Heart Attack Cases in Indonesia - Predictive and Prescriptive Analytics using the Hadoop ecosystem

Proag Tanvee, 2422814
University of Mauritius

Abstract

This study is an attempt to create a predictive and prescriptive model of the incidence of heart attack using datasets of detailed health profiles of individuals in Indonesia. These include the general and medical attributes of the individual and syndromes they may be suffering from. The Random Forest Classifier classification method is used to predict the chance of heart attack and the performance is evaluated with an accuracy of 0.6. The Hadoop ecosystem hosted on the Google Cloud Platform used Hadoop for distributed storage and processing, Apache Hive for SQL-based querying, Apache Spark for predictive and prescriptive analytics and Apache Zeppelin for visualization.

1. Introduction

1.1 Data Analysis in Medicine

The healthcare industry is facing challenges due to the lack of quick and efficient data analysis. Medicine today generates vast amounts of data, including patient records, lab results and clinical observations but much of this information remains underutilized. The slow pace at which this data is processed and analyzed can have serious consequences impacting patient care and healthcare efficiency. One major issue is delayed decision-making. Doctors and medical professionals often rely on their experience and intuition to make critical decisions but without real-time data analysis, there is a risk of overlooking patterns or trends that could indicate a more accurate diagnosis or a better course of treatment.

1.2 Problem Statement

One of the most critical areas where data-driven approaches have shown potential is in the prevention and early detection of cardiovascular diseases—particularly heart attacks, which remain a leading cause of death globally. With the availability of health-related data, predictive and prescriptive analytics enable healthcare professionals to uncover hidden patterns, assess risk factors and make decisions based on individual patient profiles. By using structured datasets that capture demographic, lifestyle and clinical indicators, a model that forecasts the likelihood of heart attacks and provides preventative measures can be built. The research question is "How can clinical, behavioral and environmental risk factors be used to accurately predict the likelihood of heart attacks among individuals in Indonesia and what preventive insights can be drawn to reduce future cases?" This question reflects how features can predict heart attack outcomes and prescribe the risk factors for prevention based on the data.

2. Data

This study focuses on heart attack cases in Indonesia, using a dataset on heart attack incidence from Kaggle, comprising 158,355 records and 28 features. The features are individuals' details about demographics (age, gender, region, income level), clinical risk factors (hypertension, diabetes, cholesterol level, obesity, waist circumference, family history), lifestyle and behavioral factors (smoking status, alcohol consumption, physical activity, dietary habits), environmental and social factors (air pollution exposure, stress level, sleep hours) and medical screening and health system factors (blood pressure systolic, blood pressure diastolic, fasting blood sugar, cholesterol hdl, cholesterol ldl, triglycerides, EKG results, previous heart disease, medication usage, participated in free screening). Heart attack (yes/no) is the target variable. The dataset is structured to support machine learning models for predicting heart attack risks, public health research and epidemiological studies. Through data analysis, this project aims to identify the most influential risk factors associated with heart attacks and propose strategies that can support early intervention.

3. Tools

The tools and libraries used in this project are essential for performing data analysis, preprocessing and model training.

3.1 Dataproc

Dataproc is a fully managed cloud service provided by Google Cloud Platform that is used for running big data processing frameworks like Apache Spark, Apache Hadoop, Apache Hive etc. It is used for data processing at scale, data analysis, machine learning and interactive data exploration.

3.2 Hadoop Ecosystem of Dataproc

Google Cloud Storage is the storage layer. Hadoop YARN manages resource allocation across cluster nodes. In Dataproc, a bucket is created to store files like the dataset, the scripts and the processed output are saved in it too. A cluster in Google Cloud Dataproc is a group of virtual machines that work together to run big data processing tasks using Apache tools. Each cluster has a master node that coordinates the job execution and worker nodes that do the data processing. Jobs of type Hadoop or Hive or Spark are submitted along with their scripts.

