Airline Passenger Satisfaction Classification

Krzysztof Gasienica-Bednarz
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
kgasie2@uic.edu

Jonathan Chen
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
jchen311@uic.edu

Shaan Shekhar
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
sshekh7@uic.edu

Ayna Jain
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
ajain85@uic.edu

Table 1: Project interaction

Name	NetID	Question asked to	Question answered from
Krzysztof Gasienica-Bednarz	kgasie2	10	Raymond Li
Jonathan Chen	jchen311	2	Edward Liang
Shaan Shekhar	sshekh7	3	No More Questions
Ayna Jain	ajain85	11	No More Questions

Abstract

This paper discusses the comparison between binary classifiers run on a dataset of airline passenger satisfaction levels. In particular, the paper compares five different classifiers on the data: K-Nearest Neighbor, Gaussian Naïve Bayes, Decision Tree, Random Forest, as well as Deep Learning/Neural Network. The results after running each classifier on the dataset indicate that the Random Forest classifier performed best whereas the K-Nearest Neighbor classifier performed the worst. The results of the classifiers and subsequent analysis and retrospective are presented as well.

1 Summary of problem and dataset

The problem statement revolves around the implementation of several classifiers and subsequent attempts to classify airline passenger satisfaction levels as "Satisfied" or "Neutral or Dissatisfied" based on a variety of factors. The dataset for this problem was acquired from Kaggle [1]. The acquired dataset was already split into training and test sets. The training data consisted of $103,904$ samples whereas the test data consisted of $25,976$ samples for a total of $129,880$ samples of data. The data was roughly balanced between positive and negative samples (corresponding to "Satisfied" and "Neutral or Dissatisfied", respectively) with 57% of the dataset being positive samples and 43% of the dataset being negative samples. Each sample is made up of 23 attributes pertaining to the identification of the passenger, flight information, and quality of airline services. Please see Figure 1 for a visual of the samples of training data (first nine columns) before preprocessing. As shown in the figure, the dataset allows one to see the ratings a particualr airline passenger's flight information as well as the ratings given on a variety of airline qualities. These attributes, altogether,

contribute to the passenger's ultimate satisfaction level of "Satisfied" or "Neutral or Dissatisfied". The binary classifiers run against this dataset would require the data to be preprocessed. In doing so, the "Unnamed" attribute (row number in this particular CSV file) as well as the "id" attribute were dropped as their sole purpose was determined to be solely for the identification of each passenger, which would not factor into the classification of a passenger's satisfaction levels. Additionally, many of the aforementioned classifiers require the data to be of a numerical format. Therefore, unless already numerical, the dataset was transformed using an encoder to map each attribute's values to a unique integer for its respective column ("Flight Distance", for example, would not have to be encoded as an integer). Please see Figure 2 for a visual of the samples of training data (first nine columns) after preprocessing.

Figure 1: Five samples of training data (nine of twenty-four columns) before preprocessing

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3

Figure 2: Five samples of training data (nine of twenty-four columns) after preprocessing

	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service
0	1	0	13	1	2	460	3
1	1	1	25	0	0	235	3
2	0	0	26	0	0	1142	2
3	0	0	25	0	0	562	2
4	1	0	61	0	0	214	3

2 Approaches and evaluation measures

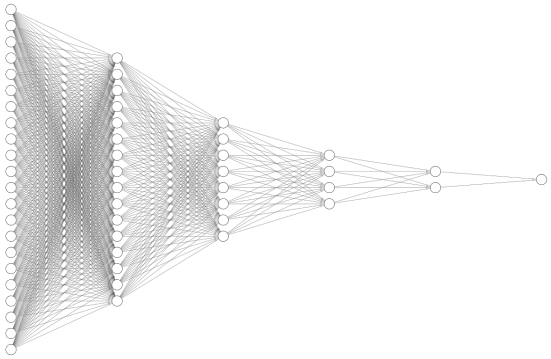
As mentioned before, five different classifiers were implemented and compared against each other for the task of binary classification of the dataset. The K-Nearest Neighbors classifier was run with hyperparameters of $(n = 9)$ neighbors. The Gausian Naïve Bayes classifier was run with the default hyperparameters (no priors, $1e - 9$ smoothing). The Decision Tree classifier was run using Gini impurity score as the splitting criterion with no max depth to the decision tree. The Random Forest classifier was run with hyperparameters of $(n = 100)$ number of estimators (trees in the forest), Gini impority score as the splitting criterion, and no max depth to each of the decision trees. The aim of this was to make the Random Forest classifier similar (in structure) to the Decision Tree classifier to see the change in performance and accuracy between each of the two models. Finally, the Deep Learning model was run using a $(22-16-8-4-2-1)$ architecture for $(n = 100)$ epochs. Please see Figure 3 for a visual of the classifier's architecture.

