Technical Project Report

Sunil Sharma — L00179485

Title:	Big Data Analytics, Project Report
Author:	Sunil Sharma
Supervisor:	Professor Shagufta Henna
Degree:	MSc in Computing in Big Data Analytics

Title

Heart Disease prediction using Predictive Analytics based on Design Thinking.

Project Overview

Having collaborated with medical professionals and data scientists, our team undertook the development of a heart disease prediction model. The final product empowers individuals to assess the likelihood of experiencing heart disease. We meticulously identified user needs and constructed an empathic map through extensive user discussions, subsequently addressing challenges to deliver a comprehensive solution.

Incorporating Design Thinking Principles, our approach centered on user-centric development and rapid prototyping. The technology stack chosen for our machine learning pipeline comprises Spark, Spark MLlib, and SynapseML, with infrastructure supported by Azure Databricks, Azure Data Lake Storage, and Azure Data Factory for scheduled yearly updates. This allows the model to access updated BRFSS data, as the BRFSS conducts surveys annually.

The culmination of our efforts yields a robust model, affording users the capability to predict their heart health based on input provided to the system. This completed project underscores our commitment to innovative problem-solving and user-focused design principles.

Data Description

A number of disorders affecting the heart and blood arteries are together referred to as heart diseases. It is another term for heart disease. It still poses a serious threat to world health and accounts for a sizable number of fatalities each year. The CDC (United States) conducts an annual survey that gathers health data from over 400,000 individuals and provides a comprehensive dataset for the study and analysis of heart disease indicators. This specific initiative aims to shed light on the intricacies of heart disease by providing significant information on its causes, risk factors, unhealthy lifestyle choices, symptoms, and preventive measures.

Ampathy Map

0.1 Says(Users perspective)

1. Why is heart disease becoming more prevalent? Recognising the underlying causes and emerging trends that are fueling the rise in cardiac illness.

2. What behaviours and risk factors exist? determining the lifestyle decisions and actions that have a major influence on heart health.

3. What benefits can artificial intelligence (AI) technologies offer over conventional diagnosing techniques? appreciating the advantages of AI, including its accuracy, efficiency, and early detection. Who can enhance and add to the existing diagnostic tools? assessing the potential benefits of AI for current diagnostic techniques and instruments.

0.2 THINKS (User's Thought Process):

1. Supports clinical decisions by taking into account the accuracy and reliability of the prediction model. confirming the clinical reliability and compliance of the predictive model with medical criteria.

2. How might accurate forecasting affect patient care and the efficiency of healthcare delivery? Taking into account how precise predictions affect the efficacy of medical care and treatment results. Considering the impact of accurate predictions on treatment outcomes and healthcare effectiveness.

3. In what ways may the AI solution enhance and support the existing diagnostic tools? assessing the potential benefits of AI for current diagnostic techniques and instruments.

0.3 FEELS (User's Emotional Experience):

1. Has an obligation to give patients the finest treatment possible. Feeling a sense of responsibility to provide optimal treatment based on accurate predictions.

2. Is grateful for the advancement of technology but finds fault with the current diagnostic methods. expressing annoyance at the constraints that exist now and recognising technological progress.

3. Look forward to relief and satisfaction as a result of being able to diagnose and treat patients earlier and preventing major heart problems. Looking forward to relief and contentment from early detection and avoidance of major heart problems.

0.4 DOES (User's Actions):

1. Makes a heart disease prediction based on the patient's health history, lifestyle choices, and current immunisation status. aggressively utilising medical data to forecast the risk of cardiac conditions. The item collaborates with data scientists and AI experts to optimise the model for improved accuracy and relevance in a therapeutic setting. working together with medical professionals to improve the predictive model for therapeutic uses.

2. Educates physicians on the benefits of AI-assisted diagnosis and how it could improve their overall patient outcomes. advocating for better health outcomes by raising awareness of the advantages of AI-assisted diagnostics.

1 Stage 2: Define—State Your Users' Needs and Problems:

1. The necessity to comprehend the rise in cardiac illnesses in light of risk factors and behavioural patterns was identified.

