Metabolic Syndrome Classification

Ajay Patel, Bryce Berkhof, Gus Vietze, Raghav Choudhary

Abstract— **This project uses the Kaggle dataset 'Metabolic Syndrome' to build prediction models aimed at classifying metabolic syndrome's presence based off various factors. In this report, we detail our process and methodology used, which include but aren't limited to XGBoost, Logistic Regression, Random Forests, and Neural Networks. These models succeeded at classifying metabolic syndrome at high accuracy, improving on results we found from previous literature. Our goal was to create very accurate classification methods that can be used by doctors and hospitals everywhere to establish a standard process for classifying the syndrome.**

I. Introduction

In this paper, we present the results of our final project which focused on the classification of Metabolic Syndrome using various modeling techniques. Metabolic Syndrome is a health condition characterized by a combination of risk factors, including high blood pressure, high blood sugar, excess body fat, and abnormal cholesterol levels. It greatly increases the risk of developing chronic diseases such as diabetes, heart disease, and stroke. Early detection and intervention are crucial in preventing these serious health issues.

Our project workflow involved several key steps to ensure accurate classification. The first step was data pre-processing, where we performed a thorough cleaning of our dataset, removing any missing values. We also applied one-hot encoding to transform categorical attributes into numerical variables for better model compatibility.

To address the issue of high dimensionality in our dataset, we employed Principal Component Analysis (PCA). PCA allowed us to reduce the number of features while retaining the most informative structure of the data. In addition to dimensionality reduction, we explored the use of K-Means clustering to identify potential clusters within our dataset.

Moving on to the modeling phase, we employed a range of techniques to classify Metabolic Syndrome. One of the techniques explored was logistic regression. Logistic regression is a statistical method used for binary classification problems. It models the probability of a certain event occurring based on the values of the independent variables. Furthermore, we employed XGBoost, a powerful machine-learning algorithm known for its effectiveness in handling structured data. XGBoost combines the strengths of gradient boosting with regularization techniques to improve accuracy and handle complex relationships within the data. Another modeling technique we utilized was random forest, an ensemble learning method that combines multiple decision trees to make predictions. Random forest has been proven to be robust against overfitting and able to handle high-dimensional data effectively. Lastly, we employed a neural network model to classify Metabolic Syndrome. Neural networks are a family of machine learning algorithms inspired by the structure and functioning of the human brain. They consist of interconnected nodes organized into layers and can learn complex patterns from the data.

To evaluate the performance of our models, we utilized cross-validation accuracy, overall accuracy, and F-1 score. Cross-validation allowed us to detect potential overfitting and assess the generalization capabilities of our models. Overall accuracy provided us with a holistic measure of how well our models were able to classify Metabolic Syndrome. The F-1 score, which takes into account both precision and recall, allowed us to evaluate the balance between correctly identifying true positives and avoiding false positives and false negatives [7].

Our project aimed to build a highly accurate model for the classification of Metabolic Syndrome.. Our neural network model emerged as the top performer, achieving an impressive accuracy of 91.4% and an F-1 score of 0.91, but XGBoost was found to do a similar job while being more interpretive. Furthermore, we identified areas for further improvement, such as expanding the dataset to include diverse population subgroups and dedicating more time to feature selection. The successful implementation of our classification model has the potential to greatly improve early detection and possible disease prevention. By automating the classification process, we can save manual labor and improve response times for patients. This project lays the foundation for further research and provides a valuable tool for healthcare professionals to mitigate the risks associated with Metabolic Syndrome.

II. Literature Review

Plenty of research has been done detailing the effects of metabolic syndrome and what it can lead to. It's noted to "greatly increase the risk of developing diabetes, heart disease, stroke, or all three" [1]. While it isn't a disease itself, the fact that it can lead to those potential diseases is frightening enough. It's also quite prevalent, as about one in three adults have metabolic syndrome [2]. Its prevalence only stresses the need for early identification.

Currently, the process for identifying metabolic syndrome includes identifying three of the following criteria: excess abdominal weight, high triglyceride levels, low HDL levels, elevated blood sugar, and high blood pressure[8]. While each of these has clearly defined thresholds that allow for their identification, it is a complicated process for doctors to go through for every patient they encounter. Part of our inspiration for this project was finding a way to simplify this process to improve response times and ease the lives of doctors.

In previous classification studies, BMI is often used as the most common measure of obesity, where it along with triglyceride count has served as a good predictor of metabolic syndrome [3]. We made sure to find a dataset that incorporated these features as if past research found use with them, we likely would as well. Luckily, our dataset did include them, which we will discuss more in the Data section. One previous study, done by Yang et al., was able to produce an accuracy value of around 85%, although their F-1 score was only 0.58 [4]. Their group found that XGBoost was the best performer, which we made sure to include in the models we would try out. This study also helped us set our baseline for what we should expect in terms of model performance.

