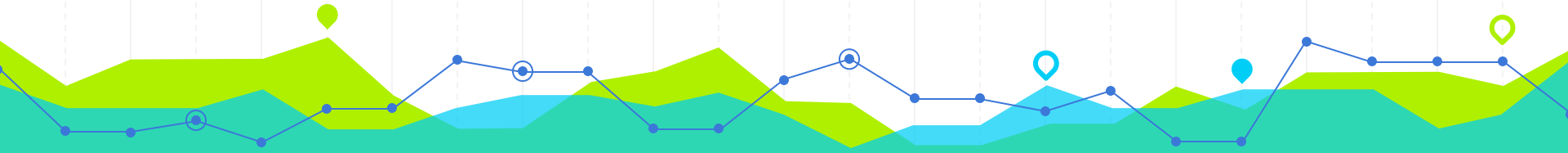
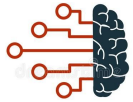


Heart Disease Prediction

INTRODUCTION

- Upon researching diseases, I came to know heart disease is ranked as the number-one cause of death in the United States.
- The number of deaths per 100,000 people is 211, which is frightening.
- Heart diseases are caused by a multitude of factors.
- So, is there a way people who are at risk could be alerted?

Machine Learning to the rescue

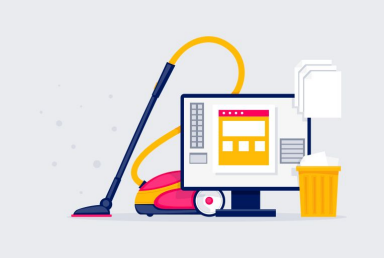


- Using historical anonymized heart patient data that contains their daily habits and characteristics, we could predict if someone's at risk of a heart disease.
- Supervised classification Machine Learning algorithms could be used to classify people who are at risk and not at risk for heart diseases.

Project Approach

1. Collect Data

2. Data Cleaning



3. Analysis & Modelling

4. Result

Collection of Data

- I acquired the dataset from kaggle.com.
- “Heart Disease Health Indicators Dataset” is the name of the dataset.
- The owner of this dataset is Alex Teboul.
- This dataset contains 22 columns (21 features and 1 target variable) and 253,680 rows of instances.
- The heart disease predicted using this dataset includes all the different types of coronary heart diseases.

Data Sample

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	HeartDiseaseorAttack	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	Diabetes	PhysActivity	Fruits	Veggies	HvyAlcoholConsump	AnyHealthcare	NoDocbcCost	GenHlth	MentHlth	PhysHlth	DiffWalk	Sex	Age	Education	Income
2	0	1	1	1	40	1	0	0	0	0	1	0	1	0	5	18	15	1	0	9	4	3
3	0	0	0	0	25	1	0	0	1	0	0	0	0	1	3	0	0	0	0	7	6	1
4	0	1	1	1	28	0	0	0	0	1	0	0	1	1	5	30	30	1	0	9	4	8
5	0	1	0	1	27	0	0	0	1	1	1	0	1	0	2	0	0	0	0	11	3	6
6	0	1	1	1	24	0	0	0	1	1	1	0	1	0	2	3	0	0	0	11	5	4
7	0	1	1	1	25	1	0	0	1	1	1	0	1	0	2	0	2	0	1	10	6	8
8	0	1	0	1	30	1	0	0	0	0	0	0	1	0	3	0	14	0	0	9	6	7
9	0	1	1	1	25	1	0	0	1	0	1	0	1	0	3	0	0	1	0	11	4	4
10	1	1	1	1	30	1	0	2	0	1	1	0	1	0	5	30	30	1	0	9	5	1
11	0	0	0	1	24	0	0	0	0	0	1	0	1	0	2	0	0	0	1	8	4	3
12	0	0	0	1	25	1	0	2	1	1	1	0	1	0	3	0	0	0	1	13	6	8
13	0	1	1	1	34	1	0	0	0	1	1	0	1	0	3	0	30	1	0	10	5	1
14	0	0	0	1	26	1	0	0	0	0	1	0	1	0	3	0	15	0	0	7	5	7
15	0	1	1	1	28	0	0	2	0	0	1	0	1	0	4	0	0	1	0	11	4	6
16	0	0	1	1	33	1	1	0	1	0	1	0	1	1	4	30	28	0	0	4	6	2
17	0	1	0	1	33	0	0	0	1	0	0	0	1	0	2	5	0	0	0	6	6	8
18	0	1	1	1	21	0	0	0	1	1	1	0	1	0	3	0	0	0	0	10	4	3
19	0	0	0	1	23	1	0	2	1	0	0	0	1	0	2	0	0	0	1	7	5	6
20	0	0	0	0	23	0	0	0	0	0	1	0	1	0	2	15	0	0	0	2	6	7
21	0	0	1	1	28	0	0	0	0	0	0	1	1	0	2	10	0	0	1	4	6	8
22	1	1	1	1	22	0	1	0	0	1	0	0	1	0	3	30	0	1	0	12	4	4
23	0	1	1	1	38	1	0	0	0	1	1	0	1	0	5	15	30	1	0	13	2	3
24	0	0	0	1	28	1	0	0	0	0	1	0	1	0	3	0	7	0	1	5	5	5
25	0	1	0	1	27	0	0	2	1	1	1	0	1	0	1	0	0	0	0	13	5	4
26	0	1	1	1	28	1	0	0	0	1	1	0	1	0	3	6	0	1	0	9	4	6
27	0	0	0	1	32	0	0	0	1	1	1	0	1	0	2	0	0	0	0	5	6	8
28	1	1	1	1	37	1	1	2	0	0	1	0	1	0	5	0	0	1	1	10	6	5

