A comparison of Naïve Bayes (NB) and Random Forest (RF) on Predicting Employee Departures

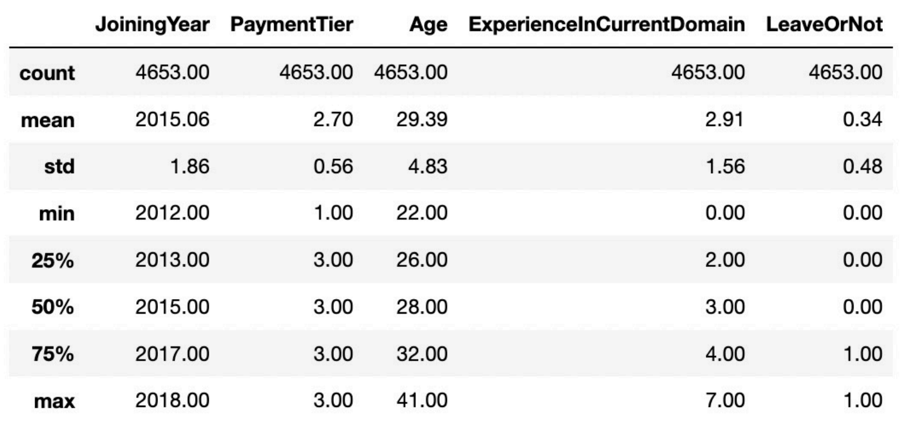
Edward St John

Description and Motivation

Due to the costs associated with the hiring process of employees (interviews with applicants and training up new hires), there is a huge amount of value for companies in predicting which employees are more likely to leave in the hope that they could put in measures to persuade the employee to leave in order to avoid these costs. This study aims to predict this using the Naïve Bayes and Random Forest supervised classification models, comparing and analysing the performance of both and comparing the performance to a previous study by Rawat (2019)[1].

Initial analysis of data including basic statistics

- Dataset: Employee Future Prediction from Kaggle
- The original dataset contains a total of 4653 rows with 9 columns (8 features along with 1 target column)
- The target column (LeaveOrNot) contains a 0 if the employee stays at the company and a 1 if they end up leaving.
- Descriptive statistics have been generated for all the numerical features in Figure 1, with the most noticeable findings being that the majority of employees are in the top payment tier (3) and that there is a large range of experience from entry-level to 7 years.
- A correlation matrix of all the features (having been converted into numerical columns for analysis) in Figure 2 show a relatively flat correlation across the with the exception of the city categories which is to be expected as e.g., if the employee is based in New Delhi they wouldn't be based in Bangalore.
- Figure 3 and Figure 4 show small class imbalances both in where different genders are located along with the target column.
- Figure 5 shows the distribution of ages in the workforce along with their respective education. This shows the company employs the majority of their employees in their mid-to-late twenties indicating a relatively young workforce, with the majority only having a Bachelors with a very small minority having a PHD across the board.



	JoiningYear	PaymentTier	Age	ExperienceInCurrentDomain	LeaveOrNot
count	4653.00	4653.00	4653.00	4653.00	4653.00
mean	2015.06	2.70	29.39	2.91	0.34
std	1.86	0.56	4.83	1.56	0.48
min	2012.00	1.00	22.00	0.00	0.00
25%	2013.00	3.00	26.00	2.00	0.00
50%	2015.00	3.00	28.00	3.00	0.00
75%	2017.00	3.00	32.00	4.00	1.00
max	2018.00	3.00	41.00	7.00	1.00

