

Unveiling Predictive Variables for Liver Disease Using Machine Learning

By: Daniel Brusche

Dataset Chosen

A dataset with a sample size of 1700 records was chosen from Kaggle to investigate the prediction of liver disease, denoted by the variable Diagnosis (Binary indicator of liver disease presence: 0 or 1). The analysis focuses on variables such as Age (Range: 20 to 80 years), Gender (Male: 0, Female: 1), BMI (Body Mass Index) (Range: 15 to 40), Alcohol Consumption (Range: 0 to 20 units per week), Smoking (No: 0, Yes: 1), Genetic Risk (Low: 0, Medium: 1, High: 2), Physical Activity (Range: 0 to 10 hours per week), Diabetes (No: 0, Yes: 1), Hypertension (No: 0, Yes: 1), and Liver Function Test (Range: 20 to 100). With a strong interest in exploring public health, this dataset can provide valuable insights into the factors influencing health and contribute to a growing understanding of the intersectionality of health.

Research Question

In this paper, we will explore how variables such as age, gender, BMI, alcohol consumption, smoking, genetic risk, physical activity, diabetes, hypertension, and liver function test can predict the diagnosis of liver disease.

Data Cleaning and Preparation

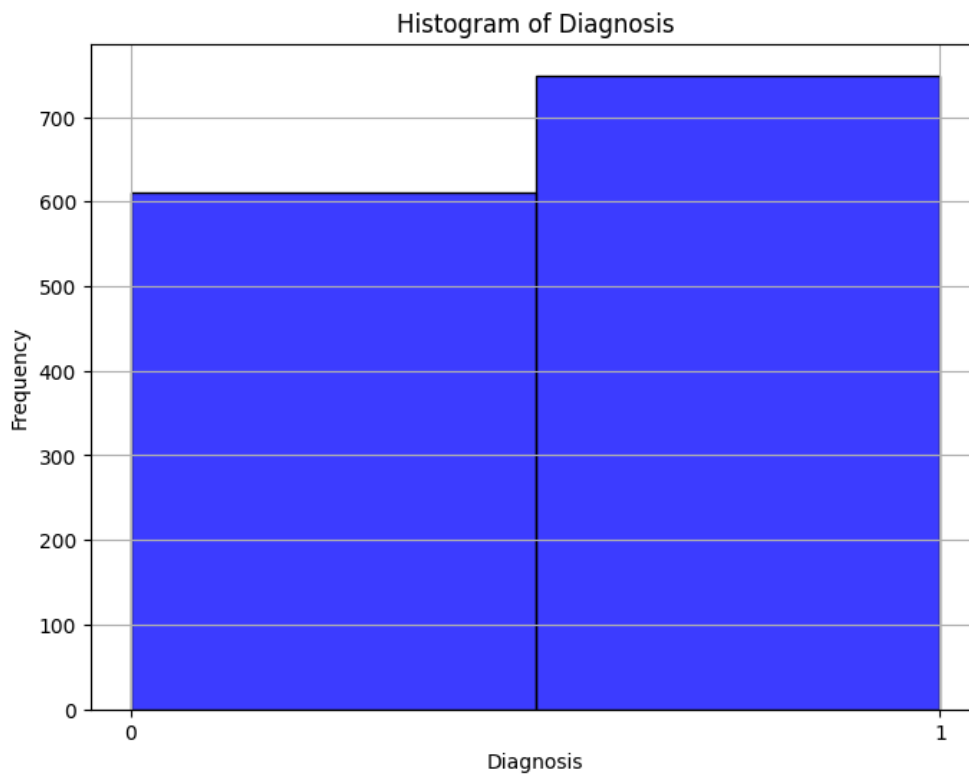
The dataset initially contained no missing data and consisted mostly of binary variables. I performed a train-test split where X represents the independent variables and Y the target variable (Diagnosis). To ensure reproducibility, I set `random_state=0` and did not stratify the split (`stratify=None`) based on the target variable. The training set (`X_train, y_train`) was used to train the machine learning model, while the test set (`X_test, y_test`) evaluated its performance.

