We want to develop a program that tells us the predicted amount of no shows for appointments at clinics governed by CHLA. This will allow us to double book some appointments that have very high likelihood of no shows allowing for our staff to make the most effective use of their time by not having to experience the down time caused by the no shows. Succes in this matter would entail correctly identifying no show appointments and booking extra appointments to not have down time while also not double booking appointments that are likely to have the patient show since that would cause long wait times for patients.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Train Accuracy** | **Test Accuracy** | **Precision** | **Recall** | **F1 Score** | **AUC** |
| **Logistic Regression** | **0.970501** | **0.972460** | **0.763158** | **0.671296** | **0.714286** | **0.977038** |
| **Decision Tree** | **1.000000** | **0.961776** | **0.629108** | **0.620370** | **0.624709** | **0.800300** |
| **Random Forest** | **1.000000** | **0.967474** | **0.748428** | **0.550926** | **0.634667** | **0.984641** |
| **Naive Bayes** | **0.609034** | **0.598053** | **0.112336** | **0.990741** | **0.201792** | **0.973618** |
| **K-Nearest Neighbors** | **0.973884** | **0.959164** | **0.613402** | **0.550926** | **0.580488** | **0.923867** |
| **Support Vector Machine** | **0.975546** | **0.969848** | **0.745856** | **0.625000** | **0.680101** | **0.978394** |

(Note I use Random Forest in my code because it was the best preforming model, but as I tinkered with the code to integrate the pred app correctly, that model performance fell. If given more time I would have made the simple switch to Logistic Regression in the final version.)

Logistic Regression is my best preforming model and I think its okay but not really good enough to deploy. The F1 score is .71 meaning the model is likely to make quite a few errors mainly in the area of recall.

The most critical observation I noticed was the Precision tends to be much higher than recall and this makes sense due to our class imbalance with the vast majority of our data containing appointment records where the patient did show up.

This is also what I believe to be a limitation of the model because given our input data the model will likely always lean towards predicting a patient to show up.

Part 2: A screenshot of a medical appointment

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

Part 1 is focused on the back-end of machine learning, which involves data preprocessing, model training, and evaluation. Part 2 is focused on the front-end deployment, integrating the trained model into an interactive application via streamlit for making real-time predictions on new data. This incapsulates the idea of developing the front-end deployment to have models shifted in and out because the input from the back end to the front end is the ML model.

My main observation was that it seems all the predictions are saying there will not be a no show which aligns with the fears of part 1. This means either that the model is not preforming well or that the limited data we tested happens to only contain patients that are likely to show.

The data can’t respond to real world events. A freeway closing, a fire, or rainy conditions cannot be captured by the model. This means external factors will impact model performance. So end users should be made aware of this limitation

o Overall, what did you learn from Part 1 and Part 2 of this project in terms of your ML ApplicaOon development knowledge?

The biggest thing I learned was the importance of saving a preprocessing pipeline. I had a terrible time trying to get my model to make predictions based on the 2024 dataset. My fix involved capturing the pipeline and features in their own pkl files to apply to the new dataset. While this was very difficult to accomplish it makes perfect sense and was a good learn experience that the data to be predicted needs to have the same transformations to fit what the predictor is expecting.

Predapp:

<https://p03hl31.streamlit.app/>

<https://github.com/Hlissner31/P03---CHLA-PRED>