BBM 467 - Data Intensive Applications
SDSP Report

Spring 2021
Dr. Fuat Akal

Arda Hüseyinoğlu
21627323

Department of Computer Engineering, Hacettepe University
May 11, 2021

1 Data Preprocessing and Model Selection

In Jupyter Notebook `feature_and_model_selection.ipynb`, you can see the data preprocessing steps, how the features are selected and the model selection methodology. After running the corresponding cells of the notebook, you will get 3 files (you do not need to run, they are already in the folder):

- `features.csv` : cleaned version of the original dataset(`sdsp_patients.xlsx`) in csv file format.

  To clean the original dataset, we first fix some problems: There are some missing values are indicated as empty strings in the given data. So, we convert them to `np.nan` like rest of the missing values. Also, data type of the Feature_3 is given wrong. We converted it to float64. Then we impute the missing values. For the numerical features we use the mean value of the corresponding column and for the categorical features we use the most frequent value to fill the missing values. After that, we use label encoding and one-hot encoding for the categorical features to convert them into numerical values. Lastly, we normalize the data using min-max scaling. However we save the non-normalized version of the data in the csv file, since we need scaling information (min,max) of every selected feature in the data whenever the user provides a new input.

- `target.csv` : includes class labels for every sample, in csv file format (labels of samples are separated from features.csv for convenience).

- `best_model_info.json` : json file that includes information for the best model obtained, which are the best classifier name, best hyperparameters and values for each for the best model and the selected features to get the highest accuracy.

  To select best features, we first get the variance of each feature to directly drop them whose variance is 0 or very close to zero. After that, we check the highly negative and positive correlated feature pairs. To drop unnecessary features, we first decide a threshold value which indicates the minimum correlation score between pairs. For each feature, we check if it has any correlation score which exceeds the threshold with other features. If so, we simply drop that feature to keep only one of them. We also use some pragmatic approaches to select features based on the feature weights of the model. After applying all the methods mentioned above. We select our best features.

  We will use three different classification algorithm (Logistic Regression, Random Forest Classifier and K-NN) to decide which one of them fit the problem and data more. For the hyperparameter tuning, we use cross validation with grid searching, which provides testing all combination of the parameter settings we give for each classification algorithm and returns the hyperparameters that gives the best accuracy for the model.

Please see the `feature_and_model_selection.ipynb` for more detailed explanations and the code.

2 Requirements for Running the App

- streamlit==0.80.0
- pandas==1.2.4
- numpy==1.19.5
- scikit-learn==0.24.1

If you don't have `streamlit` installed in your environment, the you can easily install it using `pip`:
`> pip install streamlit`

You can also do the same for other packages, if they are not installed:
`> pip install numpy`
`> pip install pandas`
`> pip install -U scikit-learn`

3 How to Run

Please check the following files if they are in the unzipped folder, to run the app properly:

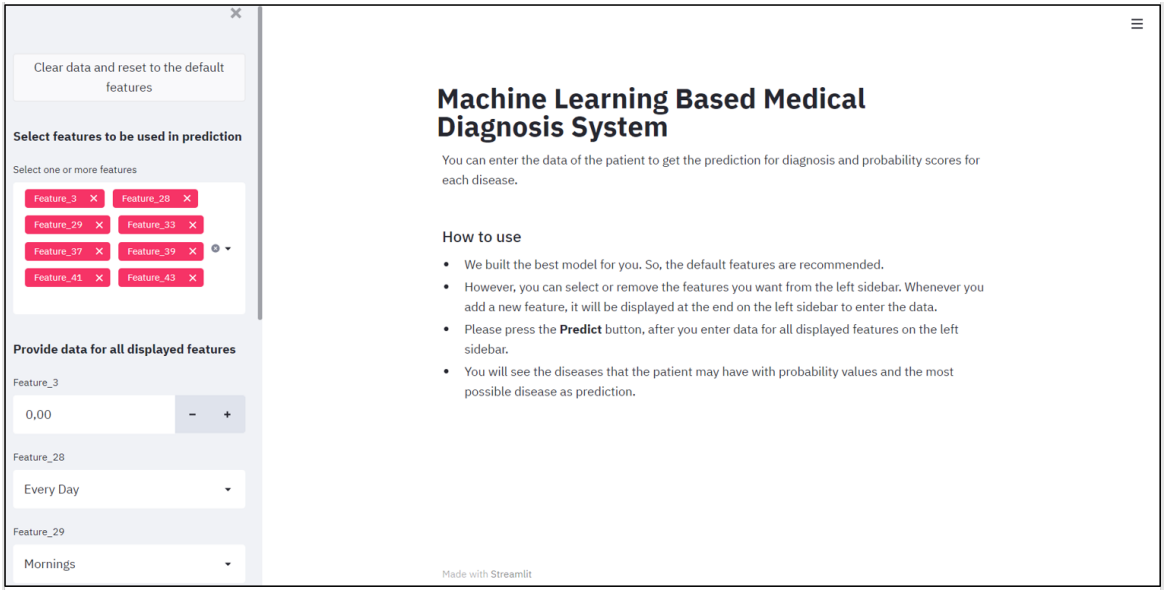
- `app.py`
- `SessionState.py`
- `features.csv`
- `target.csv`
- `best_model_info.json`

To run the app locally, please enter the following commands in terminal:

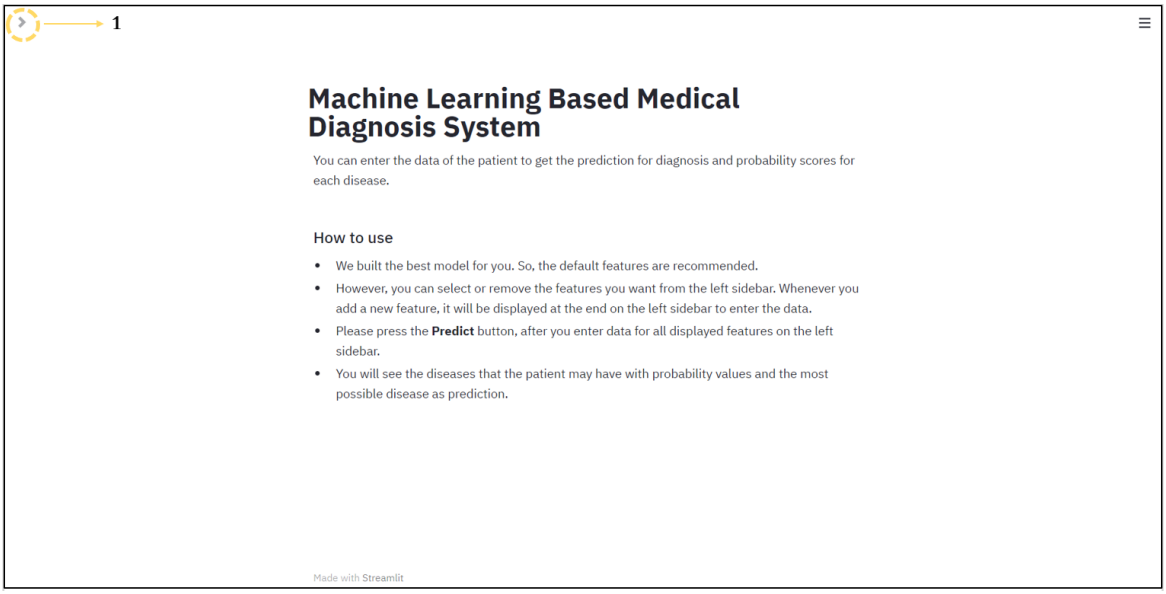
```
> cd path_to_unzipped_folder
> streamlit run app.py
```

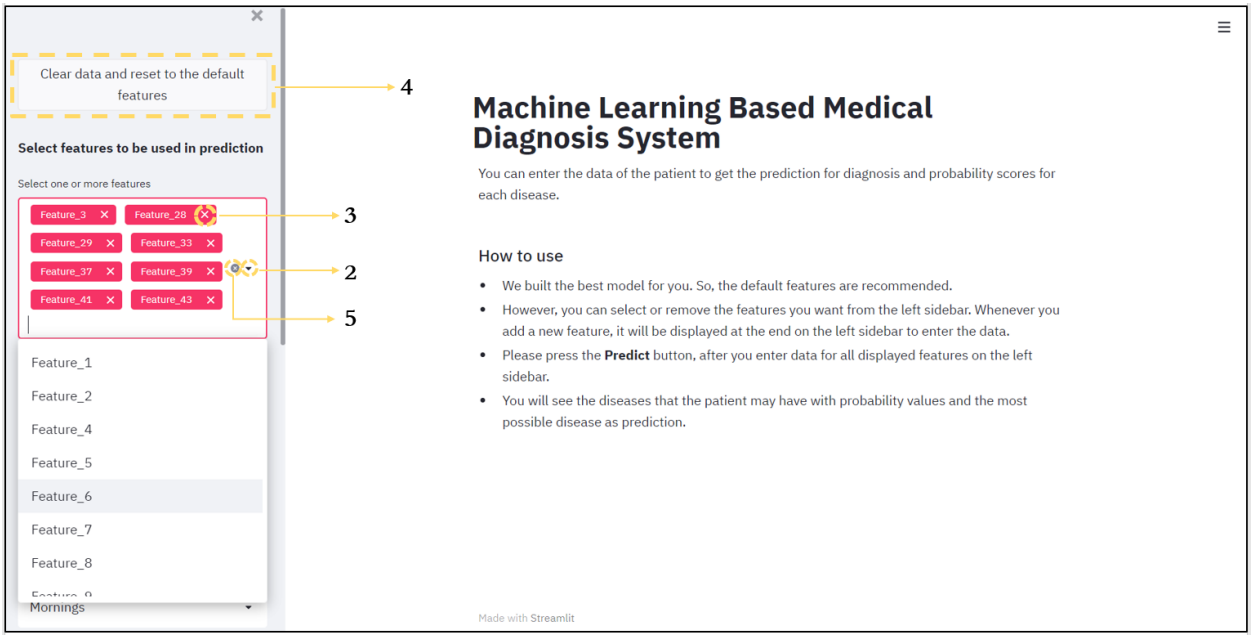
4 App User Manual

This is the page you see first after running the app. You can see the information about how to use the app to get the predictions for diseases with corresponding probabilities according to the entered patient data.



By default, left sidebar is not hidden. However if it is hidden then you can click the button which is circled and enumerated as 1 in the image below.