3.2.1 Apache Hive

Apache Hive is a data warehouse software project built on top of Apache Hadoop that facilitates reading, writing and managing the large dataset using SQL-like queries.

The HiveQL script uploaded in the bucket is used to perform data processing workflow on the dataset. A managed table called "heart attack raw" is set up with a comprehensive schema defining 28 columns of all the features. The table is configured to read comma-delimited text files. Hive translates most queries (like SELECT, INSERT, CREATE VIEW, etc.) into MapReduce jobs. After dropping any existing version of the table to ensure a clean start, data is loaded directly from the Google Cloud Storage bucket into the newly created table. Then, a table property is set to skip the header line of the CSV file, ensuring only actual data rows are processed. This raw data is transformed by creating a cleaner, more analysis-ready view called "heart attack cleaned". This view applies numerous transformations to standardize and encode the data - converting text values like "yes" and "no" to binary numeric values (1 and 0), standardizing text fields to uppercase and creating ordinal encodings for multi-level categorical variables like cholesterol level (mapping "high" to 2, "borderline" to 1 and others to 0). Finally, the cleaned dataset is written to Google Cloud Storage in the Parquet format which is optimized for analytical queries. This Parquet file is to be analysed later. This way, Hive performs ETL (Extract, Transform, Load) to injest raw data, transform it through SQL operations and then export in an optimized format for further processing.

3.2.2 Apache Spark

Apache Spark is an open-source, distributed computing system designed for big data processing and analytics. Spark works by distributing data processing tasks across multiple nodes in a cluster using a directed acyclic graph (DAG) execution engine that optimizes workflows. PySpark is the Python API for Apache Spark.

The PySpark script uploaded in the bucket is used to implement a complete machine learning pipeline for heart attack prediction using the data that was previously processed by Hive. First, a Spark Session which is the entry point for any Spark functionality is initialised and the Parquet-formatted data generated by the Hive preprocessing steps is loaded. Since the Parquet file contains generic column names (_col0, _col1, etc.), these are first mapped to meaningful feature names. Most importantly, a sophisticated machine learning pipeline is built using Spark ML. First, the categorical and numerical columns are identified based on their data types, then a series of preprocessing stages are created: categorical variables are processed through String Indexers (converting strings to numeric indices) and One Hot Encoders (transforming indices into one-hot encoded vectors) while numerical features are kept as is. All these features are then combined into a single vector using Vector Assembler and subsequently standardized with Standard Scaler to ensure all features contribute equally to the model. Then, a Random Forest Classifier is configured with 200 trees, a maximum depth of 5 and a fixed random seed for reproducibility.

After defining the pipeline, the data is split into training (80%) and test (20%) sets, the pipeline is fit to the training data and the model's performance is evaluated on the test set using the area under the ROC curve as the evaluation metric. Finally, the predictions are enriched by adding a risk category column (High Risk/Low Risk) and these are saved to the Cloud Storage bucket.

3.2.3 Apache Zeppelin

Apache Zeppelin is an open-source web-based notebook platform used for interactive data analytics and visualization. Zeppelin integrates with Apache Spark making it a powerful tool for data exploration, machine learning and real-time collaboration.

4. Data Analytics Lifecycle

4.1 Apache Hive

4.1.1 Data Ingestion

Hive is used to ingest data from the source file in the Google Cloud Storage bucket into Hive's managed environment. This happens in the LOAD DATA INPATH statement where data from the CSV file stored in the bucket is loaded into the 'heart attack raw' table.

4.1.2 Data Preprocessing

Hive is used to perform data preprocessing by creating a structured schema with appropriate data types for each column, handling the header row by setting the "skip.header.line.count" property, normalizing data formats (converting text fields to uppercase), encoding categorical variables into numerical representations (converting "yes"/"no" to 1/0), creating ordinal encodings for multi-level factors (like cholesterol level) and standardizing the dataset through the cleaned view.