Initially, I applied a linear regression model to predict the target variable, achieving a score of 0.34. This lower test score suggested potential overfitting, where the model fit too closely to the training data, hindering its ability to predict new data accurately. To address this, I introduced Ridge Regression, a regularization technique aimed at simplifying the model to prevent overfitting. However, even with Ridge Regression (using `alpha=1`), the model showed improvement in the training set score (47%) but a drop in the test set score (34%), indicating that some degree of overfitting may still persist.

Given that my variables are binary, linear regression is not ideally suited as it typically applies to continuous variables. Therefore, the choice of Ridge Regression was more appropriate for regularization purposes. Further refinements in model selection and tuning of regularization parameters may help improve predictive performance and generalization capability.

Data Visualization

Table.1



In Table 1, the breakdown of the frequency distribution of our target variable shows that diagnoses coded as 1 account for a frequency of 750, whereas those coded as 0, indicating no diagnosis, total 610.

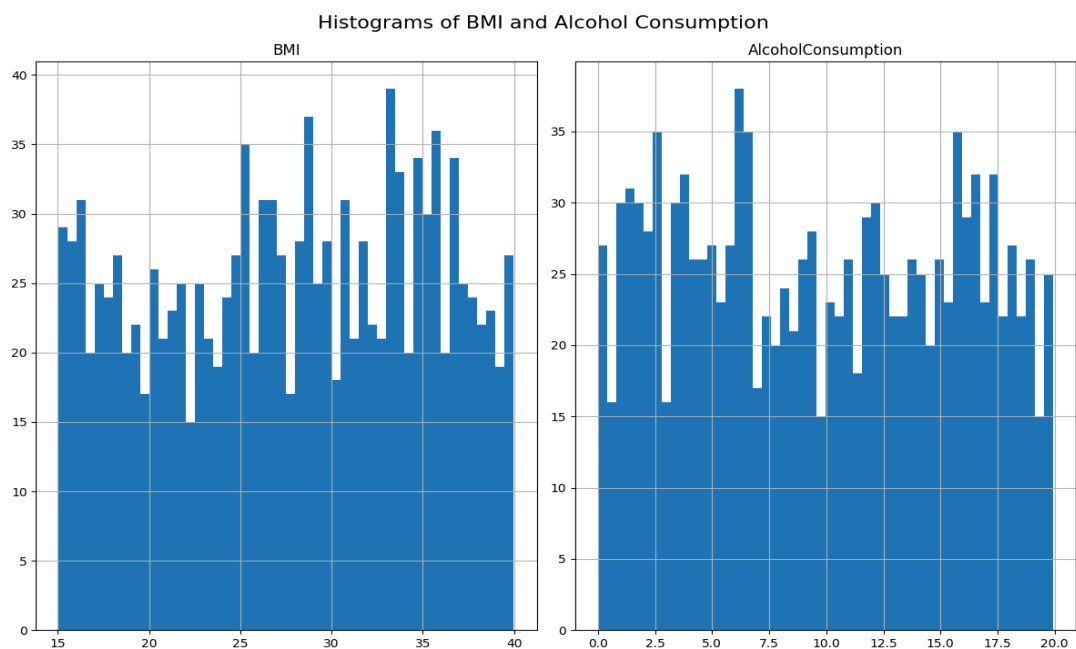


Table 2.

Table 2 shows histograms of BMI and alcohol consumption. Not much can be explained from these graphs as they exhibit non-skewed relationships.

