Mini Project 1

Machine learning (CS582), MIU [1]

Baraa Mousa Noufal Ivan Krasowski Bissio

July 23, 2021

[1] Instructor: Anthony Sander

Contents

Abstract

1 About the Dataset

The chosen dataset collects information about Airline Passenger Satisfaction. We opted for this dataset because it had a lot of entries (almost 130000), many columns (including 4 categorical) and it was oriented to a binary classification problem ('satisfied' vs. 'neutral or dissatisfied').

1.1 EDA: Exploratory Data Analysis

- The dataset collects multiple features related to each passenger-flight, including labels for whether the passenger was satisfied with the flight or not.
- It has 24 columns (22 + id + result).
- Id column is dropped (it would add weight).
- Observation: the only column with null values is 'arrival_delay_in_minutes', with 393 missing values.
- Categorical columns: 4

  Features and possible values: Gender (2), customer_type (2), type_of_travel (2), customer_class (3)

  Decision: use one-hot encoding in all of them (none of them is a clear candidate for being weighted)

- Looking for strong correlations: pairwise correlation function to check if two features show strong correlation.

  NOTE: 'satisfaction' (the label we will try to predict) is not strongly correlated (>0.7) with any of the other features.

  arrival_delay_in_minutes and *departure_delay_in_minutes* have the highest correlation rate (0.965291), which semantically makes sense; all other variables are less than 75% correlated.

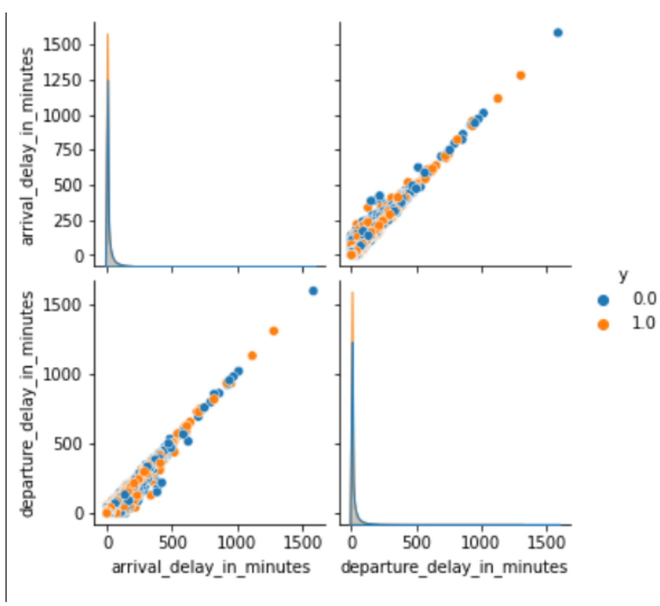

Figure 1: Correlation between most-correlated features

1.2 Data Preparation

1. X: all table except id and result ('satisfaction') columns; y: 'satisfaction' column
2. Since the arrival_delay feature is highly correlated with the departure_delay feature, and the missing values are not that many (393 out of 129879: 0.03%), we decide to remove the column.
3. We see there are 4 categorical features, with no more than 3 unique values each. So, given that none of them is a clear candidate for being weighted, we decide to use one-hot encoding in all of them.
4. Finally, we split X and y for training and validating, following a ratio of 80%/20% (the dataset is large enough).

   Variables: ***X_train, X_val, y_train, y_val***

2 Models

2.1 KNN

2.2 Decision Tree

2.3 Support Machine Vector

2.3.1 Presentation

Support Vector Machines are a model of supervised learning.
In summary, the model finds the hyperplane (kernel) that maximizes the Margin of Safety for classifying.

2.3.2 Defining Parameters

The data for training and validating is already defined by the Data Preparation step (80% train, 20% validation).
Representative parameters for the model are *gamma* and C

- gamma: kernel coefficient
- C: regularization parameter (higher C, higher variance)

We train the following values for the *gamma* parameter: [0.001, 0.01, 0.03, 0.05, 0.08, 0.1, 0.5]
To improve the predictions, we regularize the data using a StandardScaler (removes the mean and scales to unit variance) The training error and the validation error for each value allow us to plot a Complexity Curve, to select the optimal value fo the parameter.