Figure 1

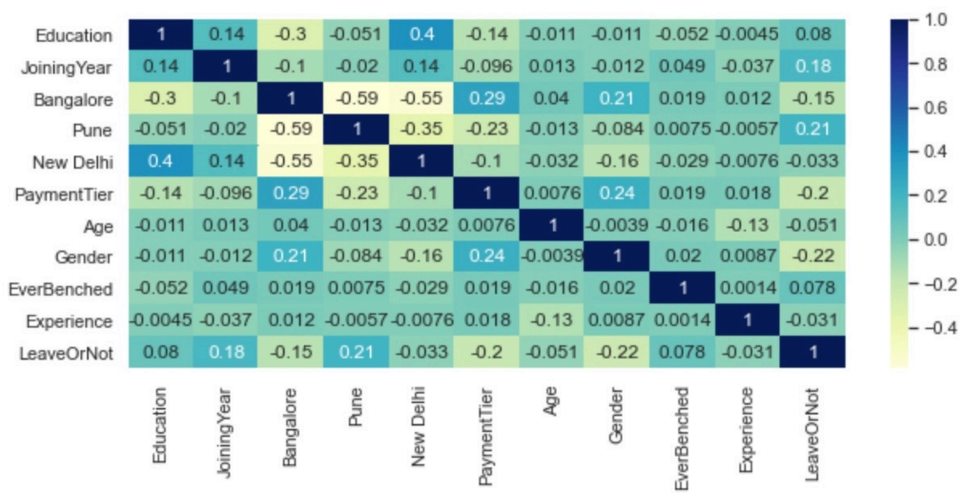
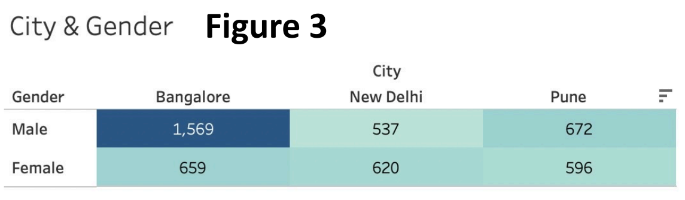


Figure 2

Figure 3



City & Gender

Gender	City Bangalore	 New Delhi	 Pune
Male	1,569	537	672
Female	659	620	596

Figure 4

Amount of Employees Leaving

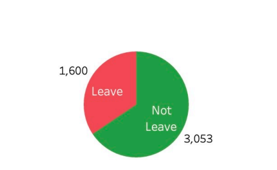
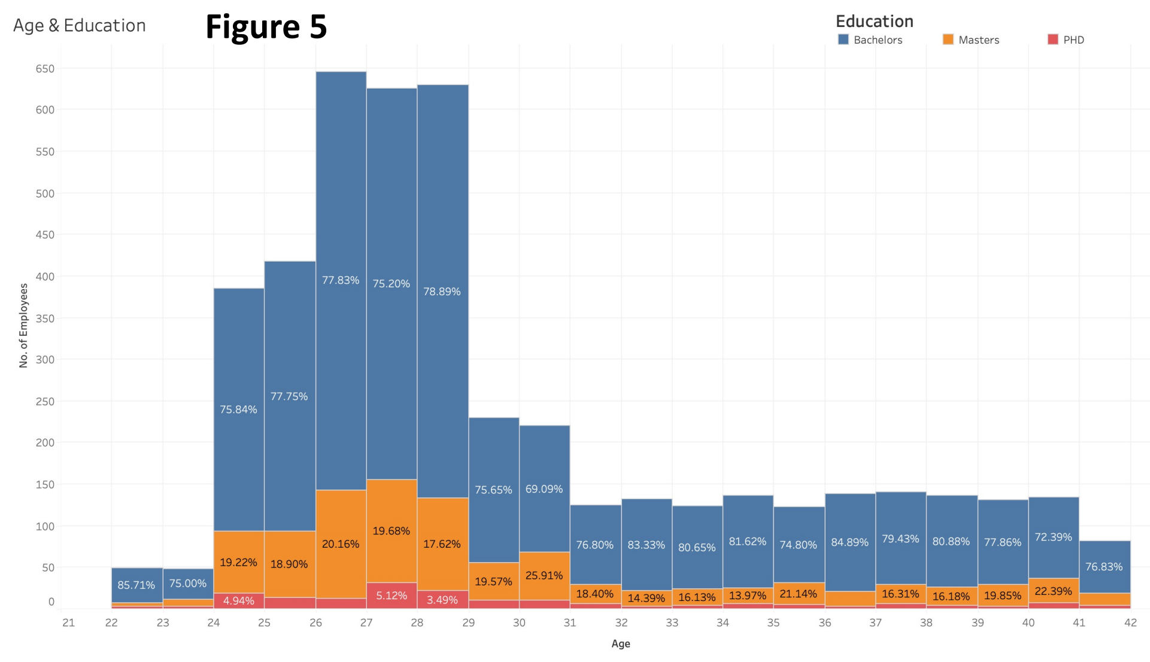