2. Acknowledged the value of AI solutions in raising the precision and effectiveness of diagnostics.

3. Acknowledged that a user with no prior experience with AI should find it straightforward to utilise the solution.

4. Acknowledged the need for accurate forecasting to improve patient care and the efficacy of healthcare.

2 Stage 3: Ideate—Challenge Assumptions and Create Ideas:

1. Using domain expertise and feature selection techniques, as well as conversations with medical professionals and data scientists, the dataset was examined to identify the most influential traits that were linked to health disorders.

2. Investigated different AI models and methods that could be used to forecast heart disease signs based on medical symptoms. The item Innovative concepts for combining AI with conventional diagnostic techniques for a comprehensive approach

3. Brainstormed concepts for a final solution that the user might potentially apply.

4. In model creation and optimisation, assumptions were challenged by taking the user's perspective into account.

3 Stage 4: Prototype—Start to Create Solutions:

3.1 Acquiring Dataset

The dataset was obtained from Kaggle. The BRFSS dataset is accessible through Kaggle in CSV format. We are mounting the dataset to our DBFS using Azure Data Lake Storage, and we can access the data from Databricks notebook using the Azure Blob File System (ABFS).

3.2 Pre-processing:

carried out extensive data cleaning, dealt with missing values, and changed features for feature engineering and model compatibility.

3.2.1 Data Cleaning and handling missing values

Changing the column names Coverting particular columns which are age_category, other categorical column values from Yes and No to 1 and 0 respectively After reading the file and using inferschema, we encountered all our features are showing as string type , to encounter this issue while using the inferschema we came to a conculsion of changing the features values to numerical values by using indexing after reading the data with inferred schema. To clean the file we are checked the file for any Na vales and duplicate records and dropping these two. there were no values in the dataset.

Some details we got from EDA

1. Count after removing duplicates 246022,
2. percentage of person with heart disease is 91
3. Total categorical columns are 32.
4. Total numerical variables are 6.

No	Name	Type	Definition
1	State	Class	State of residence.
2	Sex	Class	Gender of the individual.
3	GeneralHealth	Class	General health status.
4	PhysicalHealthDay	float64	total number of days with physical health issues.
5	MentalHealthDay	float64	total number of days with mental health issues.
6	LastCheckupTime	Class	last health checkup time.
7	PhysicalActivitie	Class	Type of physical activities.
8	SleepHour	float64	Number of hours of sleep per day.
9	RemovedTeet	Class	Information about removed teeth.
10	HadHeartAttac	Class	Indicates if the individual had a heart attack.
11	HadAngin	Class	Indicates if the individual had angina.
12	HadStrok	Class	Indicates if the individual had a stroke.
13	HadAsthm	Class	Indicates if the individual had asthma.
14	HadSkinCancer	Class	Indicates if the individual had skin cancer.
15	HadCOPD	Class	Indicates if the individual had COPD.
16	HadDepressiveDisorder	Class	Indicates if the individual had a depressive disorder.
17	HadKidneyDisease	Class	Indicates if the individual had kidney disease.
18	HadArthritis	Class	Indicates if the individual had arthritis.
19	HadDiabetes	Class	Indicates if the individual had diabetes.
20	DeafOrHardOfHearing	Class	Indicates if the individual is deaf or hard of hearing.
21	BlindOrVisionDifficulty	Class	Indicates if the individual has blindness or vision difficulty.
22	DifficultyConcentrating	Class	Indicates if the individual has difficulty concentrating.
23	DifficultyWalking	Class	Indicates if the individual has difficulty walking.
24	DifficultyDressingBathing	Class	Indicates if the individual has difficulty in dressing or bathing.
25	DifficultyErrands	Class	Indicates if the individual has difficulty running errands.
26	SmokerStatus	Class	Smoking status of the individual.
27	ECigaretteUsage	Class	E-cigarette usage status.
28	ChestScan	Class	Information about chest scan.
29	RaceEthnicityCategory	Class	Race or ethnicity category
30	AgeCategory	Class	Age category
31	HeightInMeter	float64	Height in meters
32	WeightInKilogram	float64	Weight in kilograms
33	BMI	float64	Body Mass Index
34	AlcoholDrinkers	Class	Alcohol drinking status
35	HIVTestingst	Class	HIV testing status
36	FluVaxLast12mon	Class	Flu vaccine in the last 12 months
37	PneumoVaxEver	Class	Ever had a pneumonia vaccine
38	TetanusLast10Tyear	Class	Tetanus vaccine in the last 10 years
39	HighRiskLastYearC	Class	High-risk condition in the last year
40	CovidPositive	Class	COVID-19 positive status