heart_data

Dataset Features (21)

- Blood pressure (high), cholesterol (high), Cholesterol Check within last 5 years, smoking, had stroke?, diabetes, difficulty in walking, age, sex, education, income, consumes fruits?, consumes vegetables?, exercise, alcohol consumption, BMI, Any Healthcare coverage?, Afford a doctor?, Mental Health, General Health, Physical Health

Dataset Feature Description

- Every feature is of the float data type.
- ‘BMI’ represents the exact BMI value (continuous).
- For the ‘Diabetes’ feature, 0 - no diabetes or only during pregnancy, 1 - pre-diabetes or borderline diabetes, 2 - yes diabetes. (0, 1, 2)
- ‘General Health’ is an ordinal variable (1 is Excellent -> 5 is Poor)

Dataset Feature Description (Continuation)

- The ‘Mental Health’ feature represents how many days during the past 30 days the person’s mental health was bad. (Values: 0 - 30)
- The ‘Physical Activity’ feature represents physical activity or exercise during the past 30 days. (Values: 0 - 30)
- The ‘Physical Health’ feature represents how many days during the past 30 days the person’s physical health was bad. (Values: 0 - 30)

Dataset Feature Description (Continuation)

- The ‘Age’ feature is ordinal. Value ‘1’ is 18-24 and all the way up to Value ‘13’ which is 80 and older. (5 year increments) (1 - 13)
- The ‘Education’ feature is ordinal with Value ‘1’ being never attended school or kindergarten only up to Value ‘6’ being college 4 years or more. (1 - 6)

Dataset Feature Description (Continuation)

- The ‘Income’ feature is ordinal with Value ‘1’ being less than $10,000 all the way up to Value ‘8’ being $75,000 or more. (1 - 8)
- All the other 12 features are binary. (0 or 1)

Data Cleaning 

- For data cleaning, I checked if there any nulls in the dataset. There was no nulls in the dataset.
- This dataset is highly imbalanced. So, I performed undersampling on the data to create balance for the data. I will explain more on undersampling of the data and its importance in the coming slides.

Explanation of the Project code
Custom Library (Class) Explanation

- For the coding part, I **used 10 third-party libraries namely, pandas, numpy, matplotlib.pyplot, seaborn, RandomOverSampler, xgb, and DecisionTreeClassifier, RandomForestClassifier, resample, and metrics from sklearn**.
- I created a custom library in a separate python file (ClassLibrary.py) which contains a single class, ‘modelling’ with several methods.

- A ‘.csv’ data file is passed as an argument to the object created using this class and the constructor creates a pandas dataframe using the ‘.csv’ data file.
- This class also contains methods as follows to:
- return the dataframe (getData)
- Check for nulls in each column (checknull)
- Conduct statistical anlaysis on each column (stats)
- Print and also plot the correlation matrix (corr_matrix)

- Perform undersampling on unbalanced data (balancing_data)
- Perform a manual train and test data split in an 85%:15% ratio (train_test_data_split)
- Return the evaluation metrics (performance) of each Machine Learning Model (evaluation_metrics)
- Plot the confusion matrix (plot_confusion_matrix)

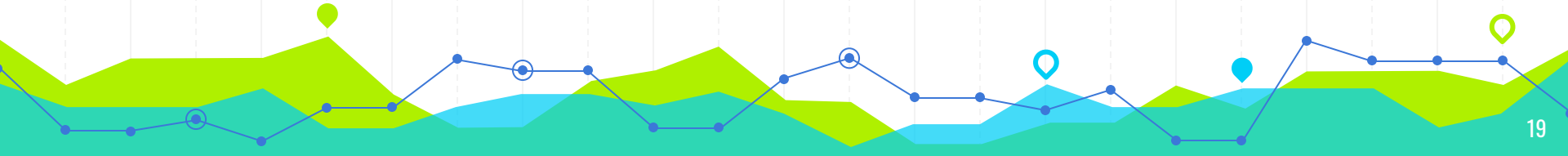
Main Program Explanation

- All third party libraries and the custom library is imported.
- An object of the class modelling, data is created using the ‘heart_data.csv’ file.
- The data file in the form of pandas dataframe is returned.
- This dataframe, ‘df’ is checked for nulls and statistically analyzed using the describe() method. The feature names are changed for better comprehension.