Each of the layers was densely connected and used a rectified linear unit (ReLU) function as the activation function. The final output layer used a sigmoid function as the activation function for the binary classification. For each of the classifiers, multiple trials were run with varying hyperparameter values to determine which values produced the best output. Thus, the aforementioned hyperparameters are the result of many trial-and-error attempts to produce the best model and metrics. For the Deep Learning classifier, model accuracy and model loss (as well as overfitting of test set) were taken into account when deciding on an architecture for best performance, both in terms of speed as well as correct classification. To keep evaluation measures standard across all classifiers, a confusion matrix (along with accompanying metrics) was used as well as the model learning speed (in seconds). This allowed for a direct comparison among classifiers using the metrics output from the learned classification model based on the predicted labels for the test set.

3 Results

As mentioned before, for each of the classifiers, the performance of the classifier was visualized using a confusion matrix as well as accompanying metrics: accuracy, precision, recall, and F1 score. Please see Figure 4 above for the consfusion matricies and their accompanying metrics. For the

Figure 3: Deep Learning (22-16-8-4-2-1) architecture



purposes of measuring the classifiers' performance; accuracy, F1 score, and speed were primarily used. Please see Table 2 below for these metrics. Based on the resulting metrics, out of the five classifiers, it was determined that the Random Forest classifier performed the best and the K-Nearest Neighbor classifier performed the worst.

Table 2: Classifier comparison

Classifier Name	**Accuracy**	**F1 Score**	**Speed**
K-Nearest Neighbor	0.673	0.742	3.913s
Gaussian Naïve Bayes	0.864	0.834	0.054s
Decision Tree	0.908	0.893	0.767s
Random Forest	0.949	0.940	13.683s
Deep Learning	0.928	0.920	55.090s

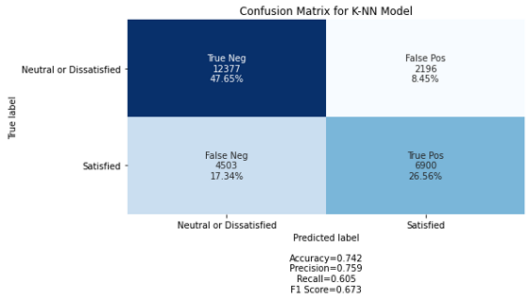
In the case of the K-Nearest Neighbor classifier, it is obvious that this classifier performed the worst. The classifier had subpar performance in just about every metric, save for speed in some cases. However, the best performing classifer was not as clear cut. Three out of the five classifiers scored greater than 90% on accuracy as well as two scored above .9 on the F1 Score. The decision to declare the Random Forest classifier was based on a number of factors, but most importantly the Random Forest classifier was best at predicting the correct class of the airline passenger's satisfaction levels with minimal error. It could be argued that the Decision Tree classifier performed better than the Random Forest classifier if more weight was put on the speed at which the model learns. If speed is more heavily weighed than Accuracy or F1 Score, for example, then the argument for the Decision Tree model performing the best could definitely hold water. As a result, the "best" classifier is subject to the requirements put forth as to what determines the "best" model (can the model afford to make False Positives or False Negatives, is model speed - both learning and predicting - crucial; etc.).