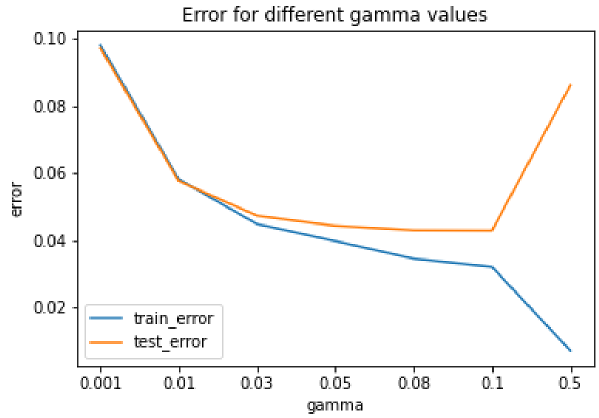

Figure 2: Complexity Curve: *gamma* value for SVM

By observing the plot, we determine that the best value for gamma is 0.03
We train the following values for the C parameter: [0.02, 0.2, 0.8, 1.2, 2, 5, 10]
Again, we regularize the data using a StandardScaler (removes the mean and scales to unit variance)
The training error and the validation error for each value allow us to plot a new Complexity Curve, to select the optimal value fo the parameter.

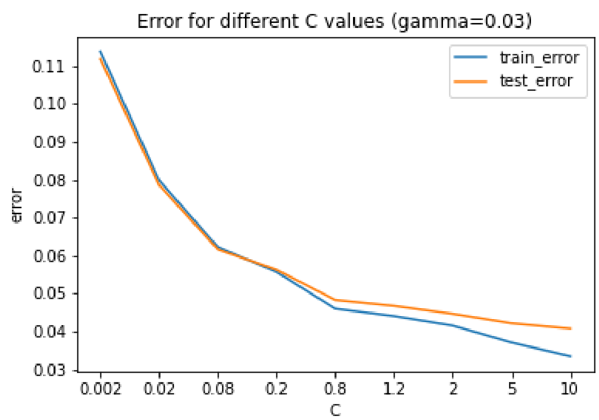

Figure 3: Complexity Curve: C value for SVM ($gamma$=0.03)

By observing the plot, we determine that the best value for C is 0.8

2.3.3 Model Evaluation

Chosen the parameters: ($gamma$=0.03, C=0.8), a learning curve shows us the training and validation scores for different data sizes.
This way, we are able to say that the score is bounded below 95%, and the model doesn't seem to continue learning after 65000/70000 training rows.

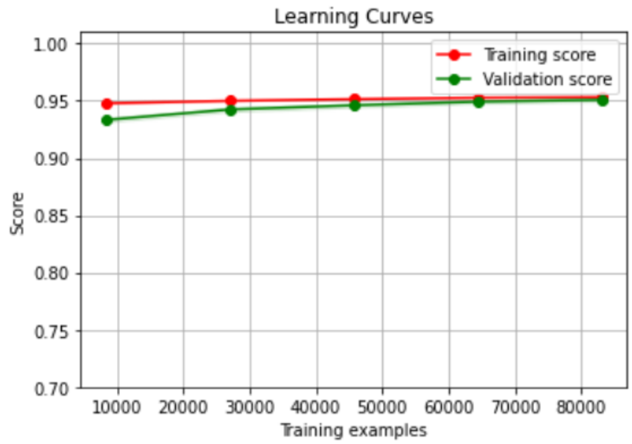

Figure 4: Learning Curve for SVM

2.4 Neural Network

2.5 Model 5

2.6 Ensemble: Random Forest

2.7 Voting Classifier

3 AUCs

4 AutoML

5 Best model: model n

PCA

Conclusion