This repository | Search | Pull requests | Issues | Gist

acardocacho / DSI_LDN_1_LESSON_NOTES Private
forked from ga-students/DSI_LDN_1_LESSON_NOTES

Unwatch 5 | Star 0 | Fork 23

Code | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Branch: master | DSI_LDN_1_LESSON_NOTES / projects / project-05 / readme.md | Find file | Copy path

juanginzo make local version optinoal | 9b3934f on Nov 10, 2016

1 contributor

109 lines (69 sloc) 5.26 KB | Raw | Blame | History

Project 5: Disaster Relief + Classification

Overview

This week, you've learned about access and utilizing remote databases, and more advanced topics for conducting logistic regression. Now, let's put these skills to the test!

You're working as a data scientist with a research firm that specializes in emergency management. In advance of client work, you've been asked to create and train a logistic regression model that can show off the firm's capabilities in disaster analysis.

Frequently after a disaster, researchers and firms will come in to give an independent review of an incident. While your firm doesn't have any current client data that it can share with you so that you may test and deploy your model, it does have data from the 1912 titanic disaster that it has stored in a remote database.

In this project, we'll be using data on passengers from the Titanic disaster to show off your analytical capabilities. The data is stored in a remote database, so you'll need to set up a connection and query the database (using Python!). After, you'll construct a logistic regression model and test/validate it's results so that it will be ready to deploy with a client.

Goal: Your job is to perform the following tasks:

- Collect your data from an AWS PostgreSQL instance and then import with Python.
- [BONUS] Before loading into Python import it into your local PostgreSQL database (look into https://www.postgresql.org/docs/9.6/static/app-pgdump.html for an utility that can help you with it).
- Perform any necessary data wrangling in advance of building your model
- Create a logistic regression model to figure out the likelihood of a passenger's survival
- Gridsearch optimal parameters for the logistic regression model
- Create a kNN model and optimize it's parameters with gridsearch
- Examine and explain the confusion matrices and ROC curves
- Create a report of your findings and detail the accuracy and assumptions of your model
- [BONUS] Change the decision threshold for positive labels using predicted probabilities
- [BONUS] Examine precision-recall instead of accuracy/ROC curves
- [VERY BONUS] Construct decision tree classifiers and bagging classifiers on the data

Pro Tip: Here are some questions to keep in mind:

- What are we looking for? What is the hypothesis?
- How can we train the model?
- What is the overall goal of this research project you have been assigned?

Requirements

- A local PostgreSQL database housing your remote data.
- A Jupyter Notebook with the required problem statement, goals, and technical data.
- A written report of your findings that detail the accuracy and assumptions of your model.
- ***Bonus:***
- Create a blog post of at least 500 words (and 1-2 graphics!) describing your data, analysis, and approach. Link to it in your Jupyter notebook.

Necessary Deliverables / Submission

- Materials must be in a clearly labeled Jupyter notebook.
- Materials must be submitted via a Github PR to the instructor's repo.
- Materials must be submitted by the end of week 5.

Starter code

Go ahead and grab the starter code to get started.

> Instructors: Solution code is provided for you here

Dataset

Your data is stored within an AWS Postgres remote database. Here are your connection instructions:

- Connecting to an AWS Postgres instance
- PostgreSQL manual

> Instructors: A local dataset is provided for you here

We have imported the Titanic data into an AWS PostgreSQL instance, which you can find by connecting here:

```
psql -h dsi.c20gkj5cvu3l.us-east-1.rds.amazonaws.com -p 5432 -U dsi_student titanic
password: gastudents
```

Alternatively, you can use a python library like pandas, sqlalchemy, or psycopg.

Suggested Ways to Get Started

- Read in your dataset
- Write pseudocode before you write actual code. Thinking through the logic of something helps.
- Read the docs for whatever technologies you use. Most of the time, there is a tutorial that you can follow, but not always, and learning to read documentation is crucial to your success!
- Document **everything**.
- Look up sample executive summaries online.

Useful Resources

- Documentation for Logistic Regression in Python
- PostgreSQl and Python

Project Feedback + Evaluation

Attached here is a complete rubric for this project.

Your instructors will score each of your technical requirements using the scale below:

Score	Expectations
0	*Incomplete.*
1	*Does not meet expectations.*
2	*Meets expectations, good job!*
3	*Exceeds expectations, you wonderful creature, you!*

This will serve as a helpful overall gauge of whether you met the project goals, but **the more important scores are the individual ones** above, which can help you identify where to focus your efforts for the next project!