CS 221 Final Project Report

Arjun Jain, Devanshu Ladsaria, Ishan Mehta

See code, data, and video

Introduction: With increasing globalization, people are flying more than ever. As a result, airlines strive to create the perfect flying experience for their passengers. There are a lot of factors that airlines have to consider to ensure their passengers are satisfied with their flying experience: flight duration, inflight service, ease of booking, etc. With this project, our goal is to see whether we can predict passenger satisfaction based on these factors.

Literature Review: There are a few related studies that have been conducted in this field. In terms of airline passenger satisfaction, J.D. Power conducted a study where they concluded that passenger satisfaction has been steadily dropping over the last few years. They used a 1000 point scale to rank various airlines and have found that JetBlue ranks highest in customer satisfaction with consistent scores sitting around 878. The study measures performance and satisfaction by cabin class. JD Power grouped the report into three categories: First/Business, Premium Economy and Economy. They claim, "The modified factor model measures passenger satisfaction with airline carriers based on performance in seven factors: Reservation, Check-In, Boarding, Baggage, Aircraft, Flight Crew, In-Flight Services, Costs & Fees."

There is also a study published in the Science Direct journal which explored more of the market factors that have affected airline satisfaction in recent years to help explain some of the trends in the industry. Most of the data in this study was collected via questionnaire. Passengers were sampled and asked to rank factors as far as importance in satisfaction, but were also asked to name qualitative observations regarding satisfaction. Many answers talked about better hygiene and service on flights.

For more technical works, there is a study published in Nature which uses RF-RFE-LR modeling to predict airline passenger satisfaction and determine which factors are most important to increase this satisfaction. RF-RFE-LR stands for "recursive feature elimination based on a random forest." In summary, the article claims, "RF adopts the method of random selection for each node attribute set in the decision tree, first randomly selects an attribute subset from all attributes, and then selects an optimal attribute from the subset. Therefore, based on the sample disturbance brought by bagging, the RF further introduces attribute disturbance, which increases the generalization performance of the integration." With a more mathematical and recursive approach to finding the attributes that are most related to airline satisfaction, this study guides some of our thinking and modeling of the problem at large. Utilizing random attributes as a methodology to keep numerous doors open and not limit the primary factors of satisfied passengers is a very intelligent and unique approach to the issue.

Dataset: For this classification problem, we used the following dataset from Kaggle: https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction. This dataset is based on an airline passenger satisfaction survey. It includes a testing dataset (26k examples) and training dataset (104k examples).

For each example, the output is the airline satisfaction level and has a value of either "satisfied" or "not satisfied". There are 22 features, most of which are satisfaction rankings from 1-5 of various factors such as inflight service, convenience of departure/arrival times, ease of online booking, leg room, etc. Other features that are not satisfaction rankings are flight distance, departure delay, arrival delay, age, gender, customer type, and travel class.

For pre-processing the dataset, we loaded the training and testing data into pandas dataframes. Then, for the output column, we coded "satisfied" as 1 and "not satisfied" as 0. From there, we coded all non-quantitative features (gender, customer type, travel class) into integer values. After that, we removed any rows with missing or NA values. The preprocessing code is in data_cleaning.py and our datasets (original and cleaned) are here.

Main Approach: For our main approach, we applied three different models to the dataset and compared their performance: logistic regression, Naive Bayes, and K-means clustering.

> **Logistic Regression**: We trained a logistic regression model to map input features for a passenger to a probability value between 0 and 1, representing the likelihood of the passenger being satisfied. If this probability was greater than a threshold of 0.5, the model classified the training instance as "satisfied" otherwise it was classified as "not satisfied". Training was performed with gradient descent (1000 iterations, step size = 0.01).
>
> The model was implemented from scratch without any libraries (see log_reg.py).
>
> **Naive Bayes**: We trained a Naive Bayes classifier to learn the probability distribution of features and the conditional probability of features given class labels of "satisfied" or "not satisfied". When given a new input, the classifier calculates the probability of each label based on the features and predicts the label with the highest probability. Like logistic regression, Naive Bayes was implemented from scratch without any libraries (see naiveBayes.py)
>
> **K-means Clustering**: We wanted to explore whether unsupervised learning will perform better or worse than supervised learning. To do so, we ran k-means clustering with k=2 on both the training and testing data. Given our time constraints, we used scikit-learn to run k-means on the data (see kcluster.py). However, we made sure to understand how scikit-learn ran k-means clustering on our dataset. First, k-means clustering (k=2) ran on the training data with the class labels ("satisfied", "not satisfied") removed. This partitioned the training data into two clusters and labeled each cluster with the majority class. Given the centroids of these clusters, the code then proceeded to find the nearest

centroid for each test instance and assign it the class label of this centroid. Finally, it calculated accuracy by computing the proportion of correctly classified test instances.

In summary, our main approach involves applying three different models (logistic regression, k-means clustering, and Naïve Bayes) to classify passenger satisfaction. We compare these algorithms on the same dataset to see which would be more successful in accurately classifying airline passenger satisfaction.

As part of our approach, we also experimented with removing different features and seeing how the accuracy would change. Specifically, we decided to remove the following features: Gender, Age, Flight Distance, Arrival Delay, Departure Delay, Gate Location. We decided to remove flight distance, arrival delay, and gate location because these factors are not under the control of the airline themselves and should not be considered when determining airline passenger satisfaction. As for gender and age, we removed them because they can result in skewed results and can bias the classification results.

All code for our project can be found here.

Evaluation Metric: Given the context of this problem, we decided to use accuracy as our evaluation metric. Accuracy is the proportion of correctly classified examples out of all examples (both "satisfied" and "not satisfied"). Accuracy is an important metric because it measures the overall correctness of the model's predictions.

Baseline Model: Given that this is a binary classification problem, we decided that a starting baseline model can be to predict the most frequent class in the training data for all observations. The implementation of this model was very straightforward. We counted the most frequent label in our training data frame and found that the most common prediction was "not satisfied." Then for our training dataset, we predicted "not satisfied" for all the instances, and this gave us an accuracy of 55% percent. This ended up being better than a random baseline predictor.

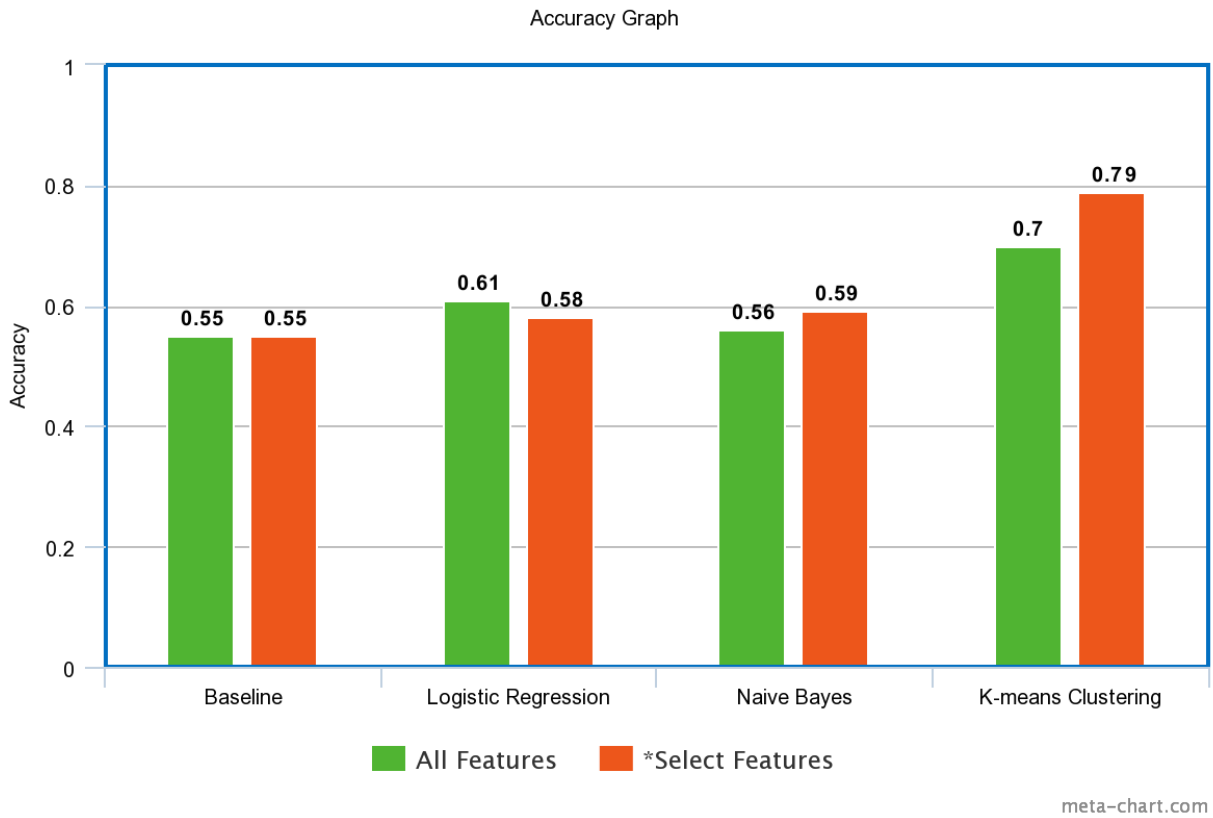
Results: Below is a table which shows each of our implemented models and the accuracy with all input features included and with certain features removed. Each model was run on the same training and testing dataset.

