Problem Statement —> Predicting whether a person has heart disease or not, based on various parameters like age,diabetic,sleep hours, past heart stroke,smoking,alcohol etc.

Github Repo containing codes and dataset: https://github.com/LegendsDen/Heart-Disease-Prediction-CS361
Things we observed as we did Exploratory Data Analysis:

1. **Effects of BMI**: We observed that **obesity does lead to more heart disease** as the percentage of obese people with heart disease was higher than the number of people less obese.(our methodology was to express the classes as a percentage, like out of 100 people with bmi range 15-25 how many have heart disease, same is followed below)
2. **Effects of Age**: We have also observed that Age plays a significant role in whether a person has heart disease or not. **Older people have a much higher chance**, and the older you are the more probable that you have a heart disease.
3. **Effects of Sex**: An interesting observation was that Men had a higher percentage of heart disease compared to women, this can be attributed to various things such as men are more likely to smoke.
4. **Effects of health**: We had a general health parameter with values 1 - 5, we found out that people who had chosen **lesser values** here were more probable to have heart disease.
5. **Effects of Sleeping Hours**: We observed that **lesser sleep (0-4 hrs) leads to higher heart disease rate**, and **sleeping 10+ hours had a higher percentage of heart diseas**e. So we searched through the internet to find if there was a research on this and it matches our observation. An observation possible is that more sleep is not the direct cause of heart disease, but it is the sign of disease instead.
6. **Effects of diabetes**: We observed that **people with diabetes had a high chance of having heart disease.** On observation of mothers who had diabetes during pregnancy, diabetes of that sort does not seem to have an effect on heart disease.
7. **Effects of smoking**: **Smokers are observed to be twice more likely to have heart disease**.
8. **Effects of Alcohol**: Our dataset did not have the amount of Alcohol drunk by each person, it just had a yes or no. We observed that people who drank alcohol were likely to have lesser heart disease, but as mentioned, **since the amount of alcohol was not given , results are inconclusive.**
9. **Heatmap analysis** showed no strong correlation between features and heart disease; the highest was age with a correlation of 0.23.

What we understood as we were learning more about our data:

- For our use case, **Recall is much more important** than our other measures because we want to minimize False Negatives. We want to make sure that people having heart disease should not be falsely marked as negative, minimizing this is our goal.
- We had a **class imbalance, more people without heart disease compared to people with heart disease** , we realised that along as we trained our models, as it led to poor results in some of the models which couldn't handle class imbalance.
- We tried to perform PCA on the dataset first, but we later realized that it did not have a significant effect on the results. This is expected with less correlated attributes.

Later,We converted categorical features into numerical form using **Ordinal Encoding** for ordinal variables and **One-Hot Encoding** for nominal (binary) classes to prepare the data for model training and evaluation.

Logistic regression:

Since we already have built in models , the goal was to optimize the hyperparameters which provided the best overall recall.

- The first parameter looked upon was class_weight, we tried running without putting a balance on the class_weight, what we observed was extremely high recall, precision,accuracy on the negative class, but very poor on the positive class. This is as expected because our data is imbalanced with a much higher number of negative classes. So we had to use **class_weight='balanced'**.
- The second parameter looked upon was penalty (regularization) and what value of c (regularization strength) to choose based on that. What we did was to try out different models with l1(lasso) as regularizer and iterate through different values of c in a range. Same was done for l2,

We found that **l1 with a C value of 0.0001 was performing optimally**, hence that was chosen. It makes sense that l1 performs better because l1 makes the contribution(weights) of certain attributes of less importance to zero, and our dataset had quite a few attributes contributing less to results.
K fold cross validation was used to choose the model.

- Third parameter was **max_iter,** the number of iterations till convergence. In almost all cases we found that the model converges quite early as shown in the graph. As a result we have chosen **10.**

Decision Trees:

- We applied K-fold Cross Validation to find optimal value of hyperparameter **max_depth**.We checked for range (1,21) and found,
- For **low max_depth(1**) → we have a low recall value , since it is a **simple model**, and it label most cases as positive class , i.e couldn't understand the complexity of model
- For **high max depth(7-20)** → model **overfits** , it memorizes training folds but generalizes poorly on validation folds.
- Around **depth (4-5)** model captures a good balance between **complexity** and **generalization.**

Thus we took **max_depth as 5** for our final model evaluation.

KNN Classifier

- We applied KNN classifiers to the data and got very bad recall which is due the fact that our data is highly imbalanced towards the negative,so there are a very high number of False Negatives.
- To reduce the effect of class imbalance we have used **SMOTE**.

SMOTE creates synthetic samples based on existing ones.
For each minority class sample, SMOTE picks **k nearest neighbors**.
It selects one or more of these neighbors **randomly**.
Then, it **interpolates** between the sample and the neighbor to generate a **new synthetic point**.

- We have also done K Fold Cross Validation to find the optimal value of **n_neighbors.** We got the best recall when **n_neighbors** were in range 12-17. So **we chose n_neighbors=15**

We later made a **Precision -Recall** graph for all above 3 models to evaluate performance and compared it with a Random Classifier, and found **our model is much better than it** .We chose it over ROC curve , as our dataset was highly imbalanced thus, ROC might say our model is "good" just because it handles negatives well.

Note: We also tried **K-means++** but it performed poorly as it is a clustering algorithm and does not use labels. Reason for the failure:K-Means++ clusters based on **Euclidean distance** to centroids. If one class dominates the dataset , the **majority class forms a dense blob**, and K-Means++ just sticks centroids inside or near that blob.

Code—> We have commented on the part where K-fold Cross Validation is used, as it takes a considerable amount of time to get the optimal value of any hyperparameter, as it finds the best metrics across various folds and hyperparameter ranges.

Transformer Model—>We later tried with ChatGpt and Deepseek with training on data and then testing. We got a result of 0.58 recall,which is slightly worse than our result of 0.82 recall which we got with Decision trees,which could be due to -

- ChatGPT is trained mostly on **text**, not structured datasets like CSVs with numerical and categorical features.It **can't "learn" patterns from structured tables** the way a Decision Tree,Logistic Regression etc.
- We also did **feature encoding**, **SMOTE for imbalance**, **correlation filtering**, **hyperparameter tuning**, etc.LLMs can't do this kind of preprocessing on their own — they work with raw, often poorly structured data, leading to poor predictions.