Fig 1: Data Table

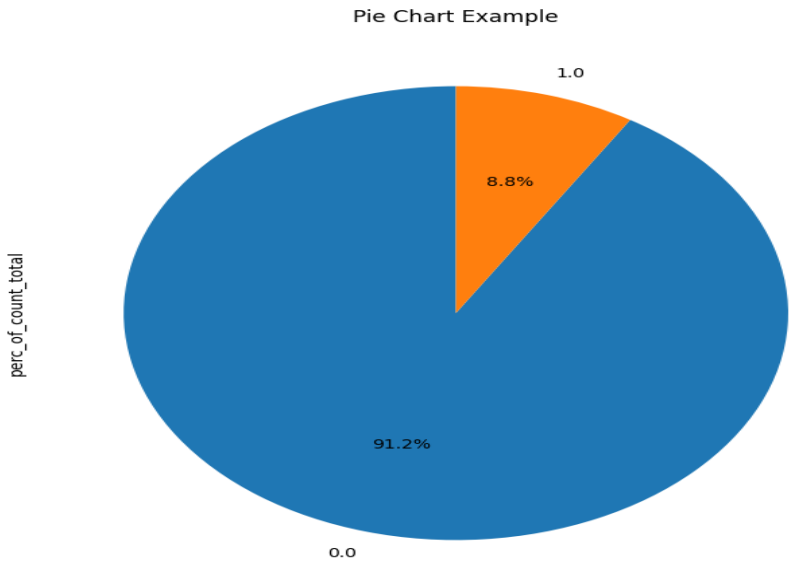
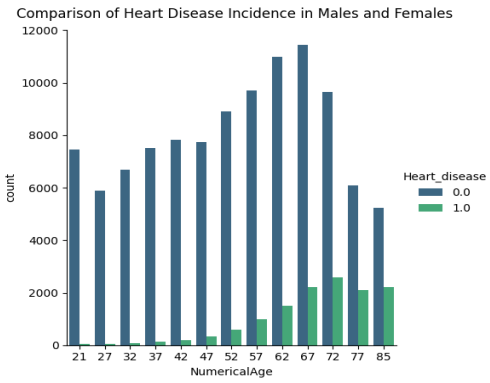


Fig 2: Heart Disease percentage

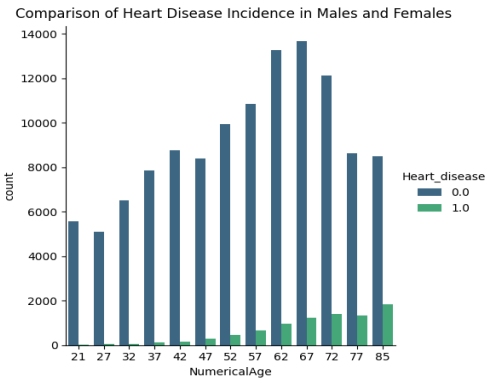
As our main object is to predict the possibility of having a heart disease, so we are making a feature named "heart_disease" which will be true if any feature associated to heart disease is true, in our data set their are two columns which are heart disease(HadAngina, HadHeartAttack).

Goals achieved

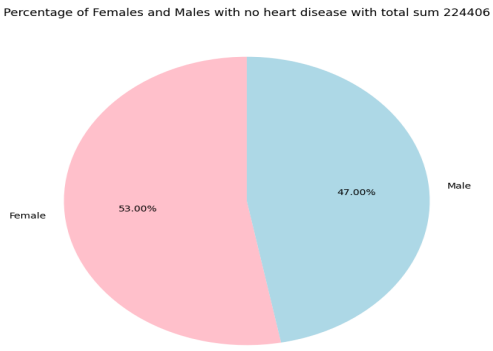
1. Comapring Heart disease occurences between male and female as age increases please see fig 2.2 and 2.3
2. To find the Gender-based difference in the prevalence of heart disease across various age groups, and how does age affect the chance of getting heart disease? see Fig 3
3. Determine the main factors affecting the prevalence of heart disease and corr between them please see Fig4.
4. Examine differences in heart disease risk amongst different ethinicity groups. please see Fig6
5. Comparing ratio of BMI against heart disease for male and female. please see Fig7
6. Examine Heart Disease with habits like smoking and Physical activities see fig 8