4 Retrospective and lessons learned

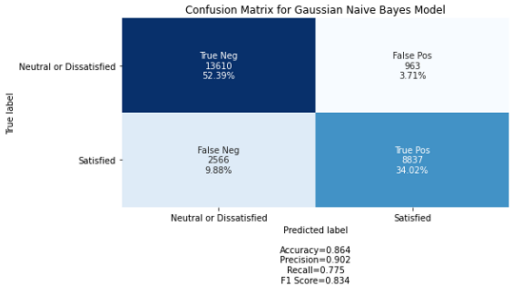
4.1 Retrospective

There were quite a few things that could have been changed in the implementation and use of the classifers in this paper. An important change came to light during the peer review process of receiving questions about the course project presentation via VoiceThread. One peer questioned the feature importance of Age and Gender and why such features were considered for the classifiers (particularly Decision Tree and Random Forest). Please see Figure 5 below for the visualization of

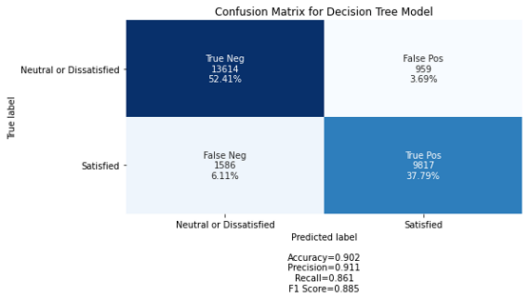
Figure 4: Classifier confusion matricies



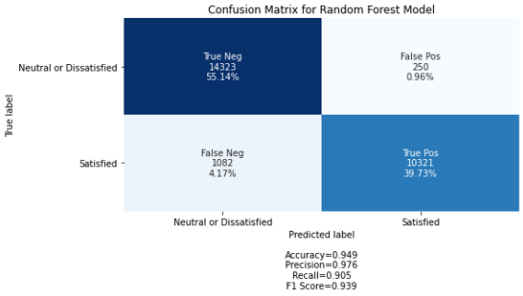
(a) K-Nearest Neighbors confusion matrix



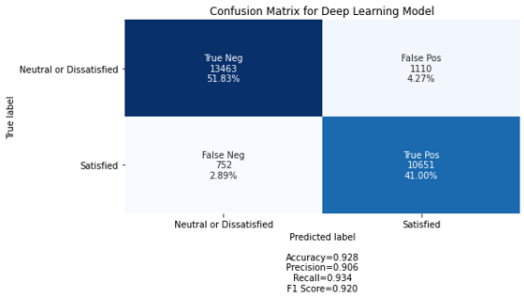
(b) Gaussian Naïve Bayes confusion matrix



(c) Decision Tree confusion matrix



(d) Random Forest confusion matrix



(e) Deep Learning confusion matrix

the feature importance of the dataset. This question brought up two good points. The first being: why have features been included in the processed dataset which are seemingly non-airline related? The second being: why is Age so high up on the feature importance graph? To address the first point, one perspective could be that it can be reasonably assumed that features such as Age or Gender don't *have* to be included and we could think of each airline passenger as an anonymized data point. Although, to refute this, it could also be said that it is not guaranteed that the attributes of each sample are independent of each other. Meaning, Age and/or Gender could, in fact, affect the decision making and classification of whether or not a passender is satisfied with the airline. The second point ties in closely to the first. Logically, Age is high up on the feature importance graph due to how the decision tree training process works and how feature importance is determined. To speculate, the reason Age is more important than other (seemingly more "important" airline services) could be due to how the Random Forest classifier and its 100 decision trees split on the Age attribute and determined the Gini impurity score of the attribute. That is to say, splitting on Age resulted in a lower impurity score than some of the other attributes leading to higher feature importance (and easier/better classification of the dataset).