- X (feature set data columns) and Y (Target data column) data are created on the raw unbalanced data.
- Correlation matrix for this data is plotted.
- The distribution of the target (y) feature values are checked to observe the balance of the data. There are 229,787 values of ‘0’ and 23,893 values of ‘1’ in the target feature. Since the data is unbalanced, undersampling (removing rows of the majority class for equal distribution) of the target feature value ‘0’ is performed.

- After undersampling, both classes, ‘0’ and ‘1’ have 23,893 rows of instances each.
- Train and test data split (85%:15% ratio) is done on both the unbalanced and the balanced data.
- Three Machine Learning models namely, **Decision Tree, XGBoost, and Random Forest Classifier** are used. Once train data is used to fit the data, the test data is used for prediction.

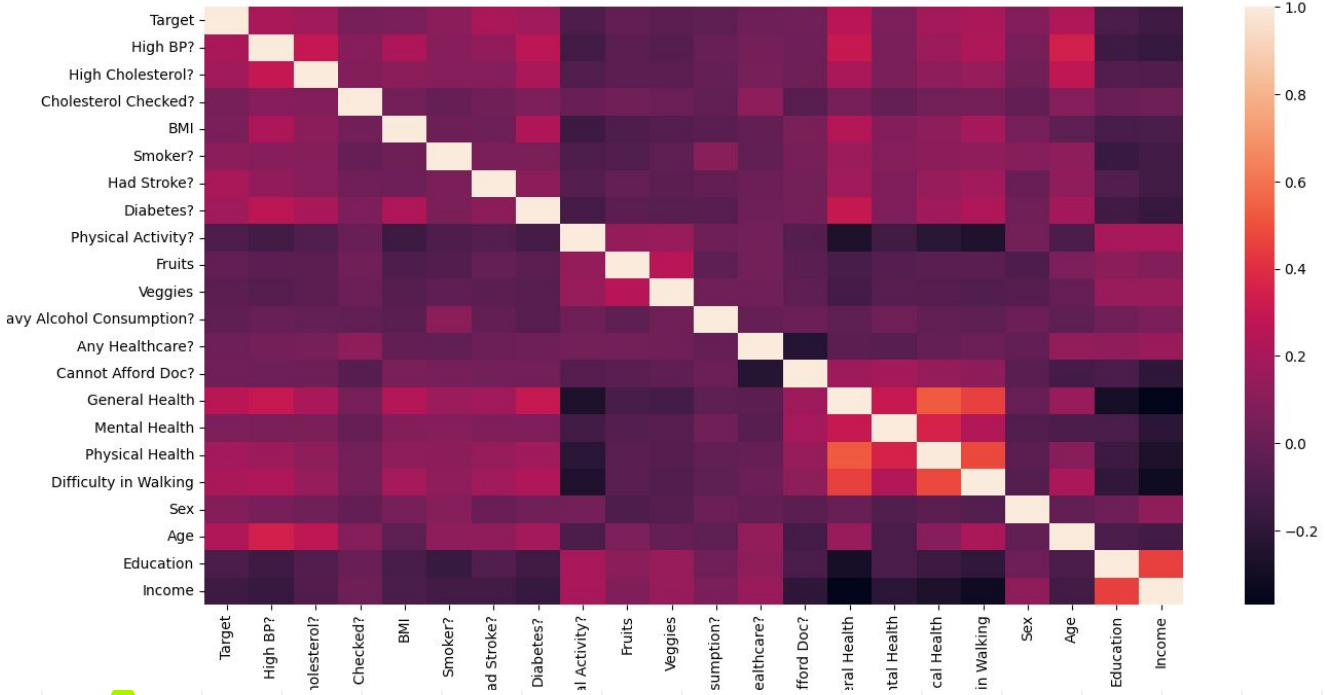
- The evaluation metrics accuracy, precision, recall, and f-1 scores are calculated and a dictionary is created for these metrics.
- A confusion matrix is plotted for all the models.
- The dictionaries of evaluation metrics are converted into a list and then a dataframe.
- This dataframe is written into a file, 'Evaluation_of_each_Model.txt', and then output using File output.