Another study from Zou et al. found similar results in that XGBoost worked best for predicting metabolic syndrome, using obesity, triglycerides, waist circumference, and other features in their model [5]. At the same time, we knew that different methods were worth investigating that could improve performance and build upon previous work.

III. Data

Before we got to constructing our models, it was important for us to understand our dataset. Our dataset came from Kaggle, published by Albert Antony, who compiled the data from the CDC [6]. It includes 2,401 total observations with 15 different columns. Our target variable was the MetabolicSyndrome column, which was labeled a 1 if the syndrome was present and a 0 if not. The other features consisted of the following:

- seqn: sequential identification number
- Age: individual's age
- Sex: whether the individual is a male or female
- Marital: individual's marital status
- Income: individual's income level
- Race: individual's racial background
- WaistCirc: individual's waist circumference
- BMI: individual's body mass index
- Albuminuria: measurement related to albumin in urine
- UrAlbCr: individual's urinary albumin-to-creatinine ratio
- UricAcid: individual's uric acid levels in the blood.
- BloodGlucose: individual's blood glucose levels
- HDL: individual's high-density lipoprotein cholesterol levels
- Triglycerides: individual's triglyceride levels

First, we analyzed the distribution of metabolic syndrome to see how frequent it was. In Figure 1, it's seen that 34.2% of observations had metabolic syndrome and 65.8% did not, meaning 1,579 observations did not have it and 822 did. This was a higher number of observations with metabolic syndrome than we expected, which we took as a positive as we wouldn't have to implement class balancing methods to adjust the dataset. Going back to our literature review, we read that about one-third of adults had metabolic syndrome, and we found a similar pattern in our dataset, which was re-affirming.

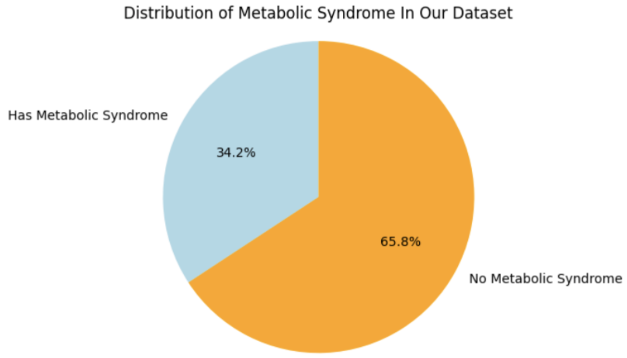
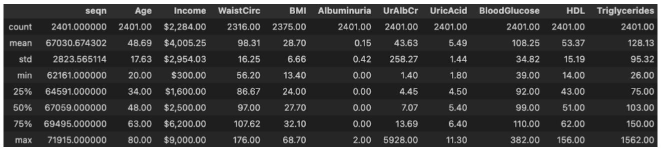


Fig. 1. Distribution of Metabolic Syndrome

Then, we summarized the numerical features in our dataset to understand the distribution and range of each, found in Figure 2.



	seqn	Age	Income	WaistCirc	BMI	Albuminuria	UrAlbCr	UricAcid	BloodGlucose	HDL	Triglycerides
count	2401.000000	2401.00	$2,284.00	2316.00	2375.00	2401.00	2401.00	2401.00	2401.00	2401.00	2401.00
mean	67030.674302	48.69	$4,005.25	98.31	28.70	0.15	43.63	5.49	108.25	53.37	128.13
std	2823.565114	17.63	$2,954.03	16.25	6.66	0.42	258.27	1.44	34.82	15.19	95.32
min	62161.000000	20.00	$300.00	56.20	13.40	0.00	1.40	1.80	39.00	14.00	26.00
25%	64591.000000	34.00	$1,600.00	86.67	24.00	0.00	4.45	4.50	92.00	43.00	75.00
50%	67059.000000	48.00	$2,500.00	97.00	27.70	0.00	7.07	5.40	99.00	51.00	103.00
75%	69495.000000	63.00	$6,200.00	107.62	32.10	0.00	13.69	6.40	110.00	62.00	150.00
max	71915.000000	80.00	$9,000.00	176.00	68.70	2.00	5928.00	11.30	382.00	156.00	1562.00

Fig. 2. Numerical Summary Statistics

Right away we noticed that the ranges of features all varied, as some did not reach triple digits whereas some got up to the high thousands. This led us to employ scaling in our preprocessing later on to prevent any feature from having an overly strong effect. We also noted that the dataset only contained adults, as the minimum age was 20. Additionally, from the count row, we see that WaistCirc and BMI had missing values, which would require imputing once we began preparing to model our data.