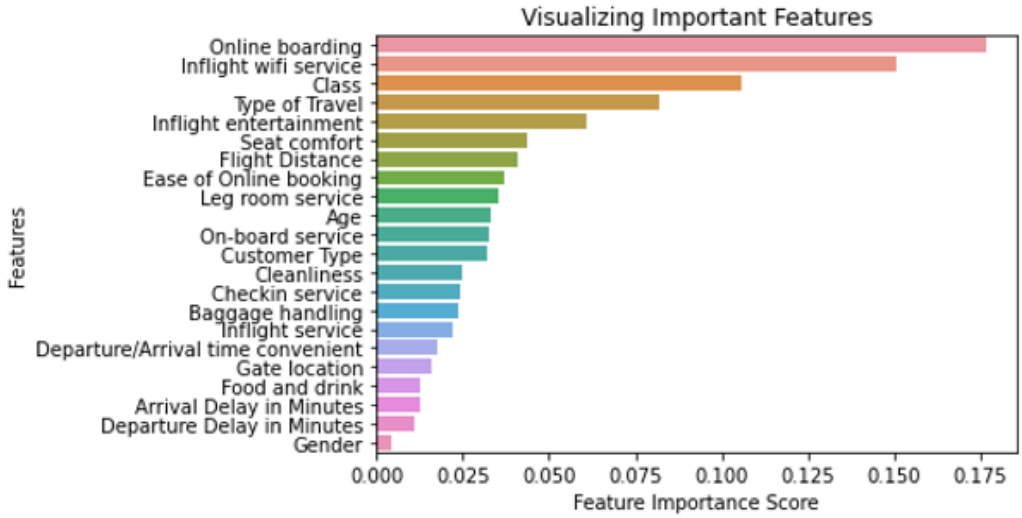
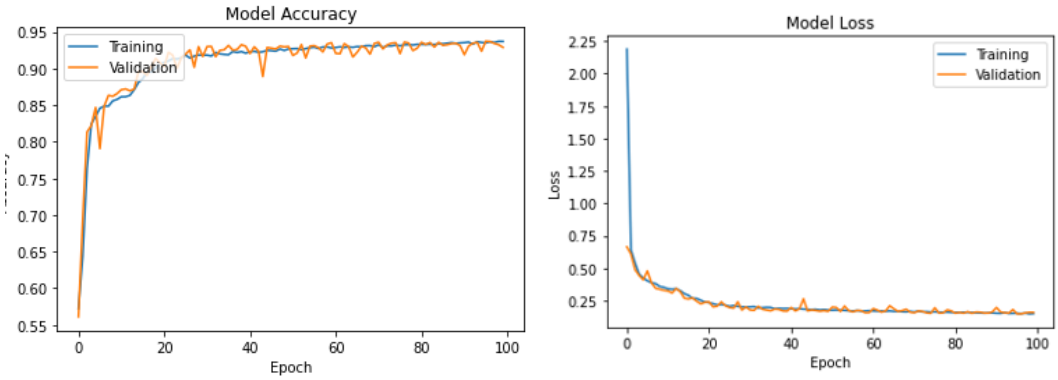


Figure 5: Random forest feature importance.

Finally, another change which could have had some impact on model performance is the number of epochs in the Deep Learning model. The performance of the model presented in this report uses $(n = 100)$ as the number of epochs for training. This can be seen as unnecessarily large for the resulting model. As seen in Figure 6, the model's accuracy increase and loss decrease diminishes, almost asymptotically, at around 50 epochs. It was observed to behave similarly across most trials in attempting to find the best hyperparameters for the model. However, since the model was still performing objectively worse than the Random Forest and Decision Tree models, the increase in epochs led to an ever so miniscule increase in model accuracy. This, however, came at a large drop in performance when it came to speed and, as a result, the number of epochs can now be deemed too high for future model improvement or experimentation.

Figure 6: Deep learning classifier metrics



(a) Deep learning model accuracy

(b) Deep learning model loss

4.2 Lessons learned

There are many key lessons that can be learned from this paper, especially as novice computer scientists/machine learning engineers. The most important lesson being that not all classifiers are created equal and the problem at hand will determine which classifier should be used based on its strengths and weaknesses. Additionally, it was observed that hyperparameters play a significantly large role in the performance of a model, especially when it comes to Deep Learning classifiers. Performance of the Deep Learning model changed with almost every fine change in hyperparameters - from batch

size and how many layers (as well as their size and structure), to activation functions and the number of epochs. Ultimately, although the Deep Learning classifier performed worse than the Random Forest and Decision Tree classifiers, with more experimentation it is believed that the Deep Learning classifier could perform just as well as, if not better than, the aforementioned classifiers. That being said, given more time to experiment, all of the models presented could likely perform better if each of the models' hyperparameters were tuned some more to better fit the data (keeping overfitting in mind). Along with tuning the hyperparameters, it is also important to note the size of the dataset. The training and test set are by no means large (less than 15MB total). With a larger dataset in mind, it can be inferred that each model could potentially learn the relationships among the data and perform better. As mentioned in the retrospective above, it is also very important to consider the data being used by the model and whether or not the selected input features are relevant or make an impact on the overall model efficiency. Data (pre-)processing, classifier selection, hyperparameter tuning, and interpretation of results all play a large role in the way various machine learning techniques can be applied to tackle a problem.

References

[1] Klein, TJ. (2020). Airline Passenger Satisfaction, Version 1. Retrieved March 16, 2021 from https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction.