Airplane Passenger Satisfaction Investigation

CS6140 Final Report

Christopher Daly, Edgar Alan David and Jacqueline Girouard

Northeastern University

Boston, MA, USA

Abstract - The machine learning field can be briefly described as enabling machines to make sound decisions using previous experiences. This has exhibited an impressive development in recent years due to the increased processing power of computers. As this field expands, it is essential to understand the basics of machine learning. This paper report discusses the findings and motivations from the final project for CS6140 Machine Learning. The dissatisfied customers, advancement of air travel and increase in people mobility have led to the development of sophisticated machine learning algorithms to improve the experience. The report details the motivation for investigating airport satisfaction data, analysis on dataset features and an investigation into model analysis via the various tools introduced to us over the course of the summer 2022 semester. We first provide the introduction of this report and provide a brief background of the motivation. Then the data and implementation are discussed to review the fundamental concepts for machine learning using SVM, Naïve Bayes and Logistic Regression. Then we point out the results of designing machine learning experiments and their performance evaluation. Finally, we provide the group's conclusion for this final report.

Keywords: Machine Learning, Support Vector Machines, Feature Analysis, Naïve Bayes, Logistic Regression, Kernel

I. INTRODUCTION

As the world slowly emerges from the Covid-19 pandemic, travel is reentering the lives of millions of Americans. How can airline companies capitalize on this return to travel and make the experience as positive as possible so industry can quickly recover? We are going to use machine learning, particularly the support vector machines (SVM) method, to explore what makes an airline trip positive and satisfying. This paper explores what key features an airline should invest in and use for prediction and qualify their relationship between each other. It is essential to understand the needs and comfort level of a multitude of customers. Therefore, customer feedback is the key for any airline industry to prosper and advance. There are multitude of possible ways of collecting customer response and one traditional way is getting feedback during the actual journey. But this method sometimes is not appealing to customer by filling in feedback forms. Another form of feedback is through using the online website or mobile application during booking of a trip or after completion of a journey, an email with the link for feedback can be sent to the passenger. One better approach is to use social media like Facebook and Twitter to request feedback.

The customers are asked to rate (1 for poor and 5 for excellent) the experience and service. Because of the enormous amount of data available, machine learning made it possible to analyze these huge datasets to highly predict and classify the models. In this report, machine learning techniques such as SVM are used to analyze the class satisfaction sentiment classification model for the survey data provided.

II. DATA

The public "Airline Passenger Satisfaction" dataset from Kaggle was utilized for this investigation (1). The data set contained 25 possible features and over 1500 data entries for a robust training and test set (1). Data was encoded as integer values. Missing values were placed in median values or class means. Each column on the dataset will be the identified features for the customer feedback and the label would be the column "satisfaction". The label will be either satisfied or dissatisfied(neutral) for the customer response. Most of the features were rated from 1 as poor to 5 for excellent. Other features that were continuous namely age, flight distance, departure delay and arrival delay are measured. Using this data can determine factors that are highly correlated to a satisfied customer.

III. IMPLEMENTATION

To effectively track the progress of our SVM models, we decided to initially train two baseline models: a Naïve Bayes and a Logistic Regression Model. We chose Naïve Bayes and Logistic Regression to get our initial baseline results because they're fast and easy to use. These models are the "go to examples" for generative and discriminative classifiers, respectively.

A. *Naïve Bayes Classifier*

Naïve bayes uses Bayes theorem to make predictions based on training set data (3). By using this model, we are establishing a baseline with simple probabilistic calculations.

B. *Logistic Regression*

Logistic regression produces a probability of a linear combination of input variables belonging to a class, and then with a threshold value, determines which class a result belongs to (4). Using logistic regression provides an alternative means of calculating baseline results with an approach which is more resistant to potentially correlated features.

C. *Support Vector Machines*

In order to investigate various decision boundaries, scikit-learn was utilized to implement a SVM with linear, polynomial and radial-basis-function kernels (2). SVMs can be described as a "hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks" (5). By transforming the feature-space through kernels, using the Kernel-Trick, SVMs can be applied to various decision boundary shapes. Various visualizations were presented to illustrate the overall results.

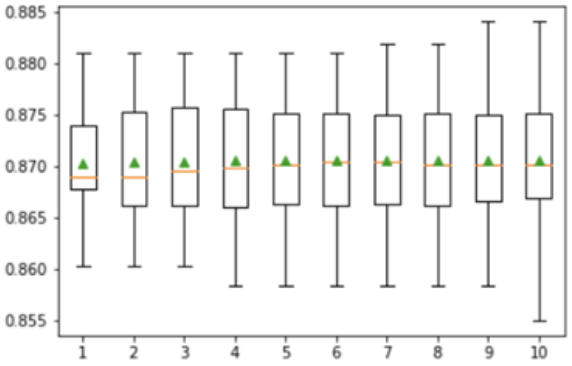
IV. RESULTS

A Naïve Bayes implementation which treated 0-5 ratings as categorical features resulted in the following prediction statistics: Accuracy 0.886, Precision 0.8968692449355433, Recall 0.8935779816513761 and F1 score 0.8952205882352942. We used these

metrics as a baseline performance for analysis.

The Logistic Regression model was able to take the data prepared with the integer encodings to perform training on the same features. A 10-fold cross validation was performed to reveal an accuracy of 0.8735 (Figure 1). Logistic Regression assumes that "the variable follows a binomial distribution of the linear combination of the input variables" which may not be true of all features used during training (4). This is one possible explanation for a slightly worse performance than Naïve Bayes. An alternative explanation may be due to overfitting the model.

Figure 1: Training Results Across 10-Fold Validation for Logistic Regression Model



With baseline results from Naïve Bayes and Logistic Regression combined, we can establish a baseline expectation of at least 0.88 accuracy.

We began our analysis utilizing SVMs with Linear Kernels. This linear boundary was explored through 2, 3 and all features. The linear boundary (Figure 2), was extremely similar across feature combinations for 2 feature experimentation. This, linked with similarly low accuracy results ranging from 68-80% which were significantly below the baseline, led us to suspect high correlation between features (Table 1). High correlation between features suggests a potentially dependent relationship between features, breaking the underlying premise of many of the models we intended to investigate. Similar suspicions continued when investigating the 3-feature combination space as accuracy results were not significantly improved upon (Table 2).

Figure 2: 2 Feature Linear Boundary

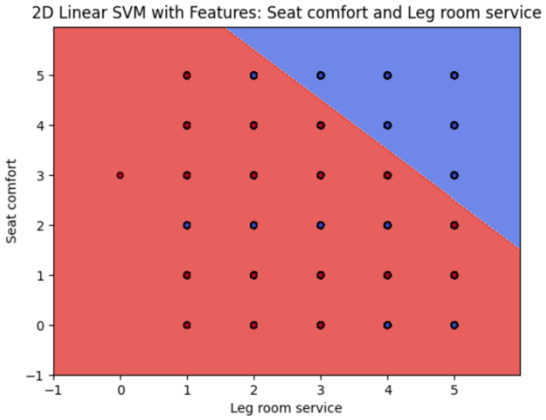


Table 1: 2 Feature Linear SVM

Features	Accuracy	F1
Online Boarding and Inflight Service	0.8804	0.8304
Inflight Entertainment and Online boarding	0.8109	0.8396
Seat Comfort and Online Boarding	0.7882	0.8006
Inflight Entertainment and Leg Room Service	0.7155	0.7457

Table 2: 3 Feature Linear SVM

```
Results using kernel: linear
[[13006  1522]
 [ 2568  8797]]
              precision    recall  f1-score   support

           0       0.84      0.90      0.86     14528
           1       0.85      0.77      0.81     11365

    accuracy                           0.84     25893
   macro avg       0.84      0.83      0.84     25893
weighted avg       0.84      0.84      0.84     25893

Accuracy score: 84.20422508013749%
Precision score: 85.25050877022967%
Recall score: 77.40431148262209%
execution time: 18.112325429916382 seconds
```

When using all features in linear kernels, the best trained model resulted in an accuracy of 0.873 (Table 3), which approached our baseline but did not improve upon it. An interesting note from our data revealed that decreasing the slack variable C for values 0.1, 1, 10 and 100 did not assist in improving the performance. Accuracy remained approximately 0.873 across all these experiments. It is likely that the solution could not be encapsulated by a linear decision boundary and the SVM constraints were not the limiting factor.

Table 3: All Features Linear SVM

```
Results using kernel: linear
[[13233  1295]
 [ 2003  9362]]
              precision    recall  f1-score   support

           0       0.87      0.91      0.89     14528
           1       0.88      0.82      0.85     11365

    accuracy                           0.87     25893
   macro avg       0.87      0.87      0.87     25893
weighted avg       0.87      0.87      0.87     25893

Accuracy score: 87.26296682501062%
Precision score: 87.84836257858684%
Recall score: 82.37571491421029%
execution time: 42.69868588447571 seconds
```

Our suspicion of the limiting nature a linear boundary placed on the model was confirmed by our investigation into polynomial and RBF Kernels. Polynomial kernels performed with an average of 0.928 and RBF kernels performed with an average of 0.947 (Table 4, Table 5).

