Devon Midkiff

Dr. Weihua Zhou

Big Data Analytics SAT5165

10/15/2024

Cardiovascular Disease Prediction Using Spark

This project aims to predict the likelihood of cardiovascular disease by utilizing Spark for data preprocessing, logistic regression, and correlation analysis in a distributed computing environment. The project employs two virtual machines (VMs) for distributed data processing and analysis. The dataset used for this project contains more than one hundred thousand records sourced from the cardiovascular disease dataset on Kaggle. It includes variables such as age, health habits, and diagnosed medical conditions, which are used to assess the risk of cardiovascular disease.

The implementation is done using the following methods:

* Preprocessing: Data is cleaned and prepared for analysis by scaling numerical features and converting categorical features into numerical representations.
* Statistical Analysis: Logistic regression is used to predict the likelihood of cardiovascular disease, and correlation analysis identifies key relationships between health features and disease risk.
* Distributed Computing: The project uses Spark with two VMs, simulating a distributed environment. This ensures that the computation is spread across multiple nodes, enabling scalability.

Tools and Technologies Used

* Spark: For data preprocessing, logistic regression, and correlation analysis. Spark handles the distributed execution on the VMs.
* Python: The programming language used for writing the code and working with data processing libraries.
* PySpark: For working with Spark from Python, handling distributed data and machine learning tasks.
* Logistic Regression: Used to train the model and predict cardiovascular disease.
* Correlation Analysis: Performed to find relationships between health-related features and the risk of cardiovascular disease.
* Github: Used to manage changes and version control throughout the project located: [DevonMidkiff/Cardiovascular-Disease-Prediction-Using-Spark](https://github.com/DevonMidkiff/Cardiovascular-Disease-Prediction-Using-Spark)

**Project Workflow**

Step 1. Data Preprocessing

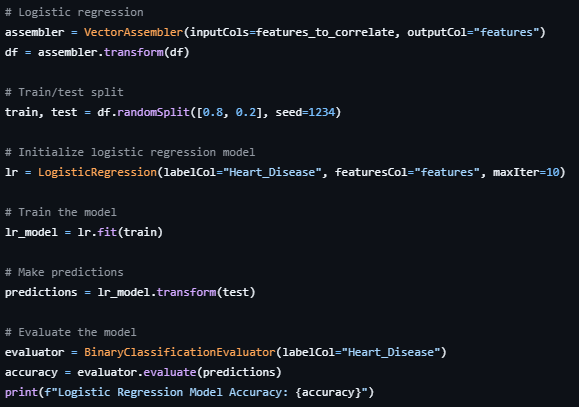
The data preprocessing stage ensures the dataset is ready for logistic regression modeling. The following steps were taken:

* Data Loading: The dataset was loaded into a Spark DataFrame using spark.read.csv().
* Conversion of Categorical Variables: The 'Checkup', 'General\_Health', and 'Age\_Category' columns were converted into ordinal numerical values. Similarly, Boolean features like 'Exercise', 'Heart\_Disease', 'Smoking\_History', and 'Sex' were converted to binary values.
* Scaling of Numerical Variables: MinMaxScaler was used to scale features like alcohol consumption, fruit and vegetable consumption, height, weight, and BMI, bringing all values into a uniform range.

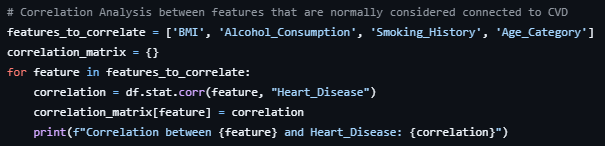


2. Statistical Analysis

The logistic regression model was trained using PySpark's MLlib. The model predicts the likelihood of cardiovascular disease based on the preprocessed features. The model was evaluated using accuracy.



Correlation analysis was conducted to identify relationships between the features and cardiovascular disease. Correlations between health features like BMI, smoking, and alcohol consumption, and the target variable (heart disease) were computed.



3. Performance Comparison

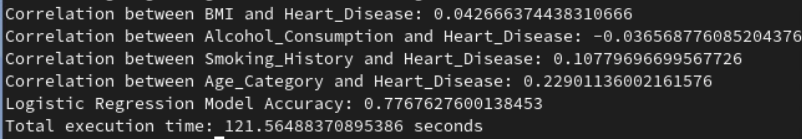
The project was implemented across two VMs, using Spark's distributed computing framework to handle the dataset in parallel. To compare the performance, we ran the process using:

* One VM: The code was executed on a single VM to observe performance in a non-distributed environment.
* Two VMs: The code was executed in a distributed environment where the workload was shared across the nodes.

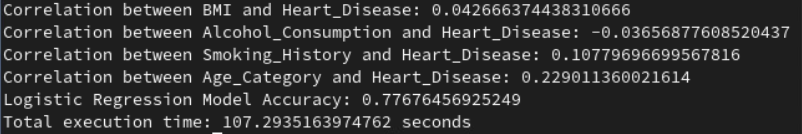
Using two VMs reduced the runtime, allowing for faster analysis than just one VM.

Performance Results:

One VM:



Two VMs:



The two-VM setup reduced the total computation time by approximately twenty-two percent compared to the single VM, demonstrating the scalability of distributed processing in Spark.

**Challenges Encountered**

* Worker Node Communication: The worker nodes could not communicate properly with the master, causing disconnection issues. This was resolved by adjusting /etc/hosts to the proper IP addresses.
* Resource Allocation: As the VMs were limited to one core each, I optimized resource usage by repartitioning the data and ensuring efficient utilization of available cores.

**Conclusion**

This project successfully demonstrates the use of Spark for distributed data processing and machine learning. By implementing logistic regression and correlation analysis in a Spark environment, I efficiently processed a large dataset and predicted cardiovascular disease. The performance comparison between one and two VMs shows the advantage of distributed computing, reducing the time needed for analysis. Future improvements could involve expanding the number of worker nodes or implementing additional machine-learning models for higher accuracy.