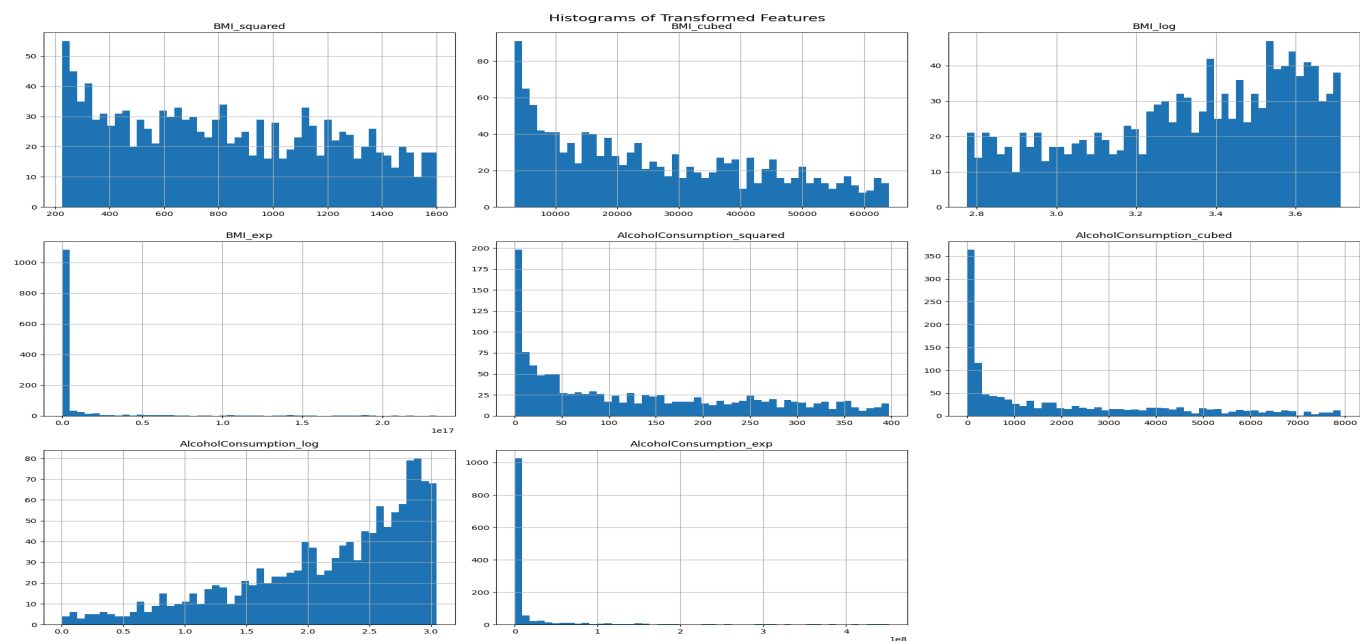


Table 3.

In Table 3, the variables BMI and Alcohol Consumption were transformed through squaring, cubing, logarithm, and exponential functions. This was done to enhance their analytical capabilities and suitability for modeling. These transformations aim to address issues such as skewness and to create distributions that are more normal or to expand the data's range to avoid clustering around specific points. Compared to Table 2, we can observe right-skewed and left-skewed distributions in Table 3, indicating that the transformations have made the variables more symmetric.

Supervised Machine Learning: Gradient Boosting Decision Trees

I performed a Gradient Boosting Decision Trees analysis, a technique used to create new trees that correct the errors of the previous ones, thereby improving the model. I used `n_estimators=100`, indicating the number of trees used; more trees can potentially lead to a better model. The parameter `random_state=0` was set to ensure consistent results with each new run of the model. From these results, the training set accuracy was 96.3%, and the test set accuracy was 87.6%, indicating good generalization to new data.

To further support the model, cross-validation was employed to evaluate its performance on the entire dataset. Using cross-validation, the evaluation metrics were as follows:

- **Precision:** 92%, indicating the model correctly predicts liver disease 92% of the time.
- **Recall:** 92%, indicating the model can correctly identify 92% of all instances of liver disease.
- **F1 Score:** 92%, showing a good balance between precision and recall.
- **Mean Accuracy:** 91%, indicating the average proportion of correctly classified instances, whether positive or negative, in the dataset.

These high metrics suggest that the Gradient Boosting Decision Trees analysis is effective in predicting liver disease.

Supervised Machine Learning: Gaussian Naive Bayes Model

The second algorithm used is Gaussian Naive Bayes, which assumes that the variables are independent of each other. Cross-validation was done to evaluate how the model will perform on different subsets of the data. Precision, recall, and F1 score were also calculated to assess how well the model can predict liver disease.