4.1.3 Data Transformation

Hive is used to transform the data through the CREATE OR REPLACE VIEW statement which applies a series of CASE statements and other SQL functions to convert text-based categorical values into numerical formats more suitable for analysis and modeling.

4.1.4 Data Storage and Export

Hive is used to export the processed data in Parquet format (a columnar storage format optimized for analytics) to Google Cloud Storage for analysis by Spark.

4.2 Apache Spark

4.2.1 Data Loading

PySpark is used to initialise a Spark session and loading the preprocessed heart attack dataset from the Google Cloud Storage bucket in parquet format which was previously processed by Hive. Then, generic column names are mapped to meaningful ones like "age", "gender" and "heart attack" for better readability.

4.2.2 Data Preprocessing

During preprocessing, categorical and numerical columns are automatically identified, then a series of data transformation stages is constructed. For categorical variables like gender and region, StringIndexer is applied to convert strings to numerical indices and OneHotEncoder is used to create binary dummy variables. Numerical features such as cholesterol levels and blood pressure readings are combined into a single feature vector using VectorAssembler, then standardized with StandardScaler to normalize their scales. These preprocessing steps are collected into a pipeline to ensure consistent transformations during both training and prediction.

4.2.3 Data Modeling

Random Forest Classification is an ensemble learning method that combines the predictions of multiple decision trees for classification. It builds several individual decision trees during training. Each decision tree makes its own prediction and the final result is obtained through a voting system. Random subsets of the original dataset are created through resampling. Each tree is trained on a random subset of features to make the model more diverse and reduce the chance of overfitting. Decision trees are built independently from each subset by splitting nodes based on the best feature that minimizes the impurity. The final prediction is made by majority voting among all trees.

The modeling phase configures a Random Forest Classifier with 200 trees and a maximum depth of 5, using the preprocessed features to predict heart attack risk. The pipeline approach chains all preprocessing steps with the model training. After splitting the data into 80% training and 20% test sets with a fixed random seed for reproducibility, the model is trained on the training data.

4.2.4 Model Evaluation

For evaluation, the model generates predictions on the test set and calculates the area under the ROC curve (AUC-ROC) using BinaryClassificationEvaluator to assess its discrimination ability between high-risk and low-risk patients. The script then enhances the predictions by adding "risk category" column that classifies patients as either "High Risk" or "Low Risk" based on the model's output. These predictions, key demographic and outcome columns are saved back to GCS bucket for further analysis.

5. Results

5.1 Predictive Analysis

For the dataset of 158,355 individuals, 80% was allocated to the training set and 20% to the test set. Random Forest Classification model predicted with 792 and 18,580 True Positive and True Negative values respectively.

5.1.1 Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy is the proportion of correctly classified instances out of the total instances. Both models have identical accuracy which means that they both classified the same percentage of samples correctly.

5.1.2 Precision

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

Precision is the proportion of correctly predicted positive cases out of all predicted positive cases.

5.1.3 Recall

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

Recall is the proportion of correctly predicted positive cases out of all actual positive cases.

5.1.4 F1-score

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-score is the harmonic mean of precision and recall, balancing both metrics.

5.1.5 Area Under Curve ROC

The Area Under the ROC Curve is a performance metric for binary classification models (e.g., predicting heart attack risk as "Yes/No"). It measures the model's ability to distinguish between classes (e.g., "High Risk" vs. "Low Risk") across all possible classification thresholds.

5.2 Evaluation

The table below shows the metrics used for evaluation of the Random Forest classifier model trained by Apache Spark.