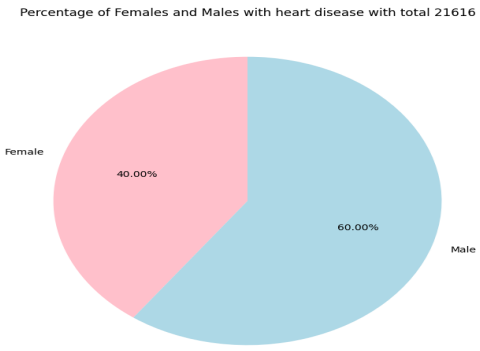
(a) Fig2.2 For females



(b) Fig 2.3 For males



(a) Fig3.1 with no heart disease



(b) Fig 3.2 with no heart disease

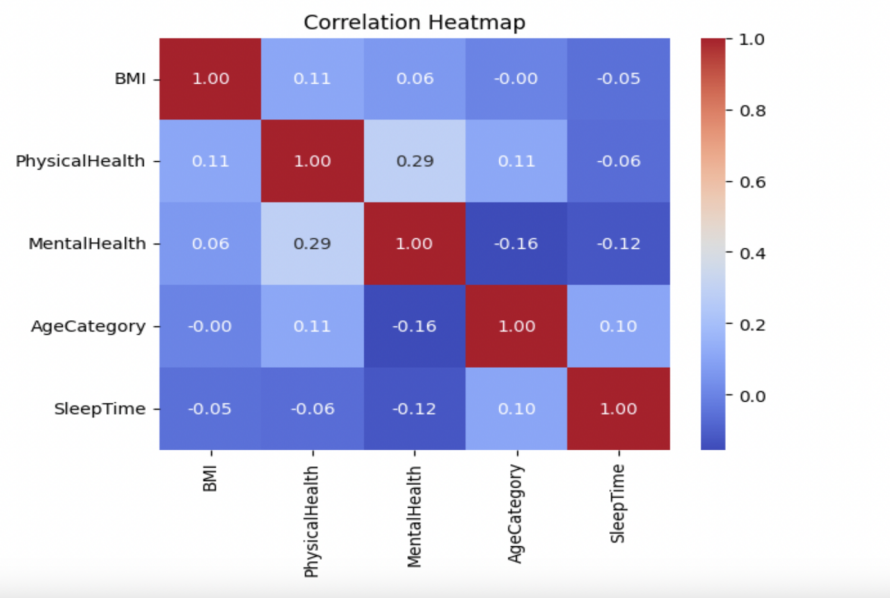


Fig 4: Correleation between risk factors and illness

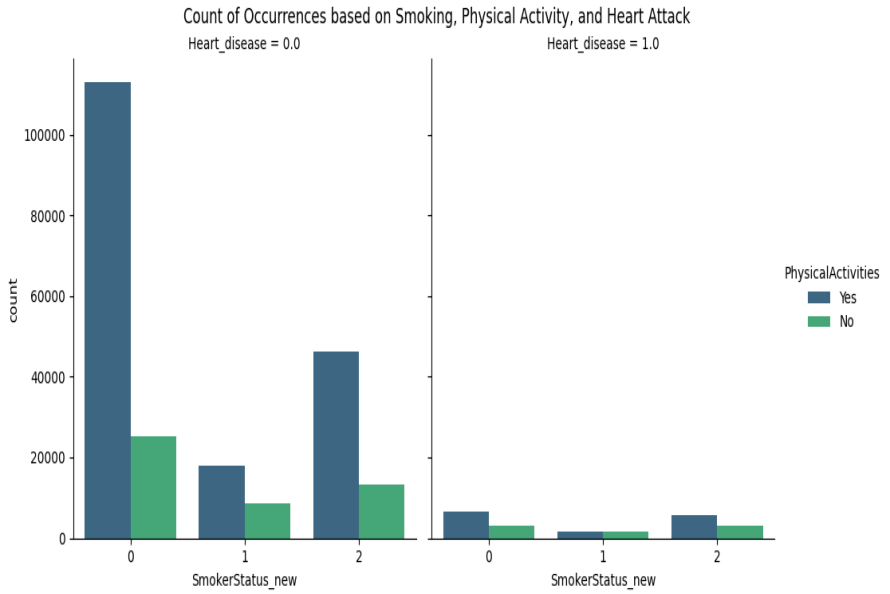


Fig 5: Heart Disease based on physical activity and smoking status

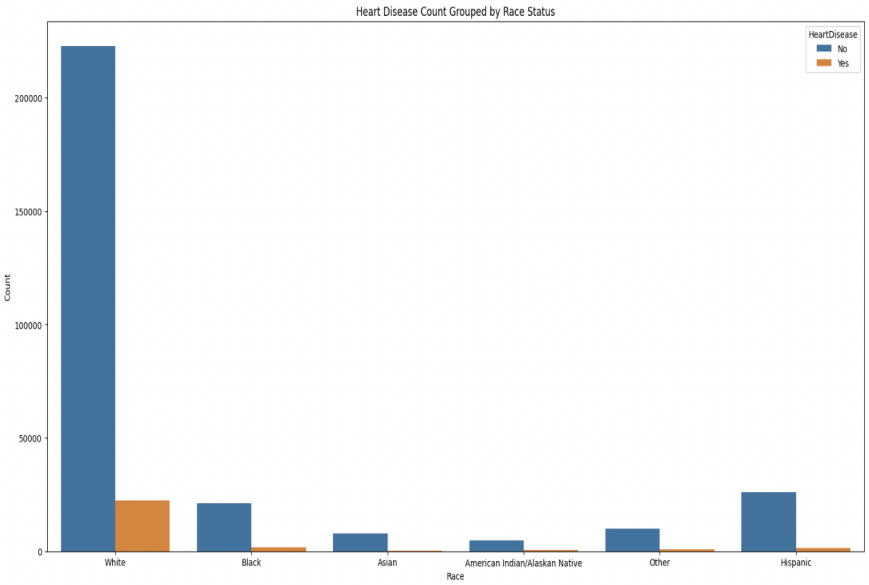


Fig 6: Heart disease based on demographics

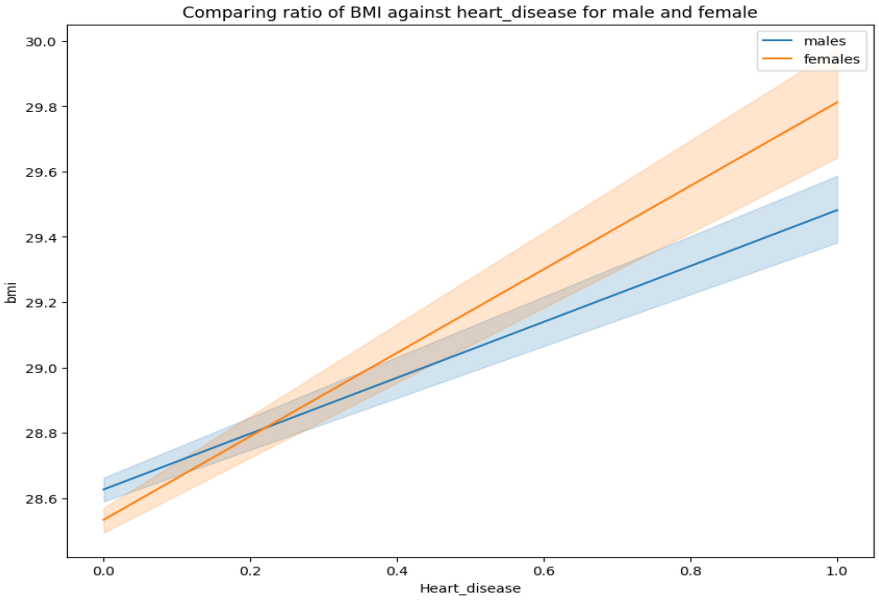


Fig 7: BMI and Heart Disease for Male and Female

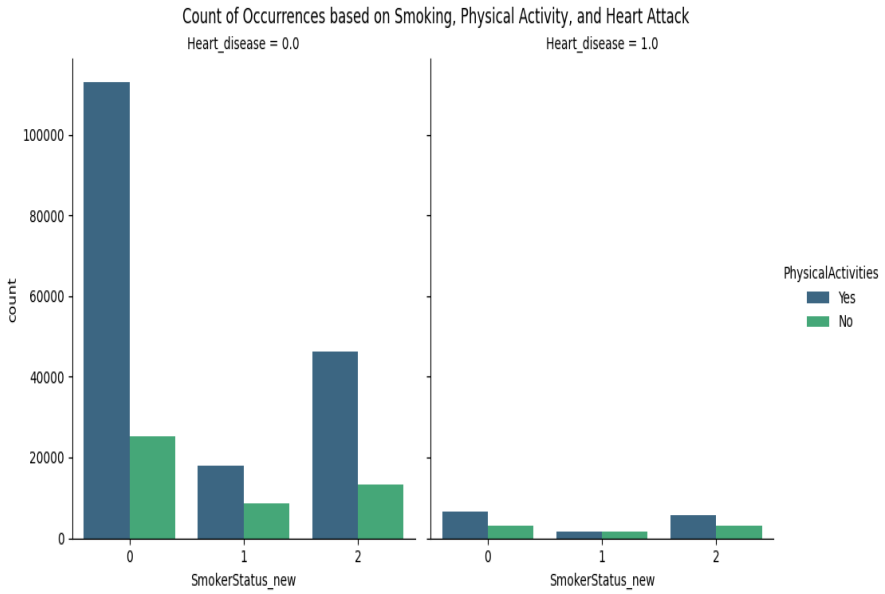


Fig 8: Relationship Between Heart disease with smoking and physical activity

As we can we can see the class imbalance in out Heart_disease variable which is our output Y. We have used oversampling(involves increasing the number of instances in the minority class to balance the class distribution.) which is a resampling technique, to solve our class imbalace, and overfitting problem.

3.2.2 Feature Engineering

Feature Transformation: A pivotal stage in the machine learning pipeline is feature transformation, a process that involves modifying or generating new features from existing ones to enhance model performance. This step is crucial for addressing issues like non-linearity, multicollinearity, and outliers, thereby improving the interpretability of the data. Various techniques were employed, including:
MinMaxScaler: Scaling numerical features to a specified range (e.g., [0, 1]). StringIndexer: Converting categorical string values into numerical indices. Vector Assembler: Combining multiple feature columns into a single vector column.
Given our dataset, where a significant portion of features is categorical, we utilized StringIndexer. To expedite this process, a function was created to segregate columns based on their types, allowing us to specifically target and apply StringIndexer to categorical features. Additionally, for date and time features, we calculated the time difference between the date-time column and a reference point or future event.

to use our date and time features we are calculating the time difference between

the date and time column and a reference point or future event.

Feature Selection: Another critical stage is feature selection, aimed at choosing a subset of relevant features to construct a model. This step reduces overfitting, improves interpretability, and enhances model performance. Relevant features were identified based on domain knowledge and user requirements. Various feature selection methods were employed, including:

Filter methods: Such as variance threshold and correlation-based methods. Dimensionality Reduction Techniques: Principal Component Analysis (PCA). Information Gain Calculation: To determine the importance of each feature. Following these methods, selected features were passed through VectorAssembler to create a single vector column, aligning with PySpark's MLlib requirement for input features.

Model Implementation: For model development, Spark MLLib and SynapseML were selected. The model training process involved exploring the predictive capabilities of different classifiers. Two classifiers were used: Random Forest (Bagging) and XGBoost (Boosting). Stacking was employed to combine predictions from these models using a meta-model like logistic regression. Unlike bagging, stacking utilizes the full training set, proving to be an effective technique for enhancing prediction accuracy beyond individual models.

3.3 Model Implementation

Model: Selected Spark MLLib and SynapseML for machine learning model development.

Model Training: Trained initial models to explore their predictive capabilities, models, we are using below 3 classifier models 1: Random Forest(Bagging) 2: GradientBoosting(Boosting) and 3:Logistic Regression.

How Boosting work? Boosting is an ensemble learning technique where the weak learners are too simple and tend to suffer from high bias. In the Boosting panel of the figure above, the base models are decision trees with only one level, a decision stump. Decision stumps can only make a decision based off of one feature at a time, causing them to underfit the data substantially.

Boosting is a sequential learning technique where each of the base models builds off the previous model. Each subsequent model aims to improve the performance of the final ensembled model by attempting to fix the errors in the previous stage, as shown in Fig9.