From the results, we see that the training set score is 81%, indicating that the model can classify 81% of the training data correctly. The test score is 80%, showing that the model generalizes well to new data. The scores being close together indicate that no overfitting is occurring.

From the cross-validation metrics, we see that:

- **Precision** is 83%, measuring the proportion of correctly predicted positive instances out of all instances predicted as positive. A precision of 83% indicates that when the model predicts an instance as positive, it is correct 83% of the time.
- **Recall** is 81%, showing that the model can identify 81% of positive cases and measures the model's ability to find all positive cases.
- **F1 Score** is 82%, indicating a good balance between precision and recall.
- **Mean Accuracy** is 80%, demonstrating that the model performs consistently across different subsets of data.

In summary, the Gaussian Naive Bayes model performs well on the dataset. The metrics show its ability to generalize to both training and test data effectively.

Differences of Supervised Machine Learning Models

Although both models performed well in predicting and generalizing the data, the Gradient Boosting Decision Trees model outperformed Gaussian Naive Bayes. One possible

reason for this is that Gaussian Naive Bayes assumes independence between features, which may not hold true in datasets where variables exhibit dependent relationships. As a result, the model might struggle to capture these dependencies effectively. On the other hand, Gradient Boosting Decision Trees can capture complex interactions between variables, making it more suitable for datasets with interdependent features. This capability allows the model to learn and adapt to the nuances of the data more effectively.

Unsupervised Machine Learning with PCA

In unsupervised machine learning, our goal is to uncover patterns and make predictions on unlabeled data. Principal Component Analysis (PCA) is a key technique for enhancing models by improving data visualization and capturing the most significant variance in datasets. For our best performing model, Gradient Boosting Decision Trees, using PCA resulted in a testing score of 80.7%. This indicates the model's ability to predict within the dataset. In contrast, without PCA transformation, the model achieved a score of 96.3%. The decrease in score can be attributed to reduced dimensionality, meaning PCA may have removed valuable information from the dataset.

Clustering: Agglomerative and PCA

Agglomerative clustering is a method used to group data points based on their similarities. PCA can enhance clustering and visualization by reducing the dimensionality of the data, thereby focusing more on the important variance. We use clustering to enhance our understanding of data patterns and to effectively reduce data complexity.

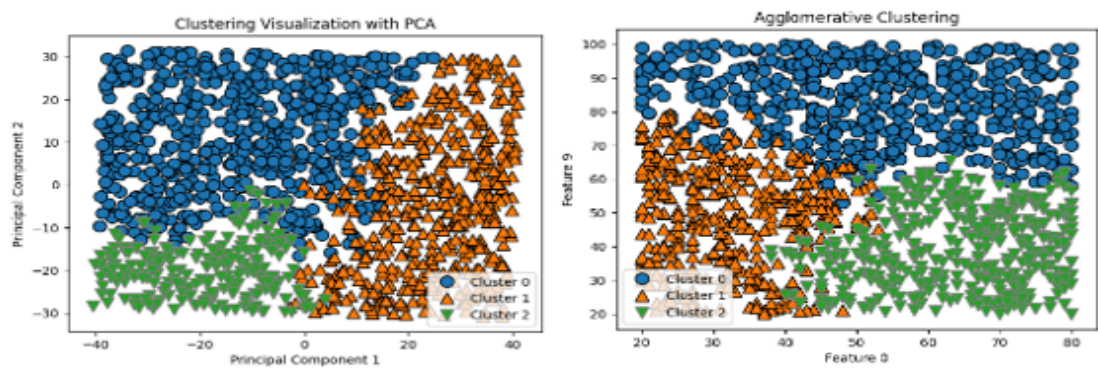


Table. 4

From Table 4, we observe that before PCA processing, the clusters were mixed together without clear defining groups. After PCA processing, however, the clusters are more clearly grouped based on their similarities.

Clustering: Kmean and PCA

K-Means is an algorithm in machine learning for clustering data into partitioned groups. It can also be used as a tool to determine the optimal number of clusters for a dataset based on Inertia representing the space within the clusters of data points. Lower inertia values indicate better clustering performance, as they signify that the clusters are more compact and well-defined.

