Devon Midkiff

Dr. Weihua Zhou

Big Data Analytics SAT5165

11/05/2024

Cardiovascular Disease Prediction Using Spark Naive Bayesian

This project aims to predict the likelihood of cardiovascular disease by utilizing Spark for data preprocessing, Naïve Bayesian, 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: Naive Bayesian 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, Naive Bayesian, 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.
* Naive Bayesian: 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-with-Spark-NaiveBayes](https://github.com/DevonMidkiff/Cardiovascular-Disease-Prediction-with-Spark-NaiveBayes)

**Project Workflow**

Step 1. Data Preprocessing

The data preprocessing stage ensures the dataset is ready for Naive Bayesian 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.



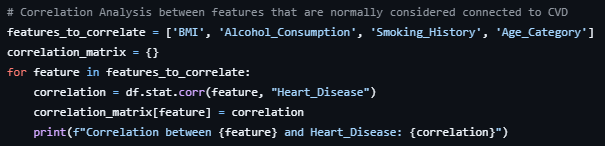
2. Statistical Analysis

The Naive Bayesian 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, AUC, recall, precision, specificity.

A computer screen shot of a program

Description automatically generated

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:

A computer screen with white text

Description automatically generated

Two VMs:

A computer screen with white text

Description automatically generated

The Naive Bayes model’s performance on predicting cardiovascular disease shows a high overall accuracy of 88.9% and a strong specificity of 95.5%, indicating effective identification of negative cases (individuals without disease). However, the model struggles significantly in distinguishing positive cases, with a low recall of 14.4% and a precision of 22.2%. This is further reflected in the low AUC of 40.8%, suggesting that the model’s predictive power in differentiating between positive and negative cases is limited. These results imply that, while the model reliably identifies non-disease cases, it often misses true positive cases, which could be critical in a healthcare setting. Comparing execution times, the model performed faster on two VMs than on a single VM, showcasing the benefits of distributed processing in enhancing computation efficiency.

**Challenges Encountered**

* Spark Cluster Issues: The cluster had some issues that persisted in both test cases where it was giving a version mismatch. I could not resolve the problem, but I found a workaround by modifying the spark config.
* 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 effectively demonstrates the capabilities of Spark for distributed data processing and machine learning by implementing a Naive Bayes classifier and performing correlation analysis in a multi-node Spark environment. Using two virtual machines, I processed a large cardiovascular disease dataset, highlighting the performance benefits of distributed computing, which reduced analysis time from 148 seconds on a single VM to 124 seconds on a dual-VM setup. The high specificity and overall accuracy of the model indicate its strength in identifying non-disease cases, although its lower recall and AUC suggest limitations in distinguishing true positives. Future enhancements could include increasing the number of worker nodes to further boost computational efficiency and exploring more advanced machine learning algorithms, such as Random Forests, boosting methods, or neural networks, to improve recall and AUC. These refinements would support a more balanced model, with enhanced ability to identify individuals at risk of cardiovascular disease, addressing critical needs in healthcare analytics.