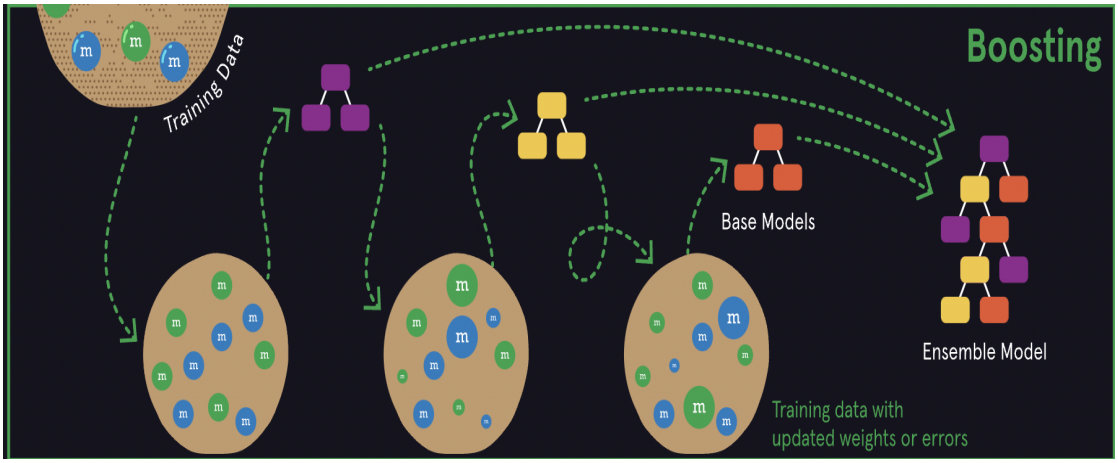


Fig 9: Relationship Between Heart disease with smoking and physical activity

In the Boosting panel of the figure, you may notice that some of the candies atop the cookies are larger than the others. These particular training instances were misclassified by the previous decision stump and are therefore given more weight by the next decision stump. This is one method in which boosting methods may learn from their mistakes.

Training Base Estimators We can select from combinations of different base estimators, such as a logistic regression model in combination with a decision tree. We could additionally select models of the same learning algorithms, but with different parameters, such as multiple decision trees with varying depths. The number of estimators is arbitrary, so it's good practice to explore how different combinations behave.

This introduces a problem, however. The estimators would be making predictions on data used in training. This puts our model at risk of overfitting. To avoid this, we use K-Fold Cross Validation as described next.

K-Fold Cross-Validation is a method used for assessing the performance and generalization of a model. It is not specifically for hyperparameter tuning.

Consider 10 segments (or folds) and a stacking model that uses a logistic regression model and a decision tree model. Each estimator can be trained using data from 9 of the segments, and make predictions on the excluded 10th segment. We then append the predictions as new features to that 10th segment. Now 1/10th of the training data has two new features: one is the prediction made by the logistic regression model and the other is the prediction made by the decision tree model.

In K-Fold we have taken n_splits = 3, shuffle= True and random state = 42 and we are getting a score of 91

We want to do the same with the other 9 segments, so we rotate the excluded

segment and repeat this process until all training data points are augmented with new features. The end result is a prediction made on each training sample, without having seen the sample during the training process.

Feature Augmentation In our stacking setup, the base estimators need to be trained to make predictions on our training data. The prediction of each estimator will be appended to the corresponding data sample as a new feature. We thus augment the training data set with this additional information. The augmented training set is used by our later-stage stacking model to make the final prediction.

Model Testing: also known as model evaluation or validation, is a critical step in the machine learning pipeline. It involves assessing the performance of your trained model on new, unseen data to understand how well it generalizes to real-world scenarios. We will use the trained model to make predictions on the test dataset. then we will get accuracy, we have used MulticlassClassificationEvaluation which is a class commonly used in machine learning frameworks, such as Apache Spark's MLlib, for evaluating the performance of a multiclass classification model. This evaluator provides metrics to assess how well a model is performing in terms of classification accuracy, precision, recall, and F1 score.

