Jonathan Chang
4/14/2020
CS598 – Data Mining Capstone

Task 7 Report: Restaurant Hygiene Prediction

Introduction

For this task, I chose to utilize the model I built in Task 6 which is used to predict whether or a not a restaurant would pass a hygiene inspection based on some features such as: text reviews of the restaurant. Zip code, average rating, etc… I created a simple web app that allows users to input some of these aforementioned features and receive a score of the likelihood of the restaurant passing the hygiene inspection.

As of the time of this report, my app has been deployed to: https://restaurant-hygiene-app.herokuapp.com/, and a screenshot of the interface is included later in this report. I was not able to access the UIUC servers while on VPN, and deployment via Heroku should allow the app to be consumed by a larger audience. The code is available at my Github repository: https://github.com/jonchang03/restaurant-hygiene-app.

App Overview

Audience Envisioned:
I would imagine any users who would like to know whether or not to eat at a restaurant based on cleanliness but may not have immediate access to hygiene information, could use this system. For instance, they could just borrow some reviews of a given restaurant from Yelp, and fill out the form to receive a hygiene prediction score. Retrospectively, users who eat at a restaurant and want to enter their own review of the restaurant and rating based on their experience would be able to get the score as well, and they can compare their own intuition of the restaurant's cleanliness to the score returned by the system.

Obviously, there are some limitations to the app in its current state. For example, the zip codes for the data received began with 981, which I believe restricts us to restaurants in the Seattle area. The model also received limited labeled training data of 546 records, so performance could surely be increased with more data. However for demonstration purposes, this app would suffice nicely.

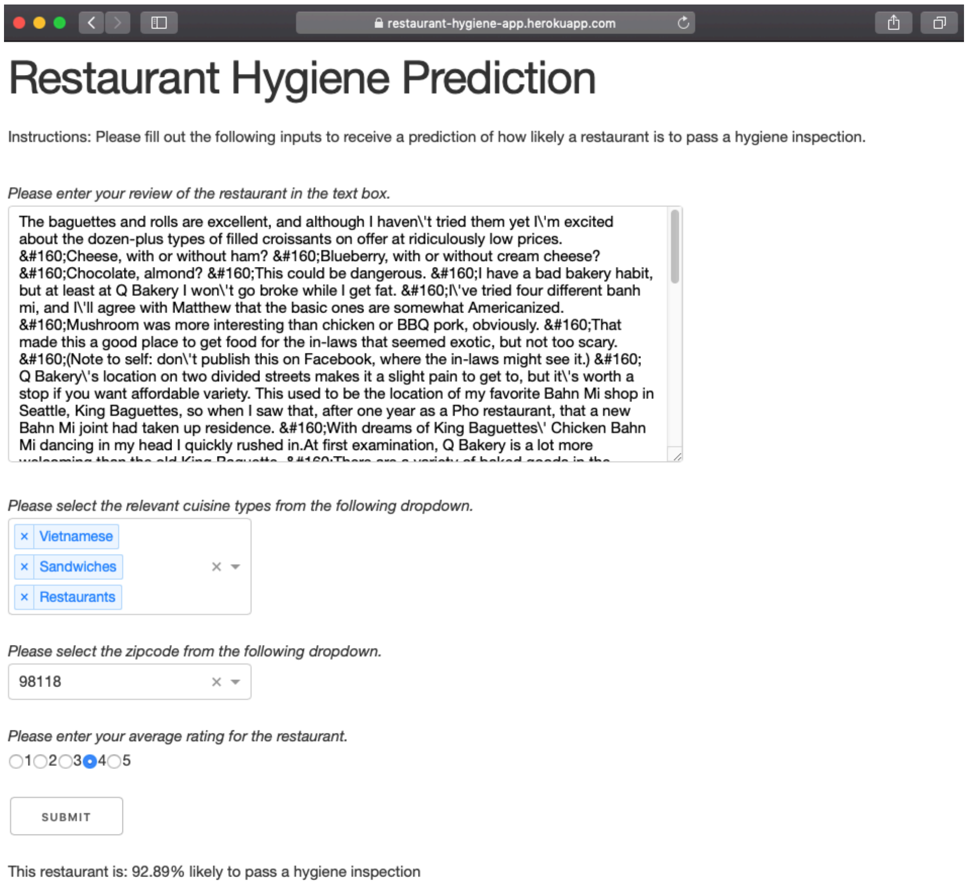
Technical Details:
I essentially reused the sklearn pipeline I had built out in Task 6. My pipeline included 2 stages. The first one was a preprocessor that used sklearn's ColumnTransformer() to process the different inputs. I used a OneHotEncoder() for the zip codes and average rating, and Finally, I used a Naïve Bayes classifier which performed "well" both locally and on the leaderboard. After training the classifier, I was able to serialize the pipeline using joblib, and reference that within my Dash application.

