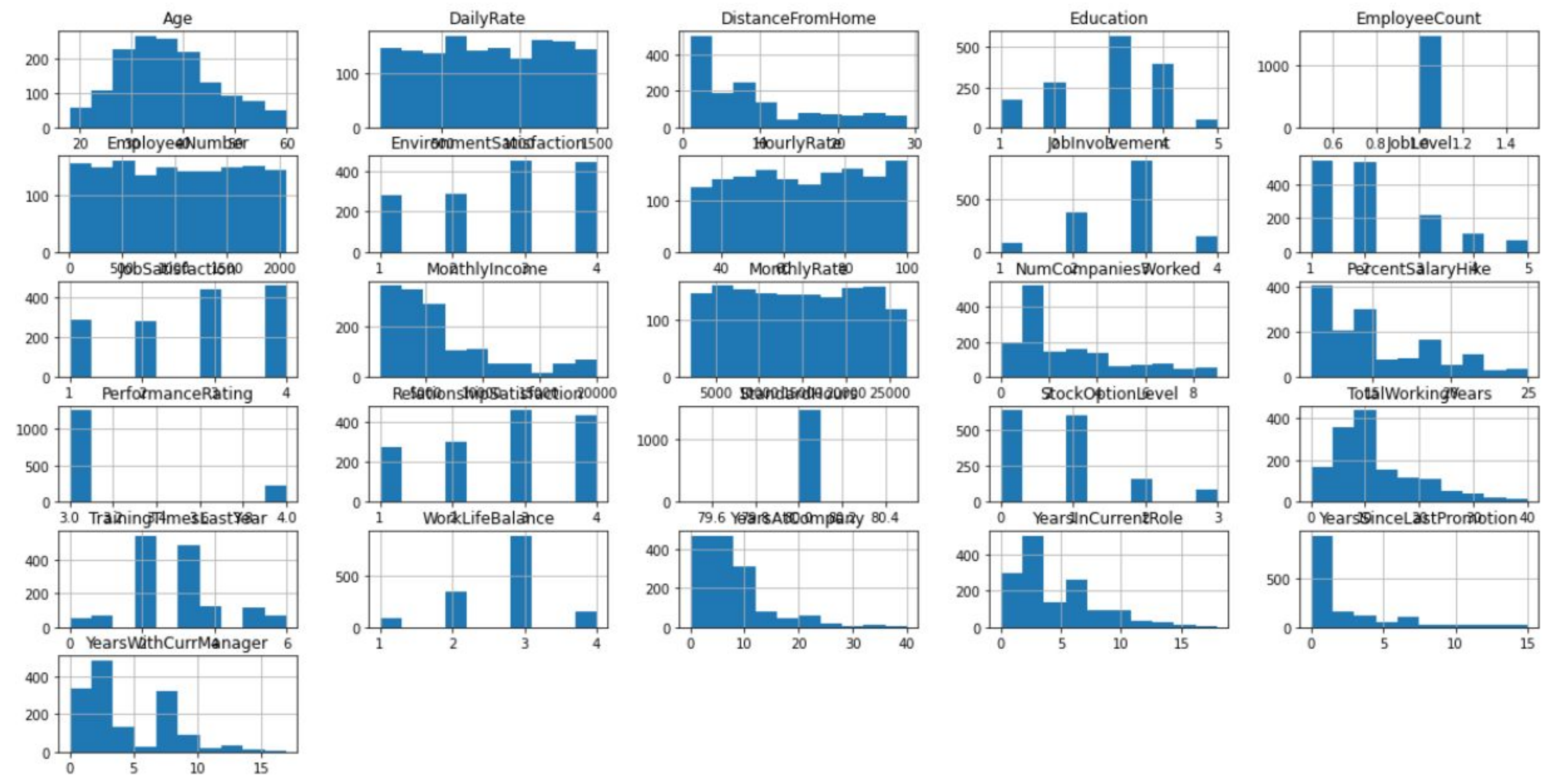
III. Requirement Analysis

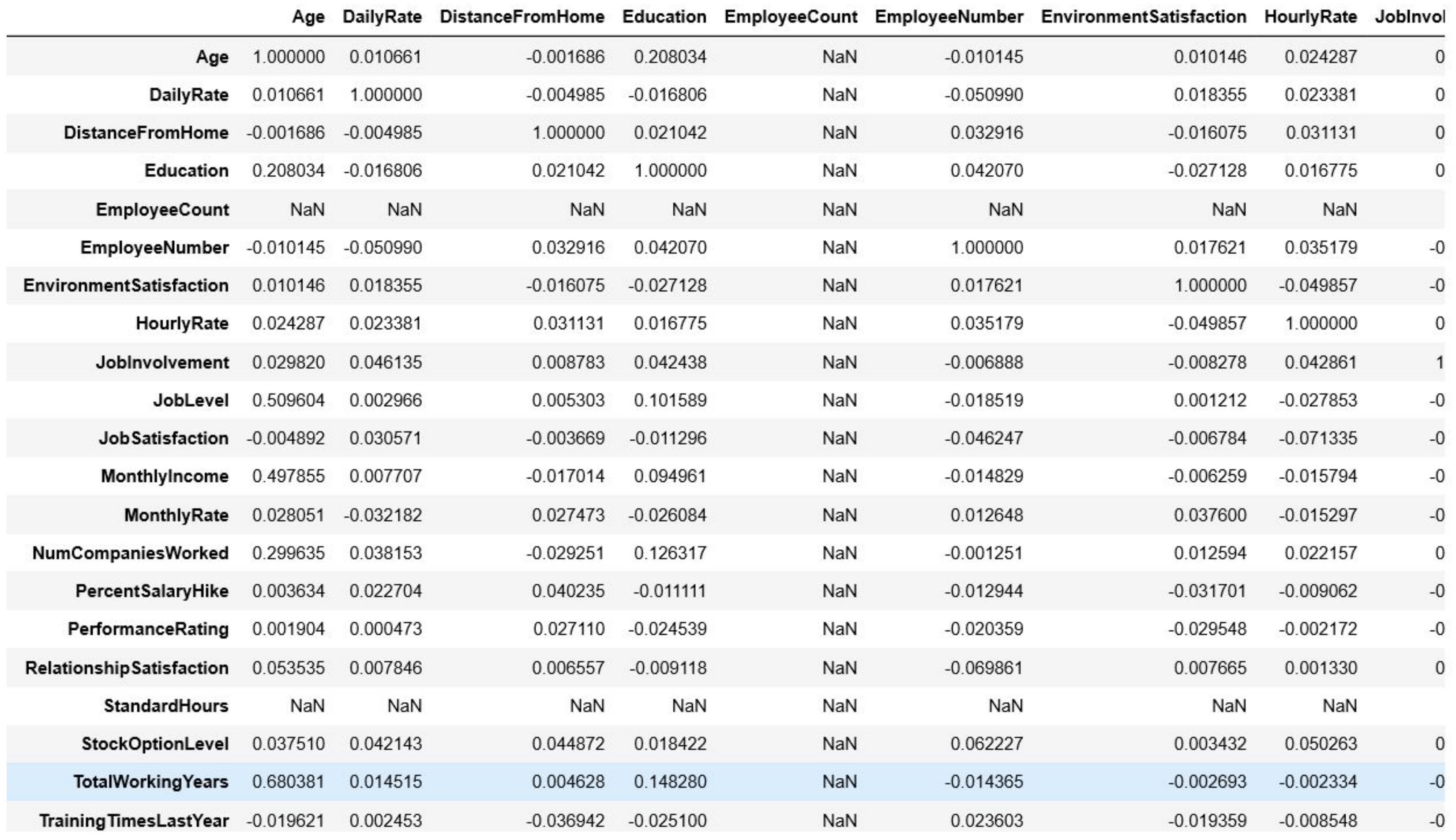
Hardware Requirements
Processor – i3 / i5
RAM - 4GB / 8GB
System Free Space - Minimum 15GB
Software Requirements
Programming Language - Python
IDE - Anaconda
User Interface - Visual Studio