Figure 5



Naïve Bayes

- Naïve Bayes is a supervised classification algorithm founded in Bayes Theorem.
- It assumes independence of attributes, often resulting in sub-optimal results[2] (hence the term 'Naïve'). Although this is a poor assumption and often not the case (e.g., our attributes are not independent of each other with the city attribute as an example), in practice it often manages to compete with more complex classifiers[3].
- It uses a prior probability which is multiplied to a calculated likelihood, creating a posterior (which is essentially a conditional probability), updating the prior probability after having access to new information.

Pros:

- It can handle both categorical and continuous features
- It's a relatively simply model making it extremely easy to implement
- Little training data is required to compute a reasonable output compared to more complex ones who require a lot more data to get an answer with much value
- Performs admirably compared to more sophisticated classifiers (as said above) despite its independence assumption

Cons:

- Most datasets don't have independent features meaning more complex algorithms often perform better
- If a categorical variable is present in test data but not in training data, NB will assign a probability value of 0 which produces no prediction. In this case, smoothing techniques need to be applied which takes time to implement

Hypothesis Statement

- It is expected that the RF model performs better overall by most metrics over the NB model given the more complex algorithm.
- However, it is also expected that RF will take longer to both compute and train compared to NB given its extra complexity.
- Both models should perform better than a guess at random in predicting if an employee will leave or not.
- The minority class of employees leaving is likely to have more misclassification compared to staying.

Choice of parameters and experimental results

Naïve Bayes

- Model hyper-parameters chosen were Distribution Name as 'kernel' and the Width set at 0.15551. A total of 20 function evaluations were performed to arrive at these parameters.
- Grid search was used as the method for hyper-parameter tuning, which performs an exhaustive search based on a defined subset of hyper-parameter space[5].

Experimental results:

- Decreasing the K-fold from 10 to 5 yielded nearly identical accuracy whilst decreasing the training time, so it was kept at 5 for the optimal model
- Figure 1 shows various metrics to evaluate the model
- Figure 3 shows the iteration process for how many function evaluations it took to achieve the minimum objective function value

Figure 1

Metric	NB
Training accuracy	79.2264
Training time	27.3s
Testing accuracy	79.5357
Precision	0.7780
Recall	0.7566
F1 Score	0.7672