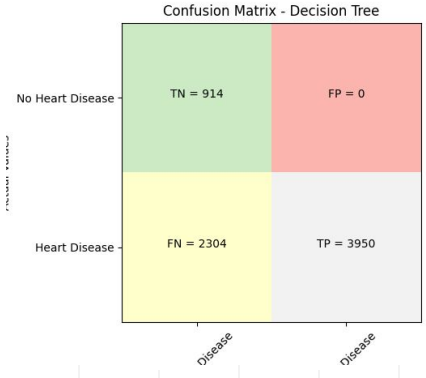
Model Results

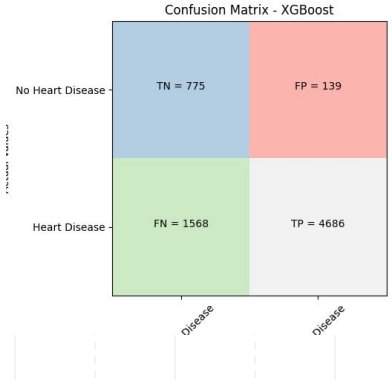
	Accuracy	Precision	Recall	F-1 score
Decision Tree	0.68	1.00	0.63	0.77
XGBoost	0.76	0.97	0.75	0.85
Random Forest Classifier	0.78	1.00	0.75	0.86

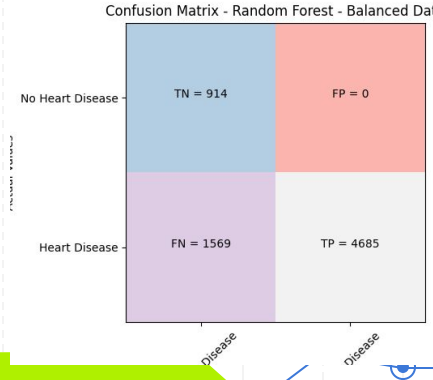
Correlation Matrix Plot

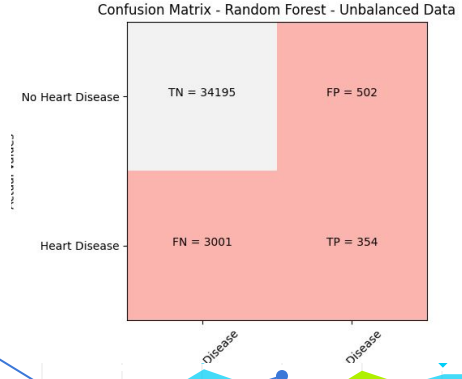


Confusion Matrices Plot









Observations from Analysis

- It is crucial that the predictions are not falsely classified as positive or negative. False positive would mean the person will be medically treated when that person does not have any heart disease. False negative would lead to the person not getting medical care at all.
- Because of this, heart disease prediction should concentrate on getting a higher precision and recall scores.

- As a result, it is important to concentrate on the f-1 score (harmonic mean of precision and recall) rather than the accuracy.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

TP = True positive

TN = True negative

FP = False positive

FN = False negative

- Since the raw data is unbalanced with 229,787 values of ‘0’ and 23,893 values of ‘1’ in the target feature, it is important the data is balanced.
- Without the use of Machine Learning, if we predict ‘0’ for all the instances, we get an accuracy of 90.58%. But predicting the patients at risk (‘1’) is pivotal. This is the reason data has to be balanced.

- We see the implications of performing Machine Learning on the unbalanced data. Among the three models, Random Forest Classifier is the best performing model.
- Random Forest Classifier was used on both the unbalanced and balanced data. The accuracy of the unbalanced dataset and the balanced data are 90% and 78% respectively. Looking at just the accuracy of these models might be misleading.

- The F-1 scores of the unbalanced dataset and the balanced dataset are 17% and 86% respectively. The recall score on the unbalanced data is a pathetic 10% (poor at predicting positive heart disease cases). As I mentioned earlier, F-1 score is crucial for heart disease prediction. This explains the importance of balancing data.
- The F-1 scores of the Decision tree, XGBoost, and the random forest classifier models are 78%, 85%, and 86% respectively. Thus, the random forest classifier model is better at predicting heart diseases.

Insights and Takeaways

- It is crucial to focus on the f-1 score (harmonic mean of precision and recall) for cases like heart disease prediction where the cost of false positive and false negative is huge.
- Even though the accuracy on the unbalanced data was high (91%), it was poor at predicting the positive cases of heart disease, which is the ultimate goal of the model.

- The previous insight proves the importance of balanced data and why we should focus on the f-1 score rather than the accuracy.
- As per the first insight, the random forest classifier was the best performing model among the three models with 78% accuracy, 100% precision, 74% recall, and 85% as the F-1 score.

THANKS!