Results



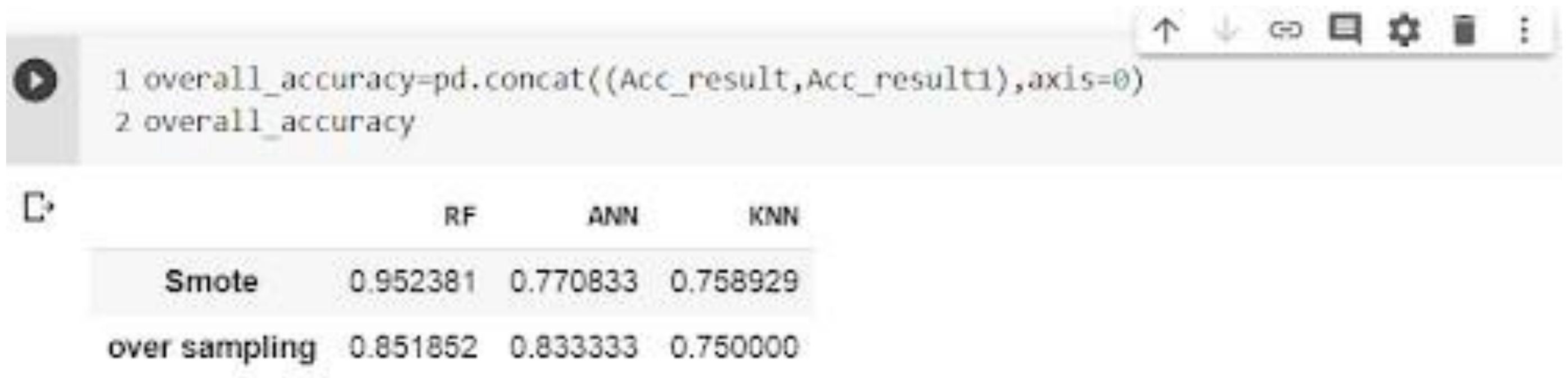
```
sns.countplot(data['Attrition'])# showing class imbalancing
<matplotlib.axes._subplots.AxesSubplot at 0x7fad1ac62860>
```





	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvol
Age	1.000000	0.010661	-0.001686	0.208034	NaN	-0.010145	0.010146	0.024287	0
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN	-0.050990	0.018355	0.023381	0
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	NaN	0.032916	-0.016075	0.031131	0
Education	0.208034	-0.016806	0.021042	1.000000	NaN	0.042070	-0.027128	0.016775	0
EmployeeCount	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070	NaN	1.000000	0.017621	0.035179	-0
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	NaN	0.017621	1.000000	-0.049857	-0
HourlyRate	0.024287	0.023381	0.031131	0.016775	NaN	0.035179	-0.049857	1.000000	0
JobInvolvement	0.029820	0.046135	0.008783	0.042438	NaN	-0.006888	-0.008278	0.042861	1
JobLevel	0.509604	0.002966	0.005303	0.101589	NaN	-0.018519	0.001212	-0.027853	-0
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	NaN	-0.046247	-0.006784	-0.071335	-0
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	NaN	-0.014829	-0.006259	-0.015794	-0
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	NaN	0.012648	0.037600	-0.015297	-0
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	NaN	-0.001251	0.012594	0.022157	0
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	NaN	-0.012944	-0.031701	-0.009062	-0
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	NaN	-0.020359	-0.029548	-0.002172	-0
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	NaN	-0.069861	0.007665	0.001330	0
StandardHours	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	NaN	0.062227	0.003432	0.050263	0
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280	NaN	-0.014365	-0.002693	-0.002334	-0
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100	NaN	0.023603	-0.019359	-0.008548	-0

Data processed using SMOTE tends to yield higher accuracy results after training with the above mentioned machine learning models when compared to random sampling. SMOTE gave accuracy of 95.2%, 77% and 75.8% for random forest, artificial neural network and k-NN, respectively.



```
1 overall_accuracy=pd.concat((Acc_result,Acc_result1),axis=0)
2 overall_accuracy
```

```
                    RF       ANN       KNN
Smote         0.952381  0.770833  0.758929
over sampling 0.851852  0.833333  0.750000
```

Conclusion

Based on the above results, we were able to conclude that of all the techniques we used, the combination of SMOTE and Random Forest has the highest accuracy. Thus we can say that SMOTE used with Random Forest gives a slightly better accuracy than ADASYN with Random Forest.

Compared with Random Oversampling, SMOTE was the better accurate oversampling technique even though it has less accuracy when combined with ANN. Random Forest algorithm provided better results compared with ANN and KNN.

References

[1] S. Kaur and R. Vijay, "Job Satisfaction – A Major Factor Behind Attrition or Retention in Retail Industry", Imperial Journal of Interdisciplinary Research, vol. 2, no. 8, 2016.

[2] I. Paparrizos, B. B.Cambazoglu, A. Gionis, "Machine learned job recommendation", RecSys '11: Proceedings of the fifth ACM conference on Recommender systems, October 2011