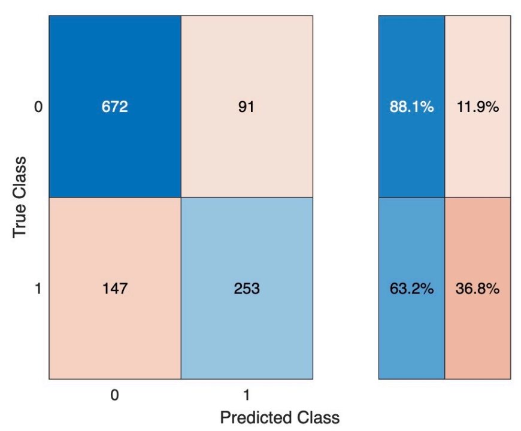


Figure 2

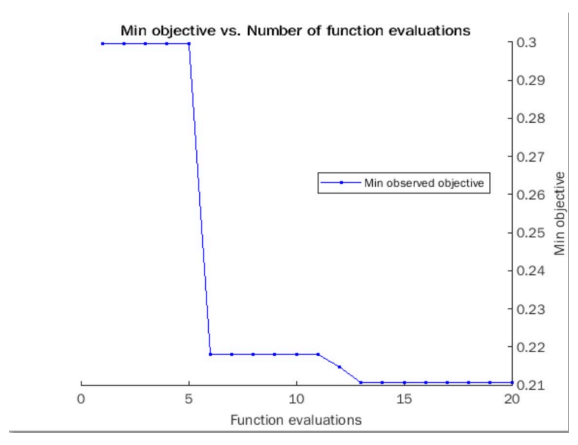


Figure 3

Random Forest

- Model fitted using the 'SampleWithReplacement' method as a bootstrapping algorithm, meaning the sample is able to contain multiple instances of the same data[6].
- Choice of parameters:
  - NumTrees = 150
  - NumPredictorsToSample = 9

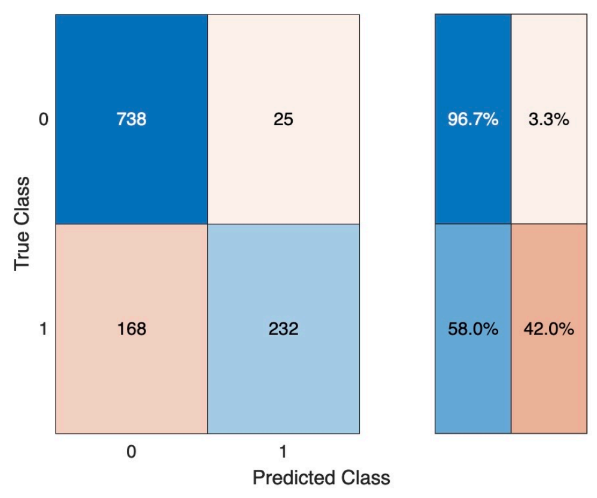
Experimental results:

- Figure 1, as above, shows various metrics to evaluate the model allowing a comparison between both the training and testing metrics
- Figure 2 (for both here and above) illustrates the confusion matrix for each model on the testing data

Figure 1

Metric	RF
Training accuracy	82.8367
Training time	32.0s
Testing accuracy	83.1470
Precision	0.8586
Recall	0.7736
F1 Score	0.8139

Figure 2



Random Forest

- The Random Forest model uses a collection of decision trees, who on their own are prone to high bias and often yield weak predictions. However, they join together to compute a majority vote for the classification often yielding a much more accurate prediction.
- Each decision tree only has access to a random set of the training data due by resampling the data with replacement (called bootstrapping). The increased variation from this with each tree improves the overall majority prediction.

Pros:

- Much less prone to overfitting compared to to regular decision trees by using the bagging method along with random feature selection
- Very versatile algorithm as it can be used for both classification and regression
- Good at managing missing values as well as outliers
- Performs well with imbalanced datasets due to the combination of ensemble learning and its sampling technique

Cons:

- By using lots of decision trees, RF can often take up a huge amount of memory for large datasets making it a lot slower than other less complex and more efficient algorithms and therefore less useful for real-time predictions
- Ensembles tend to overtrain, producing overly optimistic estimates of their predictive power[4], making it hard to evaluate the predictive quality of the model

Description of the choice of training and evaluation methodology

- Original dataset processed to convert categorical attributes into numerical attributes for ease of use in NB
- Dataset split 75/25 for training and testing data.
- Nested 'for' loops used for hyper-parameter training with RF to mimic K-fold results (creating additional freedom for writing other code within the loops) along with resampling with replacement each loop, whereas K-fold cross validation used for NB to estimate generalisation error.
- Tune models according to validation accuracies, confusion matrices and generalisation error to find the optimal model.
- Test both 'best' models with the optimal parameters by predicting the test set (split as described above) which is so far unseen.
- Investigate performance metrics for each model comparing accuracy, confusion matrix F1 score and training times

