Course Project Specifications

Course: BSAN 6198-01 – ML Model Deployment and Pipeline
Semester: Spring 2025
Instructor: Prof. Arin Brahma

Project #3: Building & Deploying No-show Predictor (CHLA Use Case)

PART 1 – Build; Select your best Predictive Model and generate your serialized ML prediction model object (.pkl or .sav)

(a) Build and select the best ML model (as per your criteria) to predict No-show (Yes/No)
(b) Test the prediction function of your final model by entering your model's predictive features of a hypothetical patient and display the outcome predicted by the model in the notebook code TEST section. Create a clearly defined TEST section in your notebook.
(c) Save your best performing model as a "trained model" object (.pkl or .sav)

Data: CHLA_clean_data_until_2023.csv
Data Location: Brightspace
Data Use Instruction:
This is client data and a protected information. You need to download the data at your local drive and use your local file location as your data Path in your code. Data and your codes must be in the same folder of your laptop. You must not share the data with anyone outside your team. You must not "post" the data in any public internet location, or share with anyone in a method that is not secure. At the end of the project you are required to delete the data from your personal laptops or devices you have used to store the data.

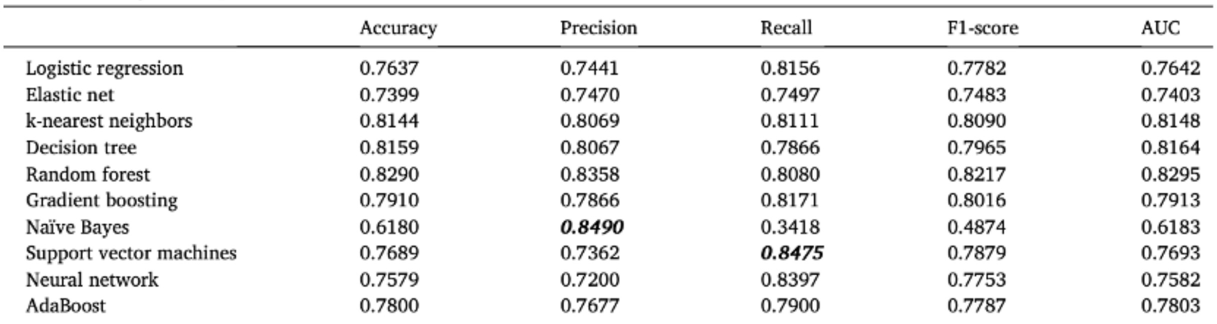
Please follow this guideline strictly. We are getting the benefit of using real client data for our project, but that also puts the burden of data privacy and confidentiality upon us.

Development Instructions:

- Create various candidate models using various two-class classifier algorithms available in scikit-learn
- Create a table of performance values of all algorithms as per the Project Report (Part 1) section below.
-

- Project Report (Part 1) – This part will have the following:
  - Business understanding discussion (half a page max)
  - Exploratory Data Analysis to present your observations and findings about the data
  - Create a performance comparison summary table like the example table given below. Include all scikit-learn classifier algorithms you have tried:

Performance Comparison



	Accuracy	Precision	Recall	F1-score	AUC
Logistic regression	0.7637	0.7441	0.8156	0.7782	0.7642
Elastic net	0.7399	0.7470	0.7497	0.7483	0.7403
k-nearest neighbors	0.8144	0.8069	0.8111	0.8090	0.8148
Decision tree	0.8159	0.8067	0.7866	0.7965	0.8164
Random forest	0.8290	0.8358	0.8080	0.8217	0.8295
Gradient boosting	0.7910	0.7866	0.8171	0.8016	0.7913
Naïve Bayes	0.6180	***0.8490***	0.3418	0.4874	0.6183
Support vector machines	0.7689	0.7362	***0.8475***	0.7879	0.7693
Neural network	0.7579	0.7200	0.8397	0.7753	0.7582
AdaBoost	0.7800	0.7677	0.7900	0.7787	0.7803

- Conclusion & Discussion (Part 1)
    - Justify (a) Is this a good model to predict no-show? Why?
    - What did you observe about this predictor you developed? Any critical observations?
    - What might be the limitations of your model?

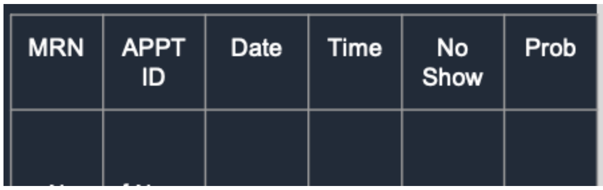
PART 2 – Deploy your best model in a way "useful and actionable" for CHLA clinics. The business goal of this solution is for CHLA clinics to receive a table of Patient-Appointments with "No-show" predictions and probabilities, when the clinic name and appointment date range is entered in the input screen. You need to do the following for this purpose:

(a) End users will input the following two fields (see example below):
- Clinic Name = "VALENCIA CARE CENTER"
- Appointment Date Range: Start Date = "1/1/2024", End Date = "1/31/2024"

(b) Retrieve the appointment records based on the above criteria from the second data file provided "CHLA_clean_data_2024_Appointments.csv"
Data Location for Downloading: Brightspace

(c) Use your saved best "trained model" from Part 1 to predict No-show and probability for each of the appointments retrieved in step (b). Assume probability > 0.5 will result in a "no-show".

(d) Output a list like shown below to the deployed user interface:



MRN	APPT ID	Date	Time	No Show	Prob

Data: CHLA_clean_data_2024_Appointments.csv

- Project Report (Part 2) – This part will have the following:

  - Include following screenshots:
    - End-user Input Screen
    - Result Output Screen (Predicted table) displayed at the end-user interface
  - Conclusion & Discussion (Part 2)
    - How would you describe the difference between Part 1 and Part 2 in terms of ML Model & Deployment architecture? (refer to lecture slides)
    - What did you observe about this predictive application you developed? Any critical observations?
    - What might be the limitations of your application?
    - Overall, what did you learn from Part 1 and Part 2 of this project in terms of your ML Application development knowledge?

- Submission Details – Submit the following at BrightSpace
  - Project Report (Part 1 and Part 2 integrated)
  - All notebook codes (.ipynb)
  - All python code (.py)
  - Requirements.txt
  - Streamlit URL for the working active App (in your report)
  - Readme.md
  - Load up all of the above in your GitHub repository and share the Github link (in your report)