Model	Accuracy (all features)	Accuracy (*w/ removed features)
Baseline model	0.55	0.55
Logistic Regression	0.61	0.58
Naive Bayes	0.56	0.59
K-means clustering (k=2)	0.70	0.79

*Removed features: Gender, Age, Flight Distance, Arrival/Departure Delay, Gate Location

Table 1. Accuracy of Passenger Satisfaction Classification

t



Graph 1. Accuracy of Passenger Satisfaction Classification

Analysis:

The baseline model achieved an accuracy of 55% which reflects the proportion of test examples of the majority class (Table 1). This baseline accuracy provides a reference point for evaluating the effectiveness of the subsequent models.

Logistic Regression achieved an accuracy of 61% when run with all input features included in the training and testing datasets. As we expected, logistic regression outperformed the baseline model. Unlike the baseline model, logistic regression is able to capture relationships between the input features and the target variable, enabling it to make more accurate predictions compared to the baseline. When we ran logistic regression with certain features removed from the datasets, the accuracy decreased by 3% (Graph 1). This drop in accuracy may have occurred because the removed features had strong association with the output variable and removing them negatively impacted the ability of the logistic regression model to accurately classify instances. While these features (e.g, flight delay, gate location) may not be factors that airlines can control, they do play a role in how satisfied passengers are with their flying experience.

Naive Bayes performed poorly and only achieved an accuracy of 56% when run with all input features (Table 1). Compared to the baseline, Naive Bayes barely achieved a higher accuracy. This low accuracy may have occurred because the dataset violated the “naive” assumption that all factors are independent of each other given the class label. In a study like this, it is not

difficult to see how linked the factors are, especially those such as “Age” and “Type of Travel” with many younger passengers always defaulting towards economy. With the data violating the “naive” assumption, Naive Bayes inaccurately classified the airline satisfaction of many passengers. However, we did observe a 3% increase in accuracy when running on a dataset with certain features removed. This increase in accuracy makes sense because we removed features such as “Age” and “Gender” that are highly interlinked with features that remained in the dataset such as “Travel Class” and “Customer Type”.

K-means achieved the highest accuracy (70%) compared to the three other classifiers and was the strongest predictor. Unlike logistic regression and naive bayes which assume linear relationships between features and target variable, K-means is not constrained by linearity assumptions. Based on the superior results of K-means, it seems like the data had non-linear relationships and patterns that K-means was able to effectively capture while logistic regression and naive bayes were unable to do so. As a result, K-means was much more successful in classifying instances compared to these other two models.

Error Analysis:

To see how efficient our implementation of the algorithms were, we decided to run the scikit library’s version of Naive Bayes and Logistic Regression. In both cases, it came up with numbers that were slightly higher than ours, with Naive Bayes coming in at 64% and Logistic Regression just under 70%. We understood that logistic regression was still a better algorithm for this dataset and we were able to show that through our implementation, but we did struggle in optimizing our algorithms to the level that scikit did.

In the vain of attempting to find the error, this might require tracing through the exact code and seeing where certain broad generalizations or defaulting is done and how that can be corrected. As far as K-means, we utilized the library to implement the algorithm. There is no way to know what errors lie within the library implementation, but a simple step could be attempting to implement the algorithm by hand and comparing accuracies to see if our implementation would exceed that of the library.

We have also attached an error chart here which shows exactly how many examples from each class we classified correctly and incorrectly.

Class	% Correctly Classified	% Incorrectly Classified
Class 0 (Not Satisfied)	73.5%	26.5%
Class 1 (Satisfied)	47.8%	52.2%r
Overall	61.5%	38.5%

Table 2: Error Analysis For Logistic Regression

Looking at this error table for logistic regression, it seems as though there is a clear difference in the ability for the algorithm to classify between Class 0 and Class 1. Specifically, the algorithm does more than 1.5 times better with Class 0 and does less than 50% with class 1. Now, since this algorithm is attempting to optimize the theta values in Class 1, it is clear that there is potentially some error in how the algorithm is handling the code. This doesn't mean that the algorithm isn't working but it is probably inefficient in finding and calculating values and perhaps incorrect in some of the classifications due to a vague threshold or computational value.