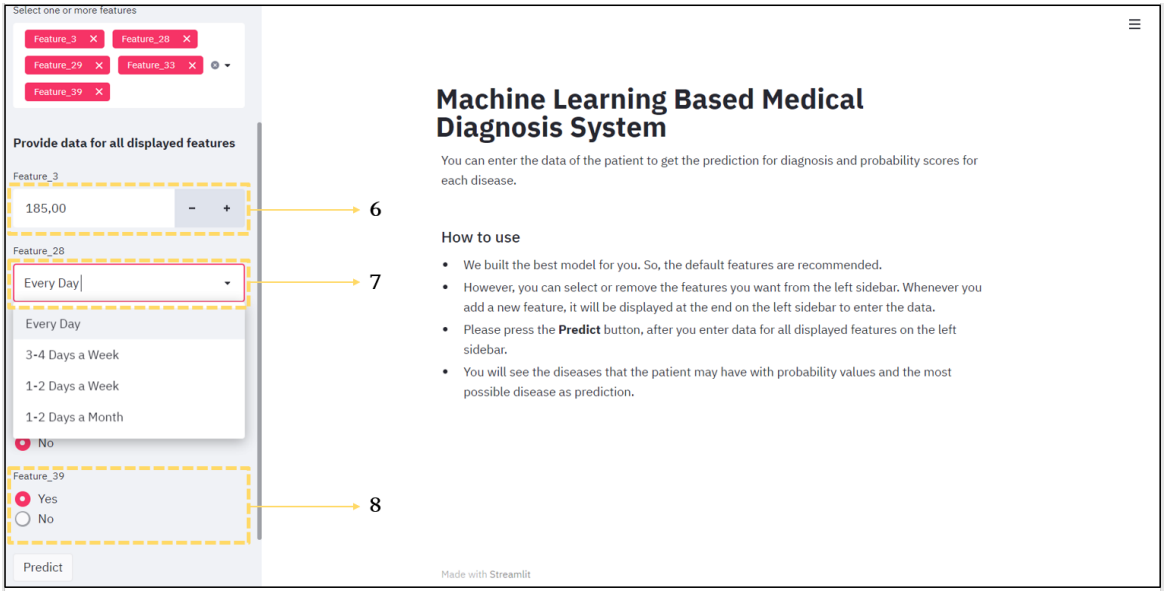




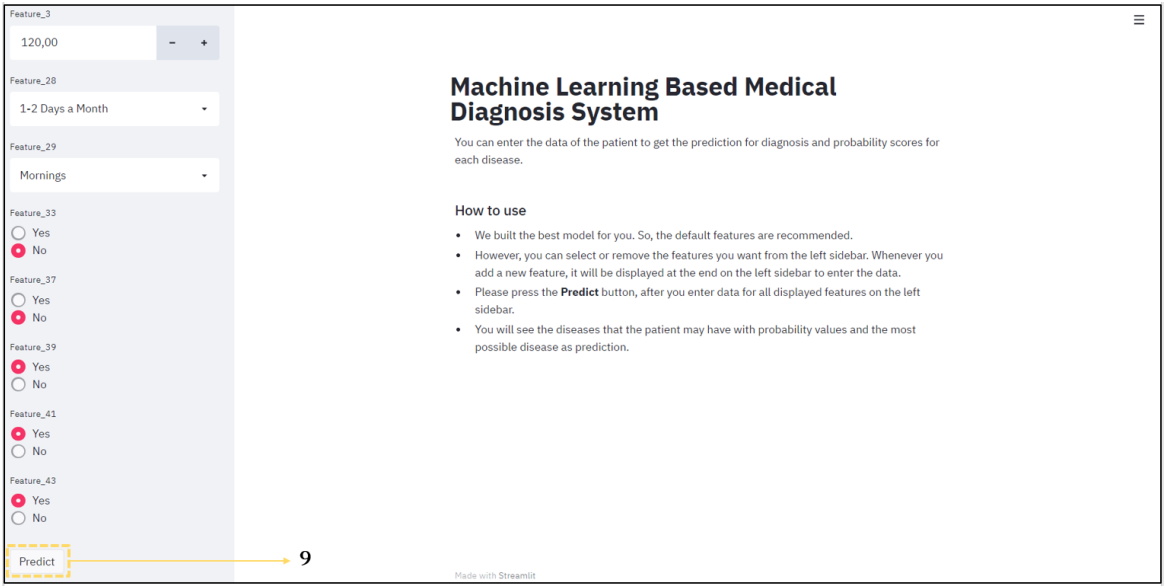
By default, features that we selected in the `feature_and_model_selection.ipynb` is displayed on the left sidebar and ready to enter the patient data. And these default selected features are recommended to achieve the best accuracy. However, user can select additional features or remove features from the selected ones easily. That provides user to train model with new features and get predictions for the new model without writing any line of code.

If you want to clear the entered data and reset to the default features you can click the corresponding button (**4**). To remove any selected feature displayed on the box, you can click the cross symbol near to the corresponding feature name (**3**). To remove all the selected features, you can click the button (**5**). To list unselected features you can click the button (**2**), then you can add any feature you want. When you add a new feature, input area for the added feature will appear at the end of the left sidebar.

You can enter the patient data for each feature displayed on the left sidebar (type of the input area -select-box(**7**), radio(**8**), number-input(**6**)- change based on input type -categorical, numerical-). You must enter data for all displayed features to get a prediction.



After you enter data for all the displayed features, you can click the 'Predict' button (**9**) to get prediction:



Finally, you can see the diseases that the patient may have with probability values and the most possible disease as the prediction (**10**)