ECE 157A/272A - Homework 1

ECE 157/272 TAs

Fall 2021

Due: Friday, 15th of October, 11:59pm
Submit at `https://www.gradescope.com/courses/322684`
GradeScope add code: 74JWNY

1 Introduction

Welcome to the first homework assignment of the quarter!

Over the course of the quarter, we're going to be helping you develop *tools* that apply machine learning and data science to real-world problems. This first assignment starts that off with teaching the initial phases of developing a tool: Exploring the data space of your problem and writing scripts to train models.

For this assignment, you work at *Insurance Corp Incorporated*, a company in the medical insurance space. You've just received an email from your boss:

> `From: bossman.bossy@icinc.med`
> `To: you@icinc.med`
> `Subject: New development direction`
>
> Hey, legal has just approved a new diabetes study for use in our tools. I'd like you to take a dig through it, try to make some classifiers to predict patient outcomes from the diagnostic data.
>
> Since we're only prototyping for now, you can just load and save the data in **CSV**, manipulate it in **Numpy**, plot in **MatplotLib**, and crib the classifiers from **Sklearn**.
>
> I've attached the study data to this email. I've also pulled some files for patients who are due for tests before the due date I've set on this data exploration. See if you can predict how those tests are going to turn out early.
>
> When you've got a handle on the modeling, get me a written report on which classifier model you think is our best bet. I hear legal's getting into a fight with the government over some of our models not being "explainable" so, try a **DecisionTree** and, if that's not good enough, get back on why.
>
> Hope to see you in the happy hour Zoom on friday!

We are not planning a real Zoom happy hour; the above email is fictional.

2 Dataset

We are using data from the *Pima Indians Diabetes Database*, a dataset of medical history metrics and diabetes outcomes – who got diabetes, who did not. We're making use of a subset, narrowed to females of at least 21 years of age.

The file is provided in a format known as Comma Separated Values (CSV). You can open it as raw text to take a look! The data columns we have are `Pregnancies`, `Glucose`, `BloodPressure`, `SkinThickness`, `Insulin`, `BMI`, `DiabetesPedigreeFunction`, `Age`, and `Outcome`. All of these fields come in numbers, but not all fields have data. Where the data is missing, the field has been left at 0.

`Outcome` is the one you want to predict! If it's a `1`, the patient had or developed diabetes during the period of the study. If it's a `0`, they did not.

To load our data, we'll use Python's built-in `csv` module. The shortest snippet to load the whole file is:

```
import csv
with open('diabetes.csv','r') as file:
    reader = csv.reader(file)
    column_headers = next(reader)
    data_rows = list(reader)
```

You're not required to use this snippet. If this is at all confusing, we discuss loading and preprocessing the data in more depth in the *Getting Started* guide.

3 Recommended Flow

Overall, we suggest that you

1. Make a function to load your data and split it into samples and labels
2. Make a function to plot your data, to see what it looks like.
3. Make a function to alter/preprocess your data (if you think you can "feature engineer" it!)
4. Make a function to split the given data between a set of training data and set of validation data (See: **train_test_split on SciKit Learn**)
5. Train a classifier (See: **Supervised Learning on SciKit Learn**)
6. Measure its performance (See: **Cross-Validation on SciKit Learn**)
7. Save plots/metrics of its performance
8. Repeat the previous three steps until satisfied
9. Predict the data labels in *unknowns.csv* using a trained classifier

4 Problem Formulation

Input	Range	Description
Pregnancies	[0,∞)	Pregnancies the patient has had
Glucose	[0,∞)	Blood glucose level
BloodPressure	[0,∞)	Blood pressure
SkinThickness	[0,∞)	Thickness of the skin
Insulin	[0,∞)	Blood insulin level
BMI	[0,∞)	Body-mass index
DiabetesPedigreeFunction	[0,1)	Familial history of diabetes
Age	[0,∞)	Patient age

Output	Range	Description
Outcome	0 or 1	Prediction for whether the patient will get diabetes

Your task is to create a classifier that converts diagnostic and historic data about patients into a prediction for whether or not they will develop diabetes.

You are required to use the Python scikit-learn library to construct your models. You are required to use the **DecisionTree** classifier and at least two of the following others:

- **Linear Discriminant Analysis** (LDA)
- **Naïve Bayes**
- **Nearest Neighbors**
- **Support Vector Machine** (SVM)

Not all of these methods can achieve good results! Links to the documentation for each of these classifiers is available on **Supervised Learning in SciKit Learn**)

After you have trained your algorithms and selected the one you think is best, train it on the whole training set, then predict on the data in *unknowns.csv*. Make a new CSV file, *score.csv*, with only one column, the predicted outcomes, and submit it to the autograded dropbox. Submit your report to the non-autograded dropbox.

You also must write a brief report answering the following questions:

- Which algorithm did you decide was best?
- Describe in your own words how each algorithm you used classifies patients.
- Some models require setting "hyperparameters" (such as the SVM tolerance and kernel function, or Nearest Neighbors' number of neighbors checked.) Which hyperparameters did you have to tune? How did you decide on their values? **Show at least one plot of a classifier's performance versus one of its hyperparameters.**

- When choosing your final model, what percentage split did you give beteween training and validation data? Why did you make that choice? **Show at least one scatterplot marking mispredicted datapoints.**
- Show a diagram of your DecisionTree classifier's decision function. Does this decision function provide any hints about risk factors for diabetes?
- **Show all tested classifiers' results using *confusion matrices*** over your validation set. Which models overfit to the data? Underfit? Which had the best accuracy?

The report need not be excessively long. If you're spending more time putting together the report than you spent playing with the algorithms and data, feel free to drop by TA office hours for clarification on what we're looking for!

5 Grading Criteria

You must choose one model as your "best" for this data. You will run your model on data for which you will not be given outcome labels. The top five students by accuracy of their predictions on this unknown data will recieve a small amount of bonus points.

You will be graded on:

- Being "in the crowd" of accuracy results (That is the range of results just over 71%.)
- Your report answering all questions laid out in the *Problem Formulation*

6 What to turn in

- `scores.csv`: A CSV file with one column containing your model's predictions for the patients in *unknowns.csv*.
- report: A PDF document answering the questions listed in the *Problem Formulation*.
- diagrams: All diagrams listed in the *Problem Formulation*, as a part of the report.