BBM467 – Data Intensive Applications

Short Data Science Project (SDSP)

Problem

There are some diseases which are hard to diagnose by physicians. For instance, the disease may be rare and the physician responsible has no experience on that disease whatsoever. Or, there may be other diseases with similar symptoms and the physician can not make sure about her/his diagnosis. Whatever the reason is, this sort of cases typically result in unnecessary and expensive tests, e.g., blood work, x-rays, MRIs, etc. Besides wasting valuable resources, putting patients into distressing situations is also an issue. Therefore, diagnosing a disease by using resources optimally and without sacrifing the healthcare quality is of great importance.

Your Tasks

In this project, you are supposed to build a machine learning (ML) model to predict possible diseases for a given patient. Furthermore, you will develop a Web application to front your model and allow any physicians to use your model while diagnosing their patients. You application will produce a list of possible diseases with their probabilities. The idea here is to allow the physician start diagnosis process by considering the most possible disease first.

Task 1.

You will be provided with a dataset (sdsp_*patients.xlsx*). In this dataset, due to privacy reasons, predictor variables (features) are named as *Feature_X* and the target variable (class name) is named as *Disease*. The target column may have values like *Disease_Y*. Here, *X* and *Y* are integers.

Your first task is to go over the dataset and prepare it for machine learning. Then, you will build a model or models to predict probabilities of diseases a patient may have. Then, you will tune your model(s) by reducing number of features you use to build the model. You will try to find the optimal number of features to diagnose a disease.

Task 2.

Your second task is to develop a Web application to front your ML model. Technologies that you can use in this part was announced through Piazza.

The application simply asks its user for data that correspond to your features used in the ML model (Figure 1). This form must be generated dynamically. That is, if you make any changes

on your model like removing or adding features, the application must still be working. Input components, e.g., text-box, dropdown-box, and radio-buttons to enter data must be dynamically chosen, too. For instance, if a value to be entered is categorical, using a list may be the best choise.

Once the user provides necessary data for a patient, she/he presses the *Predict* button (Figure 2). You can assume that all fields are mandatory.
Then, the model predicts the possible diseases for this patient. It provides a list of disease with their probability values (Figure 3).

The *Clear* button clears the form and sets fields to their default values.

Also note that, the wireframes provided here (Figures 1, 2, and 3) are to give you an idea about the look of your Web application. You can come up with your own design if you like.

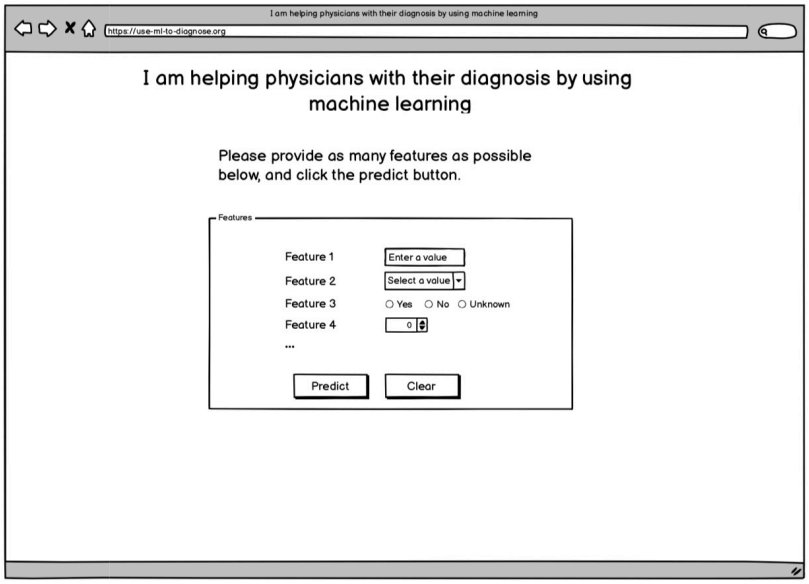

Figure 1. Start page of your application

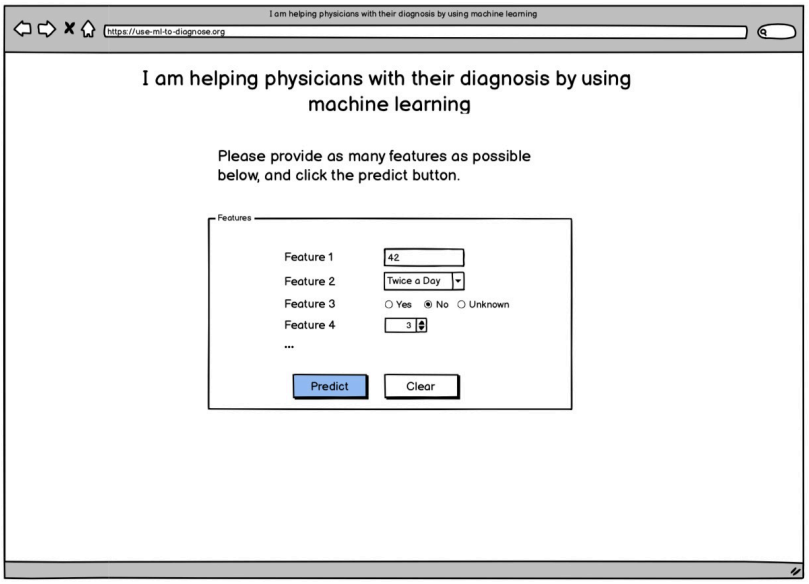

Figure 2. User provides data and ask for prediction

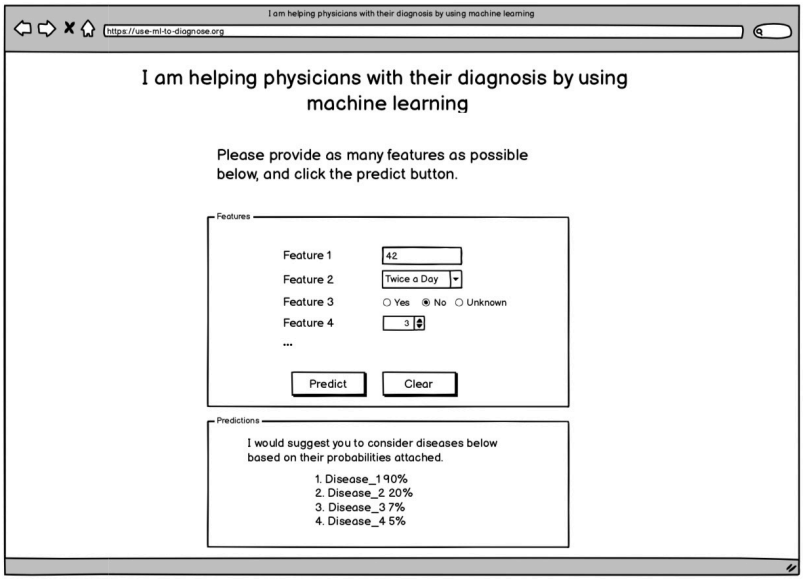

Figure 3. The application returns possible diseases with probability values.

Delivery

You will deliver your zipped source codes (both Web app and Jupyter Notebooks) and reports. Your report will include your specific solutions and instructions about deployment of your application.

Further details will be announced through Piazza.