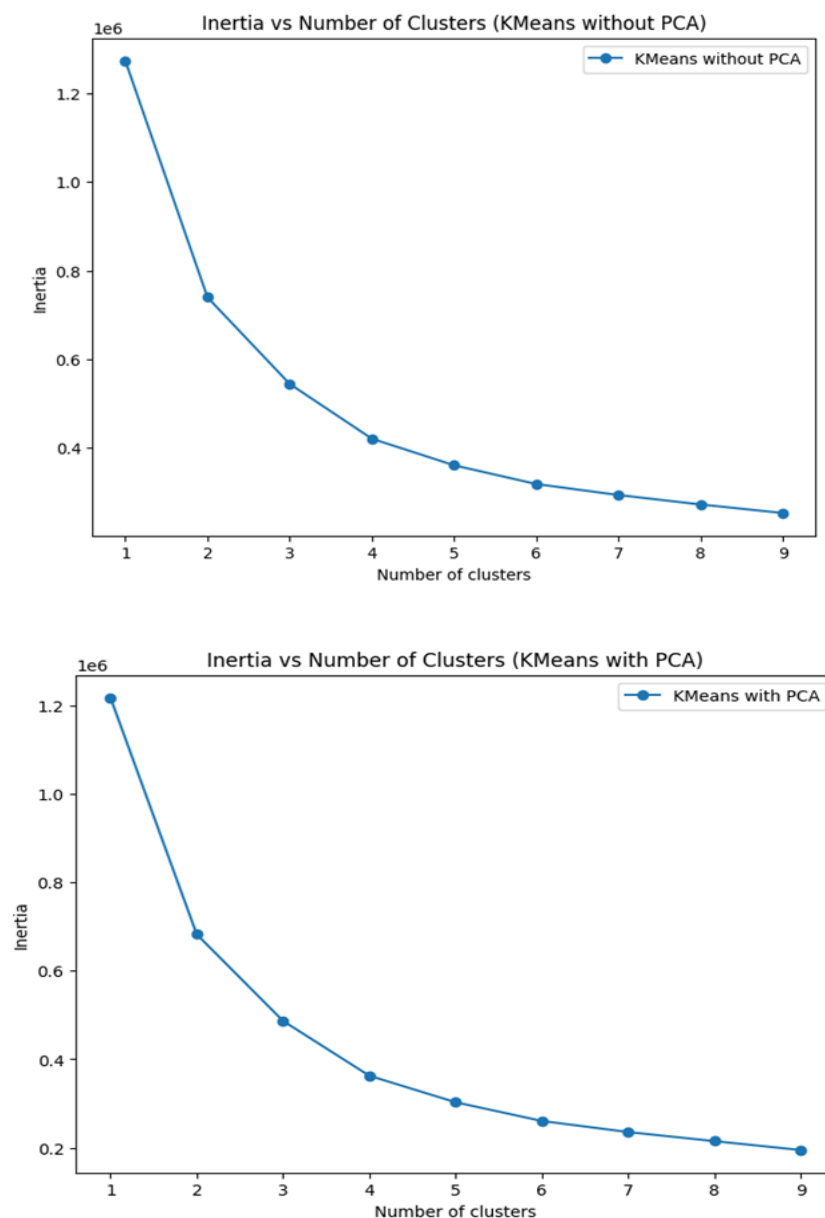


Table. 5

Table 5 shows that before PCA processing, when the cluster count is 3, the inertia is closer to 0.6. After PCA processing, with the same cluster count of 3, the inertia is closer to 0.4, indicating improved cluster formation as the inertia value decreases.

Clustering: DBSCAN and PCA

DBSCAN is a clustering technique that forms clusters based on the density of data points, identifying points in low-density areas as outliers or noise. In Table 6, without PCA preprocessing, no distinct clusters were detected. However, after applying PCA, three clusters emerged. This improvement can be attributed to PCA's ability to reduce the dimensionality of the data, which often enhances the separation of clusters by focusing on the most significant variance in the dataset.