Metric	**Value**
Accuracy	0.61
Precision	0.63
Recall	0.062
F1 Score	0.113

Table 1: Model Performance Metrics

5.3 Prescriptive Analysis

After training the model to predict the risk of having a heart attack, prescriptive analysis was performed to provide actionable recommendations based on the predicted outcomes. A predicted value of 0 signifies that a patient is not likely to have a heart attack while a predicted value of 1 indicates a high likelihood of heart attack. For individuals identified as high-risk, specific health and lifestyle changes are suggested to mitigate potential dangers. These recommendations include: exercising for at least 30 minutes daily, quitting smoking, reducing or eliminating alcohol consumption, increasing sleep duration, and managing stress levels more effectively. The goal of this analysis is not only to predict risk but to offer meaningful guidance that can help improve cardiovascular health outcomes. For individuals where no significant risk was detected, the system simply recommends maintaining current habits, labeling them as "Healthy." This is based on important features identified through feature importance analysis.

6. Visualization

Visualization is done through Apache Zeppelin notebook, using `%pyspark` code. A new cluster with Apache Zeppelin integration is created and a notebook is made.

It performs a comprehensive analysis of heart attack risk predictions with a focus on both performance metrics and demographic breakdowns. It begins by reading the file containing prediction results that was generated by the PySpark script from Google Cloud Storage, renaming and casting columns to appropriate data types. Several visualizations are created to help interpret model performance. The notebook first visualizes the counts of actual vs. predicted values to show classification outcomes. Then, it explores the distribution of actual and predicted labels to understand class balance. It further breaks down model accuracy by age group, gender, location and risk category, presenting these comparisons visually. To investigate the dataset's composition, it also shows distributions for age, gender, location, and risk labels. Error analysis is conducted by calculating false positives and false negatives across age groups, gender, and locations, with visualizations showing error rates. Overall model performance is summarized through derived metrics such as true positive rate, true negative rate, false positive rate, false negative rate, accuracy, precision, recall, and F1 score. Additionally, a demographic combination analysis shows model accuracy by cross-referencing gender and location. Visualizations are rendered using z.show() to produce interactive tables and charts in Zeppelin to support evaluation of the model's performance and its fairness across different demographic segments.

Some visualizations:

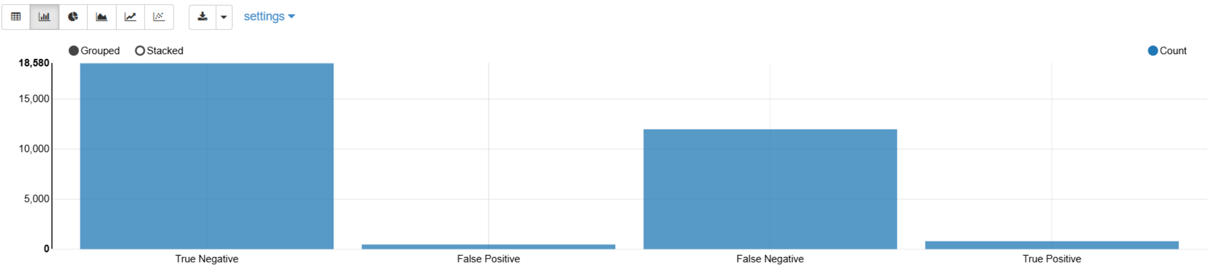


Figure 1: TP, FP, FN, TP

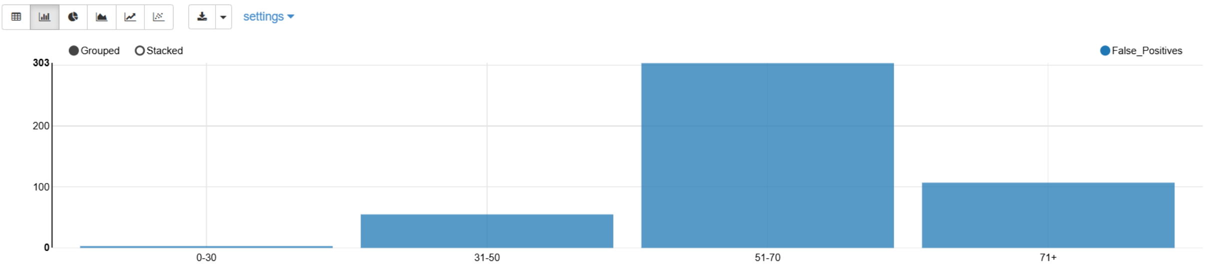
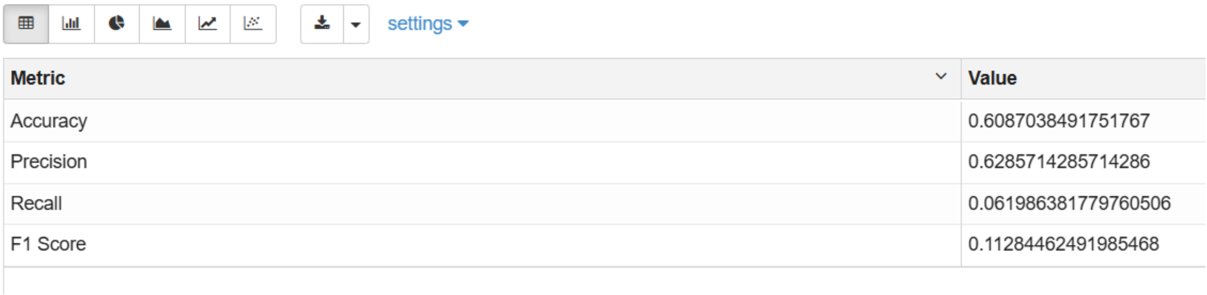


Figure 2: By Age



Metric	Value
Accuracy	0.6087038491751767
Precision	0.6285714285714286
Recall	0.061986381779760506
F1 Score	0.11284462491985468

Figure 3: Evaluation Metrics

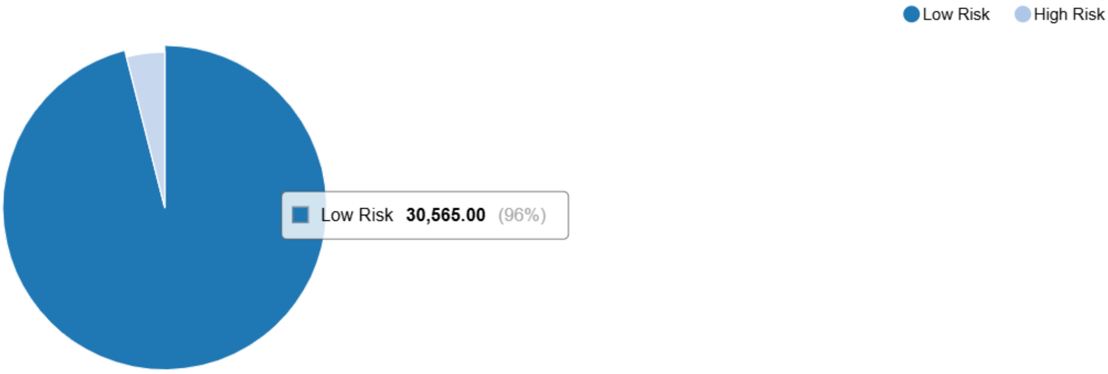


Figure 4: Low Risk, High Risk

7. Conclusion

This study has successfully made use of the Hadoop ecosystem through Google's Dataproc to develop a model that predicts the chance of having heart attack using Random Forest classification. Apache Hive and Apache Spark were used in the data analytics lifecycle to preprocess and analyse and in predictive and prescriptive analytics. Predictions were made on the test data - high or low risk of heart attack. The analysis then suggested preventive measures for high-risk patients such as encouraging better sleep and more time spent exercising. This approach provides valuable predictions about the incidence of heart attack rates in Indonesia. Future work could involve refining the models using additional data, experimenting with other machine learning algorithms and incorporating real-time patient data for dynamic decision-making.