Analysis and critical evaluation of results

- The testing accuracy of the Random Forest model has a moderately better performance result compared too the Naïve Bayes model by about 3.6%. Part of this outperformance is likely due to the independence assumption taken by the NB model (with its resulting pitfalls described as above).
- As shown in Figure 1 (for each model respectively opposite), RF takes longer to train than NB as expected from out hypothesis. However, we would have expected RF to take much longer to train rather than just a matter of about 5 seconds. This is most likely due to a relatively simple dataset, with this effect of a longer training time becoming more pronounced with a larger and more complex dataset should this experiment be repeated.
- Surprisingly, the testing accuracy for both NB and RF slightly outperformed their training accuracies. This could just be due to a relatively small test set of data (having split it 75/25 for the train/test set), meaning there could just be high variance skewed towards the '0' target class of employees not leaving meaning the models seemed more accuracy than they actually were given the class imbalance.
  - This high accuracy issue can also be illustrated in the confusion matrices, with both models showing a very high percentage (and number) of correct '0' predictions but a relatively poor (but also lower) number of correct '1' predictions skewing the accuracy figure.
- Given the unreasonably high accuracies above due to the imbalance (leading to the models likely being overfitted creating bias in favour of the majority class) other metrics need to be used in order to evaluate the results of the model performance. For this, we have used precision, recall and F1 Score.
- The RF model outscores the NB model on precision by ~8% which is relatively significant. This shows RF is much better at correctly classifying the positive class (i.e. an employee leaving). This means we can trust the RF model much more when it comes to its predictions of an employee actually leaving compared to the NB model.
- In terms of recall, RF again outperformed NB (albeit my a much smaller amount compared to precision) by around 1.7%. This illustrates that RF is better at finding all the members of the positive class by having fewer false negatives in its prediction, meaning its better at finding a higher amount of actual employees leaving the company than NB.
- In order to assess performance in terms of a trade-off between precision and recall, their F1 scores are used. In this metric, RF outperforms NB by ~4.7% which is a reasonable margin.
- Given the outperformance of the RF model over NB in all the metrics (apart from training time), this affirms that RF seems to be the superior model based on the analysis we've done.

Lessons learned

- Model performance is much better being evaluated with many different metrics rather than just one, such as Accuracy. In the case of e.g. imbalanced data, a much more reliable performance overview is achieved by combining different metrics in order to gain a better overall opinion of the performance.
- Whilst ensemble models (such as RF as used her) tend to perform better than NB with prediction, the higher complexity of these models can sometimes lead to overfitting which needs to be taken into account when optimising the hyper-parameters

Future work

- Due to the imbalance in the data, implementing the SMOTE technique for rebalancing the data could be beneficial in improving both models' performance, although analysis would need to be done to see if this would be the case
- A different dataset in the same domain would provide an interesting comparison in model performance
- The models could be optimised through feature selection[7] in order to potentially remove irrelevant features, reducing the complexity which would greatly help the training time and also increase the ease of deploying the model at a larger scale (saving resources in the process)

References

[1] Rawat, S., 2019. Heart Disease Prediction. (online) *Medium*. Available at: https://towardsdatascience.com/heart-disease-prediction-73468d630cfc [Accessed 19 December 2021].

[2] Kaviani, P. and Dhotre, S., 2017. Short Survey on Naïve Bayes Algorithm. *International Journal of Advance Engineering and Research* Development (Vol. 4, No. 11, pp.607-611).

[3] Rish, I., 2001, August. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, No. 22, pp. 41-46).

[4] Test Ensemble Quality. (online) *Mathworks*. Available at: https://uk.mathworks.com/help/stats/methods-to-evaluate-ensemble-quality.html [Accessed 19 December 2021].

[5] Syarif, I., Prugel-Bennett, A. and Wills, G., 2016. SVM parameter optimization using grid search and genetic algorithm to improve classification performance. *Telkomnika*, *14*(4), p.1502.

[6] Ghosh, S., 2021. The Ultimate Guide to Evaluation and Selection of Models in Machine Learning (online) *neptuneblog*. Available at: https://neptune.ai/blog/the-ultimate-guide-to-evaluation-and-selection-of-models-in-machine-learning [Accessed 19 December 2021].

[7] Amin, M.S., Chiam, Y.K. and Varathan, K.D., 2019. Identification of significant features and data mining techniques in predicting heart disease. *Telematics and Informatics*, *36*, pp.82-93.