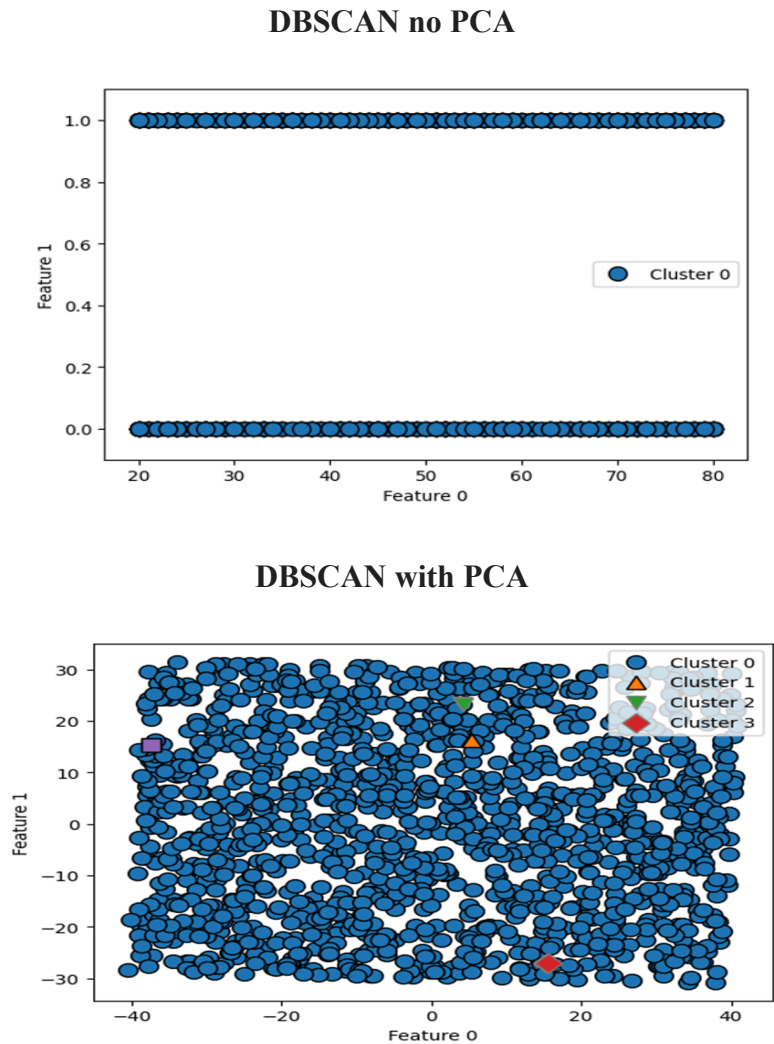



Table. 6

Summary

In summary, the three projects taught me a better understanding of using Python for statistical programming. In Project 1, I learned how to import and clean data, segmenting it into different groups suitable for machine learning. I also gained insights into data visualization and how data transformation can enhance visualization. Project 2 focused on supervised machine learning, where we used models to predict liver disease diagnoses. In Project 3, we explored unsupervised machine learning techniques, employing PCA to enhance clustering in dataset and visualization. If I were to do something differently, I would choose a more complex dataset. The dataset I used was straightforward and well-cleaned, which limited opportunities to gain experience in data cleaning and utilizing complex models.

Bibliography

Kharoua, R. E. (2024, June 10).  *predict liver disease: 1700 records dataset*. Kaggle. <https://www.kaggle.com/datasets/rabieelkharoua/predict-liver-disease-1700-records-dataset/code>

Akramz. (n.d.).

AKRAMZ/hands-on-machine-learning-with-scikit-learn-keras-and-tensorflow: Notes & Exercise Solutions of Part I from the book: "hands-on ML with scikit-learn, Keras & Tensorflow: Concepts, tools, and techniques to build Intelligent Systems" by Aurelien Geron. GitHub.

<https://github.com/Akramz/Hands-on-Machine-Learning-with-Scikit-Learn-Keras-and-TensorFlow>