Table 4: All Features Polynomial SVM

```
Results using kernel: poly
[[13802   726]
 [ 1138 10227]]
              precision    recall  f1-score   support

           0       0.92      0.95      0.94     14528
           1       0.93      0.90      0.92     11365

    accuracy                           0.93     25893
   macro avg       0.93      0.92      0.93     25893
weighted avg       0.93      0.93      0.93     25893

Accuracy score: 92.80114316610667%
Precision score: 93.37167899205697%
Recall score: 89.98680158380994%
execution time: 18.417749643325806 seconds
```

Table 5: All Features RBF SVM

```
Results using kernel: rbf
[[13985   543]
 [  828 10537]]
              precision    recall  f1-score   support

           0       0.94      0.96      0.95     14528
           1       0.95      0.93      0.94     11365

    accuracy                           0.95     25893
   macro avg       0.95      0.94      0.95     25893
weighted avg       0.95      0.95      0.95     25893

Accuracy score: 94.70513266133705%
Precision score: 95.09927797833934%
Recall score: 92.71447426308843%
execution time: 23.1636700630188 seconds
```

Despite the success of the RBF Kernel model, our suspicions of dependent data led us to calculate a correlation heat map (Figure 3 – See Extension Due to Figure Size). This heat map certainly revealed correlation possibilities which made intuitive sense as humans reviewing the map. For example, late departure and late arrival times were highly correlated. Both features described “lateness”. However, most features were observed to be relatively uncorrelated. Fears developed of potentially dependent data were put at ease when it was observed that reducing the feature space results in worse performance by the RBF and Polynomial kernels, leading us to conclude that the linear decision boundary was simply insufficient (Table 6, Table 7).

Table 6: 3 Features Polynomial SVM

```
Results using kernel: poly
[[12635  1893]
 [ 1277 10088]]
              precision    recall  f1-score   support

           0       0.91      0.87      0.89     14528
           1       0.84      0.89      0.86     11365

    accuracy                           0.88     25893
   macro avg       0.88      0.88      0.88     25893
weighted avg       0.88      0.88      0.88     25893

Accuracy score: 87.75730892519213%
Precision score: 84.1999833069026%
Recall score: 88.76374835019799%
execution time: 28.74098491668701 seconds
```

Table 7: 3 Features RBF SVM

```
Results using kernel: rbf
[[12979  1549]
 [ 1365 10000]]
              precision    recall  f1-score   support

           0       0.90      0.89      0.90     14528
           1       0.87      0.88      0.87     11365

    accuracy                           0.89     25893
   macro avg       0.89      0.89      0.89     25893
weighted avg       0.89      0.89      0.89     25893

Accuracy score: 88.74599312555517%
Precision score: 86.58758334054897%
Recall score: 87.98944126704795%
execution time: 18.844109535217285 seconds
```

V. CONCLUSIONS

Radial Basis Kernels proved to be the most successful model, beating our baseline by an impressive 5%. When performing a Grid Search to select the top 3 most informative features via a Random Forest Classifier for a feature reduced model, loss of accuracy was observed. This, coupled with analysis of a feature correlation map, dissuaded fears of potentially dependent features undermining the mathematical basis of the support vector machines. By examining various slack variables for a linear kernel, and observing consistently poor performance, we feel confident in saying that a simplified linear boundary is not sufficient in explaining the complexity of human airport satisfaction.

VI. CODE

Code Demonstrating Final Report Implementation:

https://colab.research.google.com/drive/1VxkEdSBFwLMZ-ZQmSWDgF0CtvVldpCcF?usp=sharing%20%3Chttps%3A%2F%2Fteams.microsoft.com%2Fl%2Fmessage%2F19%3AaTgtCVvF8cPjUgaHGFAFNB2Qtw8zZ7pplKu1QD0ZW341%40thread.tacv2%2F1659814472807%3FtenantId%3Da8eec281-aaa3-4dae-ac9b-9a398b9215e7&amp%3BgroupId=b5164f50-627e-42b1-a825-3fd889255232&amp%3BparentMessageId=1659808866245&amp%3BteamName=CS6140%20Team%202&amp%3BchannelName=General&amp%3BcreatedTime=1659814472807%3E

GitHub Repo:

https://github.com/erdavid8/CS6140-Final-Project-Summer2022

VII. ACKNOWLEDGEMENTS

Thank you to Professor Ahmad for a great course on Machine Learning!

VIII. REFERENCES

[1] Klein, TJ. Airline Passenger Satisfaction, Retrieved July 2022 from https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction.

[2] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Course Notes:

[3] Ahmed, Bilal, "The Naïve Bayes Model", CS6140 Machine Learning Course Notes, Jan. 26, 2015.

[4] Ahmed, Bilal, "Logistic Regression", CS6140 Machine Learning Course Notes

[5] Ahmed, Bilal, "Support Vector Machines I", CS6140 Machine Learning Course Notes, Jan. 26, 2015.

[6] Ahmed, Bilal, "Support Vector Machines III", CS6140 Machine Learning Course Notes, Jan. 26, 2015.

[7] Pavlu, Virgil, "Kernels", CS6140 Machine Learning Course Notes, Nov. 30, 2014

IX. EXTENSION

Figure 3: Correlation Heat Map of Features