We resampled the data to equal the output label Y, 91.2 to 8.8 ratio to 50 -50. as shown in Fig 10

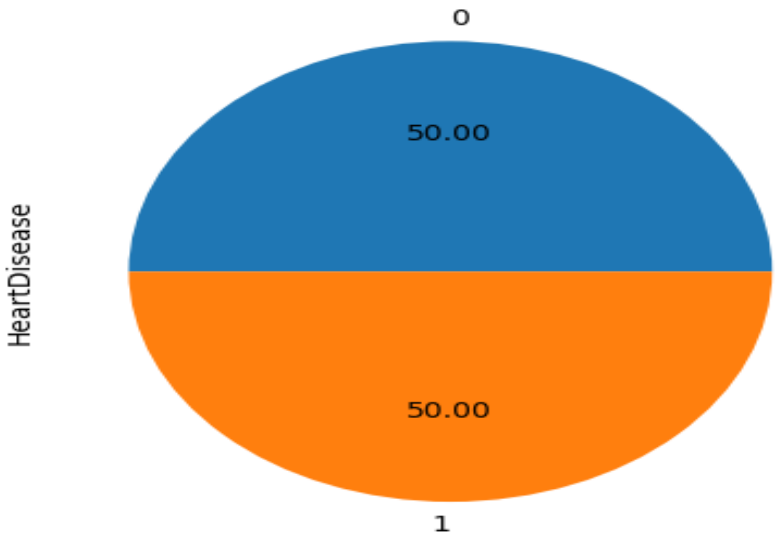


Fig 10: Heart Disease percentage after resampling

Trained the models again with the resampled data, and recalculated the accuracy,

final accuracy we are getting after training models. please see fig 11 and 12.

Model	Accuracy	Precision	Recall	Recall
Logistic Regression	0.7366	0.7323	0.7389	0.7389
Random Forest	0.8735	0.8752	0.8764	0.8865
LightGBM	0.83	0.82	0.82	0.84

Fig 11: Accuracy, precision, recall percentage

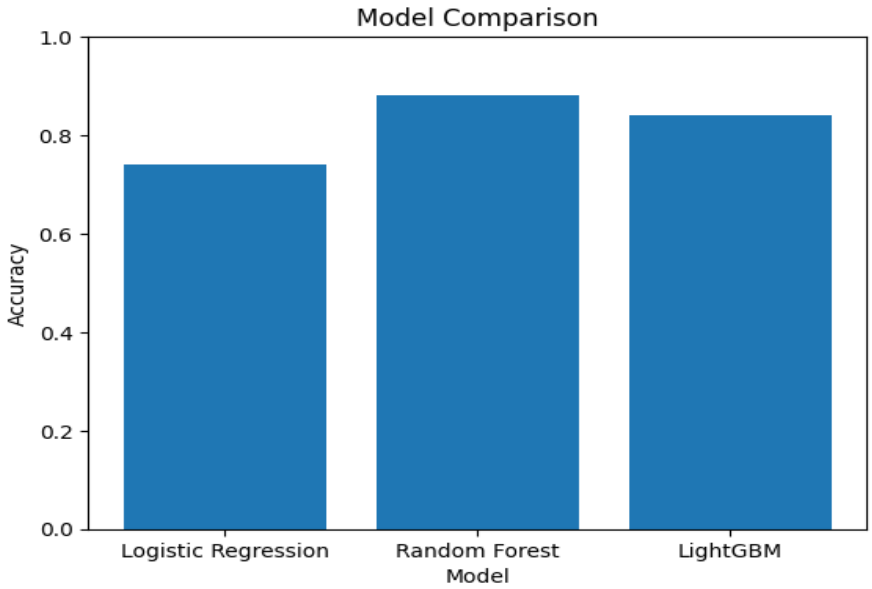


Fig 12: Accuracy comaprision

Hyperparameter Tuning: After getting the result we are implementing hyperparameter tuning to adjust the model configuration suitable for our dataset, and train our model again based on our adjusted configuration.

Create an Azure Machine Learning workspace and deployed the machine learning model into Azure Azure Container Instance (ACI)

4 conclusion and Future works:

The project successfully implemented using design thinking prinicples, empathy map based on user, identified main factors affecting heart disease prevalence, ex-

plored correlations, examined gender-based differences, and built machine learning pipeline predictive models. Design Thinking principles enriched the analysis, ensuring user perspectives were integrated. Future work involves refining models based on ongoing user feedback and addressing emerging insights, and making it available for users to predict the heart disease possibilities by manually entering the details.

References

[1] Databricks documentation https://docs.databricks.com/en/index.html

[2] Azure Documentationhttps://azure.microsoft.com/en-us/products

[3] Ensemble Learning https://www.geeksforgeeks.org/ensemble-methods-in-python/

[4] Class notes

[5] Codecademy articleshttps://www.codecademy.com/learn