For the web app itself, I used the Dash framework which is built on top of Flask, and allowed me to very easily create an app layout with various inputs that are described in the following section.

The inputs and outputs of the application interface are controlled via "callbacks" which was a defined function that would allow the output to be updated. So essentially the app takes in the user input, and within the callback function, does some simple preprocessing and massaging to get the input data into a format that my pipeline could return a predicted probability for. Finally, for deployment, I was able to follow a guide on the Dash documentation which provided an example of how to deploy a my application to Heroku which is just a platform as a service that is able to host Python (among other) applications.

Interface Design

Again, the app has been deployed to: https://restaurant-hygiene-app.herokuapp.com/



Restaurant Hygiene Prediction

Instructions: Please fill out the following inputs to receive a prediction of how likely a restaurant is to pass a hygiene inspection.

Please enter your review of the restaurant in the text box.

The baguettes and rolls are excellent, and although I haven\'t tried them yet I\'m excited about the dozen-plus types of filled croissants on offer at ridiculously low prices. &#160;Cheese, with or without ham? &#160;Blueberry, with or without cream cheese? &#160;Chocolate, almond? &#160;This could be dangerous. &#160;I have a bad bakery habit, but at least at Q Bakery I won\'t go broke while I get fat. &#160;I\'ve tried four different banh mi, and I\'ll agree with Matthew that the basic ones are somewhat Americanized. &#160;Mushroom was more interesting than chicken or BBQ pork, obviously. &#160;That made this a good place to get food for the in-laws that seemed exotic, but not too scary. &#160;(Note to self: don\'t publish this on Facebook, where the in-laws might see it.) &#160; Q Bakery\'s location on two divided streets makes it a slight pain to get to, but it\'s worth a stop if you want affordable variety. This used to be the location of my favorite Bahn Mi shop in Seattle, King Baguettes, so when I saw that, after one year as a Pho restaurant, that a new Bahn Mi joint had taken up residence. &#160;With dreams of King Baguettes\' Chicken Bahn Mi dancing in my head I quickly rushed in.At first examination, Q Bakery is a lot more

Please select the relevant cuisine types from the following dropdown.

× Vietnamese
× Sandwiches
× Restaurants

Please select the zipcode from the following dropdown.

98118

Please enter your average rating for the restaurant.

○1 ○2 ○3 [✓ 4] ○5

SUBMIT

This restaurant is: 92.89% likely to pass a hygiene inspection

As seen in the screenshot above, the app is relatively simple, and includes input options that correspond to features to my model. There is a text box that allows users to submit a review of the restaurant, a multi-selector dropdown that allows users to select the relevant cuisine types, a zip code selector (based on the unique zip codes available in the training data for the model), and the average rating using radio buttons. At the very bottom of the app, when the user clicks the submit button, an output which represents the percentage likelihood of a restaurant passing a hygiene inspection, is updated on the screen.

Given additional time, I would played around more with CSS and styling of the app to make it a little bit less minimalistic, but I believe that I was able to achieve full functionality with my app.

Another area for future work would be to reduce some of the limitations of this app, for example, by finding training data with greater scope.

References

- https://scikit-learn.org/stable/modules/model_persistence.html - model serialization
- https://dash.plotly.com/dash-core-components/input - dash example inputs
- https://dash.plotly.com/deployment - deployment via Heroku