When we think about remedies for this chart, it seems as though we need to work on helping the algorithm recognize satisfied passengers. This may mean putting more weight on some of the parameters that are highly related to satisfaction and maybe even adding in bias terms or other ways of pushing the algorithm to be more likely to select satisfied as the option. We could also lower our threshold for the algorithm from 0.5 to 0.4 or 0.35 as a way to get more passengers to be satisfied. These types of methods of altering hyperparameters or training an algorithm specific to a dataset can be tricky however because the dataset is large and nonlinear.

Overall, it does seem as though there may be some error in our logistic regression algorithm which we can work to fix.

	Training Data	**Testing Data**
Accuracy	89.2 %	79.2 %

Table 3: K-Means Accuracy

This table, although simplistic, does help answer some questions about errors in the algorithm. We see that the training data is being classified at 89.2% which proves that the model is not overfitting for the training data. However, running the algorithm on examples it has already seen before should yield an accuracy higher than 89.2%. For this reason, we also see the possibility of some error in the algorithm. The most likely scenario is the algorithm is unable to optimize certain parameters or is miscalculating a few factors which then leads to some inaccuracies in the training data.

Ideally, we want to increase our training data to about 95% which would keep us from overfitting but also increases the competency of the algorithm. That increase in training accuracy would reflect positively on the testing data as well which would increase to the low or mid-80%.

Future Work:

Finding the correct hyperparameters: We tried a lot of different hyperparameters for our code with varying levels of success, but if we had more time, tweaking it more would have yielded better results. For example, in our logistic regression algorithm, our lower accuracy may also have something to do with the 0.5 threshold we employed. Perhaps a slightly higher threshold would require more certainty before classification which may help our case. In addition, our step size initially was set to 0.001 which made the code run extremely slowly but gave us better numbers than when we increased the step size to 0.01 or 0.1. This is a parameter we can change to find the tradeoff between accuracy and speed.

Finding more correlations: We had a lot of features (over 20), some we didn't include like "Gender" to prevent the algorithm from having unwanted biases. It will be really interesting to see how these different features interact with each other, and we can definitely find surprising results and correlation that we didn't think of before. We isolated some of the features and reported the accuracy above but more of these combinations would be a great next step to take.

Optimization: A lot of our learning algorithms were written from scratch, without the use of machine learning packages. We found some difference between our value and the value returned by the packages. This suggests that we are leaving some accuracy on the table, it can be a nice project to try to get our accuracy match with the accuracy from the packages.

Challenge: Especially for logistic regression, our running time was astronomical, maybe precomputing certain values, and other optimizations can drastically reduce this time. This may have a lot to do with our chosen hyperparameters but also with the machines we were running the code on. Perhaps writing code that is more optimized for certain hardware or running the code on stronger and faster machines may also help our running time and efficiency as a whole.

Different Approach: Our data was stored as panda dataframes and numpy arrays. It can be interesting to use these algorithms using dictionary and feature vectors, instead of arrays, and see how the runtime changes. In addition, we could alter the approach to solving this problem by trying a few different machine learning and AI algorithms and seeing if they come up with different numbers. Now that we have learned that unsupervised learning can accurate classifications with this specific dataset, employing more unsupervised algorithms could be the next best step for this study.

References:

Zahraee Seyed, Shiwakoti Nirajan, Jiang Hongwei, Qi Zhuoqun, He Yunfeng, Guo Tianan, Li Yifeng, "A Study on Airline's Responses and Customer Satisfaction During the COVID-19 Pandemic," International Journal of Transportation Science and Technology, December 7, 2022, https://www.sciencedirect.com/science/article/pii/S2046043022001009

Xuchu Jiang, Ying Zhang, Ying Li, Biao Zhang, "Forecast and Analysis of Aircraft Passenger Satisfaction Based on RF-RFE-LR Model," Scientific Reports, Nature, July 1, 2022, https://www.nature.com/articles/s41598-022-14566-3

Geno Effler, "North American Airline Passenger Satisfaction Declines: Here's Why That's Good News, Says J.D. Power," J.D. Power, May 11, 2022, https://www.jdpower.com/business/press-releases/2022-north-america-airline-satisfaction-study#:~:text=Overall%20passenger%20satisfaction%20declines%20sharply,points%20from%20a%20year%20ago

TJ Klein, "Airline Passenger Satisfaction," 2020, https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction