ECE 157A - Final Project: Overview

ECE 157 TAs

Fall 2021

1 Introduction

This is the final project where we will **build a simple ML analytic Web API** using Django [2] and the Django REST framework [3]. This is the opportunity to integrate the ML techniques you have established in the first three homework towards an end-to-end solution that can be easily re-used by others.

2 Motivation

You may wonder why we teach web development in an applied ML course. Consider the following scenario:

> Recall you work as the only data expert in a medical company. You have done excellently whenever your boss assigned you a new data analytic task. However, as machine learning becomes so popular, all colleagues in your company even those from other departments come to you with their own data and ask for a ML point of view. You realize that you don't have time to solve their individual problems, and many problems actually fall into three main categories - classification, regression and outlier detection. One day an idea came across your mind: why not build an AI agent to present my knowledge and analyze data for these people? You remembered there was a talk about the Intelligent Engineering Assistant (IEA) when you visited UCSB. You knew the AI agent has to be a software system which can be easily deployed to various user cases. Aha, you said to yourself, a web app sounds like a good design choice!

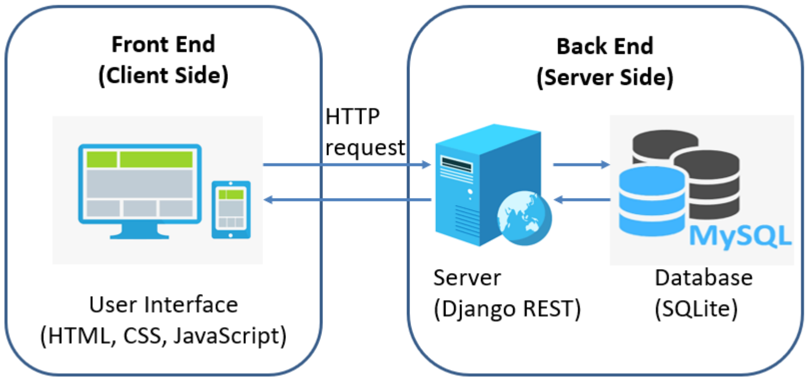


Figure 1: Full stack web development

3 Full Stack Web

A full stack web development project includes developing both the client and server software. Fig 1 shows three major components: the back-end (server), the front-end (client), and the database. The client and server communicate with each other via the standard Http request and response, a protocol for distributed information systems [5]. For example, the GET method retrieves data from the server resources based on the request specified on the client side. There are many paradigms to implement the system. In this project, we will **use the Django REST framework [3] to implement our back-end**. It is an add-on to the Django framework [2] with more powerful features. The Django framework provides us a lightweight Web server written purely in Python. Note that this server is used only in development and not for real production. Django also configures a default database engine called SQLite [7]. Again, this is for development only. We will use Django REST framework's native Request class to make standard Http request. The Django back-end can be connected to various front-end frameworks such as React [6], Angular [1], etc. But for this project, **our front-end can be a simple HTML form [4] rendered also by Django REST framework**.
In this project, you will host the server on your local machine. You also need to mock the client side in your local browser. In reality, however, the client and server can be separated and multiple clients can be distributed around the world.

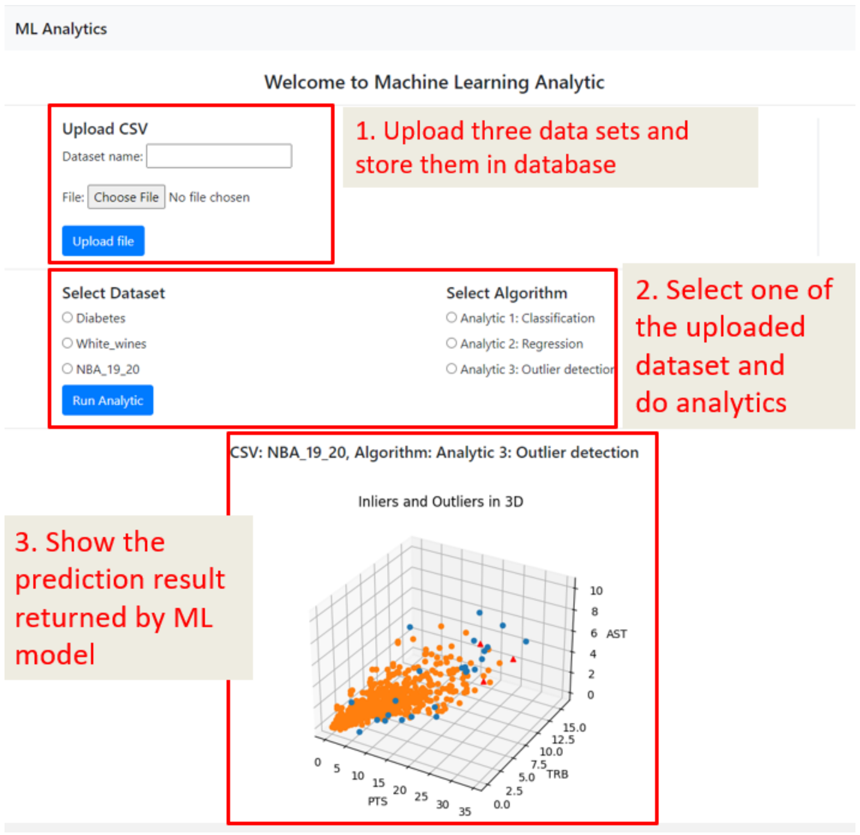

ML Analytics
Welcome to Machine Learning Analytic
Upload CSV
Dataset name:
File: Choose File No file chosen
Upload file
1. Upload three data sets and store them in database
Select Dataset
Diabetes
White_wines
NBA_19_20
Run Analytic
Select Algorithm
Analytic 1: Classification
Analytic 2: Regression
Analytic 3: Outlier detection
2. Select one of the uploaded dataset and do analytics
CSV: NBA_19_20, Algorithm: Analytic 3: Outlier detection
Inliers and Outliers in 3D
3. Show the prediction result returned by ML model
PTS
TRB
AST

Figure 2: Functionality requirements for ECE 157A and ECE 272A

4 Functionality requirements

By the end of this project, your will be able to show the following three functionalities in your web app:

- Upload three data sets (*diabetes.csv, white_wine.csv, NBA_player_stats.csv*) and store them in database.
- Toggle from the three uploaded data sets and perform inference with pre-trained ML model respectively.
- Show the prediction result returned by each of the three ML models (*classification, regression, outlier prediction*)

Fig 2 shows the GUI view of what you are expected to show during the final demo. You should host a web server on your local machine and render a HTML web page as shown in Fig 2. Note that the HTML form template is provided by the TAs. So your job is mainly on implementing the server side with Django framework.

5 Extra requirements for ECE 272A (optional for ECE 157A)

We require grad students enrolled in ECE 272A to implement the following extra functionalities by the end of project (see Fig 3 and Fig 4). Students in ECE 157A can get extra credits by implementing them:

- Make the app more robust by implementing more hands-on features:
  1. Validate file extension (.csv) before uploading.
  2. Select the latest file for analytic based on upload time, when files have duplicated names.
  3. Delete files based on file name.
- Administrator feature (see Fig 4)
  1. Log in as admin and manage the pretrained model and ML scripts
- Retrieve analytic history
  1. Implement a table listing the existing analytic results. Show the existing analytic plot when an entry is clicked .
  2. Make the 3D outlier plot more interactive

6 Grading Details

See phase 1 instruction PDF and phase 2 instruction PDF (will be released after phase 1 due) for project timeline, grading format and detailed rubrics.

References

[1] Angular. `https://angular.io/`.

[2] Django framework. `https://www.djangoproject.com/`.

