CS 6923: Machine Learning

Spring 2018

Homework 5

Submit on NYU Classes by Fri. May. 11 at 11:55 p.m. You may work together with one other person on this homework. If you do that, hand in JUST ONE homework for the two of you, with both of your names on it. You may *discuss* this homework with other students but YOU MAY NOT SHARE WRITTEN ANSWERS OR CODE WITH ANYONE BUT YOUR PARTNER.

IMPORTANT SUBMISSION INSTRUCTIONS: Submit your solutions in 4 separate files: 1) a pdf file with your report (report.pdf)
2) a csv file with your predictions for the test examples (test_outputs.csv, in the format indicated below),
3) your main code file
4) a readme file with instructions how to run your code (readme.txt)
DO NOT SUBMIT ANY ZIP FILES.

The Flight Delay Prediction Problem

This homework is designed to give you experience with a regression problem based on real data - on-time performance of commercial airline flights in the United States, provided by the Bureau of Transportation Statistics.

Your task is to predict the time delay (in minutes) of a flight based on some features recorded about the flight.

We are giving you a training set with labels. You will experiment with this training set in order to develop a good model.

We are also giving you a test set without labels. You will submit the predicted outputs for this test set, and the program you used to train and output these predicted labels.

This is your opportunity to explore and experiment with different machine learning techniques. A large part of your score will be on the report in which you describe your experiments and how you developed your final learning method. You are expected to try 2 or more different supervised learning methods, and to experiment with different parameter settings or techniques associated with your final method. Do NOT just use tools to try 6 different methods with their default parameters! You will get more credit for thoughtfully choosing 2 methods, trying both of them, and then devoting time to improve one of them to get your final program.

Features

The features are described in Figure 1. The last "feature" is actually the target value, ARR_DELAY, that you need to predict. The other features are the input features.

If you are curious, you may be able to find some more information about the features on this site : `https://www.transtats.bts.gov/glossary.asp`

As is typical with real data, in some of the example, the values of certain input features may not have

Figure 1: Features and Target Value

DAY_OF_WEEK	Value betwee 1-7 showing the day of the week (1=Monday)
FL_DATE	Date of the flight (departure) in YYYY-MM-DD format
AIRLINE_ID	An id number assigned by US DOT to identify a unique carrier
CARRIER	Code assigned by IATA used to identify a carrier. As the same code may have been assigned to different carriers over time, the code is not always unique
FL_NUM	Flight Number - one to four character alpha-numeric code
ORIGIN_CITY_MARKET_ID :	An id number assigned by US DOT to identify a city market. Consolidates airports serving the same city market.
ORIGIN	Origin airport code
ORIGIN_CITY_NAME	Origin city name and state (comma separated)
ORIGIN_STATE_ABR	Origin state abbreviated
DEST_CITY_MARKET_ID	An id number assigned by US DOT to identify a city market. Consolidates airports serving the same city market.
DEST	Destination airport code
DEST_CITY_NAME	Destination city name and state (comma separated)
DEST_STATE_ABR	Destination state abbreviated
CRS_DEP_TIME	Departure Time. Local time, HHMM format
TAXI_OUT	The time elapsed (mins) between departure from the origin airport gate and wheels off.
TAXI_IN	The time elapsed (mins) between wheels down and arrival at the destination airport gate.
ACTUAL_ELAPSED_TIME	Elapsed Time of Flight, in Minutes
DISTANCE	Distance between airports (miles)
DISTANCE_GROUP	Distance Intervals, every 250 Miles, for Flight Segment
FIRST_DEP_TIME	First Gate Departure Time at Origin Airport.
ARR_DELAY	Target value - Difference in minutes between scheduled and actual arrival time. Early arrivals show negative numbers.

been recorded. As a result, some entries for features in both the the training and test files may be empty. It is up to you how to handle these missing features.

There may be other input features that are not necessarily useful for the regression task.

Prediction Error

This is a regression problem because the outputs (delay time) are real numbers, not classes. Note that the values can be negative (early arrival).

We will use the following mean squared error function to measure the accuracy of your predictions: $\frac{\sum_{t=1}^{N}(r^t - y^t)^2}{N}$.

Input and Output Instructions

Your final program must read in two files: a training file (flights_train.csv) and a test file (flights_test.csv). Your program must use the training file to learn a predictor, apply that predictor to the examples in the test file, and then write a file called test_outputs.csv which gives the predictions for the unlabeled examples in the test file.

IMPORTANT: Your program must be able to take flights_train.csv and flights_test.csv as input. Do not modify these files prior to giving them to your program as input. Your program must write the predictions to the output file directly, in the form indicated below.

The first column of the training and test files is the Id Number. This is different for each example (flight), and will be used as the number of the example.

Your output file (test_outputs.csv) should have just two columns: "Id" and "Delay" (the output). INCLUDE the row with the column headers, "Id" and "Delay" at the top of your output file. Each row of the output file, after the row with the column headers, should have the Id of the corresponding row in the test file, with the predicted delay. (Do NOT change the Id numbers of any of the examples.) Because your output file should be in .csv format, it should have a comma between the entries in the two columns.

Thus the first three rows of your test file should look something like this (with different Id numbers and different delay times in minutes):

```
Id,Delay
54,-10
122,15
```

You must write your program in Python or MATLAB. You DO NOT have to implement everything from scratch. In Python, you may use the tools provided in scikit-learn, and in MATLAB you may use the tools in the Statistics and Machine Learning Toolbox.

Report

You must submit a report with your program. The report should be NO MORE THAN 4 PAGES. Use a reasonable font size. Quality is more important than quantity.

1. Describe any feature selection or feature creation performed by your program. Explain how it handles missing attribute values, if this is relevant. If it converts numeric features to categorical features, or categorical features to numeric features, describe how it does this.
2. Explain the method or methods your program is using to learn. In your explanation, you do not need to repeat the details of algorithms we covered in class. Do not simply restate algorithms from class. But if you are doing something that is different from what we did in class, you may want to include pseudocode or mathematical descriptions.
3. If your algorithm had tunable parameters, include a table or description with the settings of the parameters.
4. Discuss how you arrived at your solution. You should have done thoughtful experiments and analysis. A good report will describe what you tried and why.
5. You must include cross-validation results for your final method. (You choose the number of folds of the cross-validation.)

   You may also want to include a table or graph comparing the cross-validation performance of your final method to the performance of other methods/parameter values you tried.
6. If you used existing packages or referred to papers or blogs for ideas, you should cite these in your report. (You may attach an extra page with these citations, if necessary.)

Even if your results on the test set are not good, you can still score well on this assignment if you have a thoughtful report. Conversely, even if you achieve good results on the test set, you will lose points if your report does not adequately address the items discussed above.

Code

You must submit your code file along with instructions how to run it.

1. Your code must read 2 files as input - flights_train.csv and flights_test.csv

2. Your code must output just 1 file - your test_outputs.csv
3. All data cleaning, preprocessing, feature selection, model fitting, cross validation, etc. must be done on the above input files by your code. Do not modify the given files before you run your code.
4. Your code must be runnable. Include all necessary instructions in the readme.
5. Make your code readable to avoid ambiguity
6. IF YOU DO NOT FOLLOW THESE GUIDELINES YOU ARE LIKELY TO RECEIVE 0 POINTS FOR YOUR CODE AND PREDICTIONS!

Things to think about

1. Before you start fitting your models, you might want to check the distribution of the values of individual features and the distribution of the target value.
2. Most of the attributes are categorical. How many categories per feature are there? Think about the best ways to handle them.
3. What features are actually important? Do you want to use all of them? You may or may not want to combine different features into a single, more informative one.