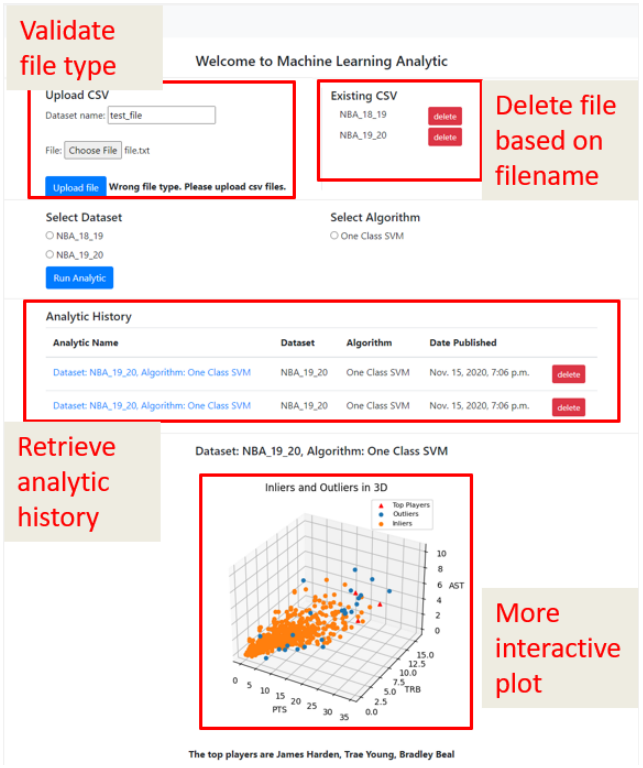
[3] Django rest framework. `https://www.django-rest-framework.org/`.

[4] Htmlforms. `https://www.w3schools.com/html/html_forms.asp`.

[5] Http. `https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol`.

[6] React js. `https://reactjs.org/`.

[7] Sqlite. `https://www.sqlite.org/index.html`.



Welcome to Machine Learning Analytic

Upload CSV
Dataset name: test_file
File: Choose File file.txt
Wrong file type. Please upload csv files.

Existing CSV
NBA_18_19
NBA_19_20

Select Dataset
NBA_18_19
NBA_19_20

Select Algorithm
One Class SVM

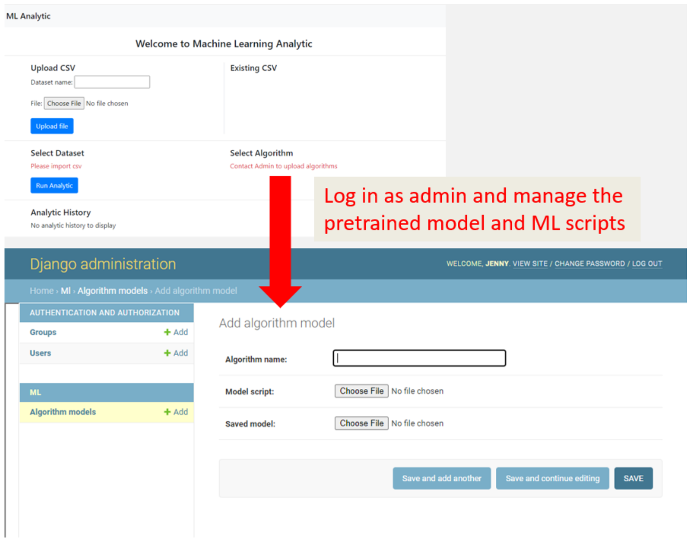
Analytic History

Analytic Name	Dataset	Algorithm	Date Published	
Dataset: NBA_19_20, Algorithm: One Class SVM	NBA_19_20	One Class SVM	Nov. 15, 2020, 7:06 p.m.	delete
Dataset: NBA_19_20, Algorithm: One Class SVM	NBA_19_20	One Class SVM	Nov. 15, 2020, 7:06 p.m.	delete

Dataset: NBA_19_20, Algorithm: One Class SVM

The top players are James Harden, Trae Young, Bradley Beal

Figure 3: Extra requirements for ECE 272A: more robust features


Log in as admin and manage the pretrained model and ML scripts
ML Analytic
Choose File
Upload file
Run Analytic
Django administration
WELCOME, JENNY. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Ml › Algorithm models › Add algorithm model
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
ML
Algorithm models + Add
Choose File
Save and add another
Save and continue editing
SAVE

Welcome to Machine Learning Analytic

Upload CSV
Dataset name:
File: No file chosen

Existing CSV

Select Dataset
Please import csv

Select Algorithm
Contact Admin to upload algorithms

Analytic History
No analytic history to display

Add algorithm model

Algorithm name:
Model script: No file chosen
Saved model: No file chosen

Figure 4: Extra requirements for ECE 272A: admin feature