Having analyzed the numeric data, we next looked at the distribution of our categorical variables to gain a deeper understanding of the data, which is seen in Figure 3. These variables include Sex, Marital (marriage status), and Race. Sex had about a 50/50 split of females and males in our dataset. Most observations, about 1,200 of them, were married, with single being found at the next highest rate, and then divorced, widowed, and separated all falling in towards the end of the chart. The predominant race in our dataset was white, with minority races including black, Asian, Hispanic, and Mexican-American appearing as well. These distributions are important to keep in mind in furthering the scope

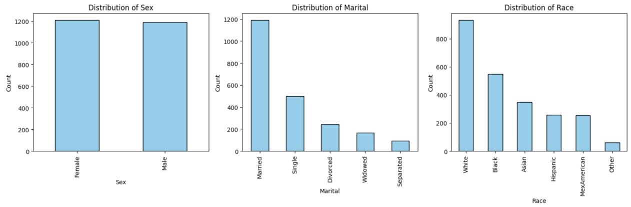


Fig. 3. Distribution of Categorical Variables

of our analysis, as about half our observations are white, so the results might not generalize to minority communities as well as models solely trained on those communities.

Additionally, we examined some of our categorical variables by whether the observation had metabolic syndrome or not to see if there were any initial strong patterns in our data.

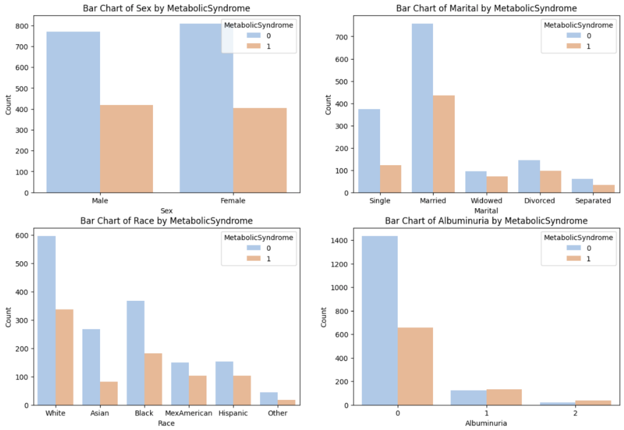


Fig. 4. Categorical Variables Examined By Metabolic Syndrome

Looking at Figure 4, the Sex bar chart in the top left shows similar distributions of metabolic syndrome for males and females, with a slightly higher prevalence in males but within reason. Next, the Marital bar chart shows a lower percentage of single persons having metabolic syndrome than married equivalents, while widowed and divorced persons appear to have metabolic syndrome at a higher rate. There were no strong disparities in metabolic syndrome rate for the Race variable, while the few observations that did have Albuminuria marked as a 1 or 2 showed a stronger increase in metabolic syndrome than those with a 0. This told us it would be a variable to keep an eye on when modeling, as it may perform well with strong correlations to our outcome.

IV. Hypotheses/Goals

Going into this project, we hypothesized that we would be able to classify metabolic syndrome's presence at a high accuracy, ideally above 80%, and at a high F-1 score, around or better than 0.65. We not only wanted to focus on achieving a high accuracy as a predictive project should but also on making realistic models that could be implemented with ease and not take too much run time, especially if doctors were to ever use them. With that said, we mainly focused on interpretive models including XGBoost, Logistic Regression, and Random Forests, and then one for performance in Neural Networks. We wanted to focus on presentable and low runtime results because we hypothesized that one can certainly make a high-performing model given the past work done on the topic, but the scalability of those methods might not be the best.

Furthermore, we wanted to see what features would play a key role in identifying metabolic syndrome. Previous research showed that obesity and triglyceride count had high importance, and we hypothesized that those two along with HDLs and waist circumference would be very important. If all went well, our goal was to be left with valuable insight from our models into predicting metabolic syndrome that can help prevent the onset of secondary diseases.

V. Methods

On top of the prior descriptive visuals, we also performed PCA and K-means clustering on our dataset to further understand how the features worked with each other and if there were any apparent patterns we should be aware of. PCA was chosen to see how effectively the dataset could be reduced, and if information would be transferred. We wanted to employ it in conjunction with a model, but this required us to see if it was even worthwhile first. We utilized sklearn's StandardScaler and PCA function to perform PCA, first scaling all the data to an equivalent range. Then, PCA was applied with 5 components. The resulting explained variance plot is in Figure 5.

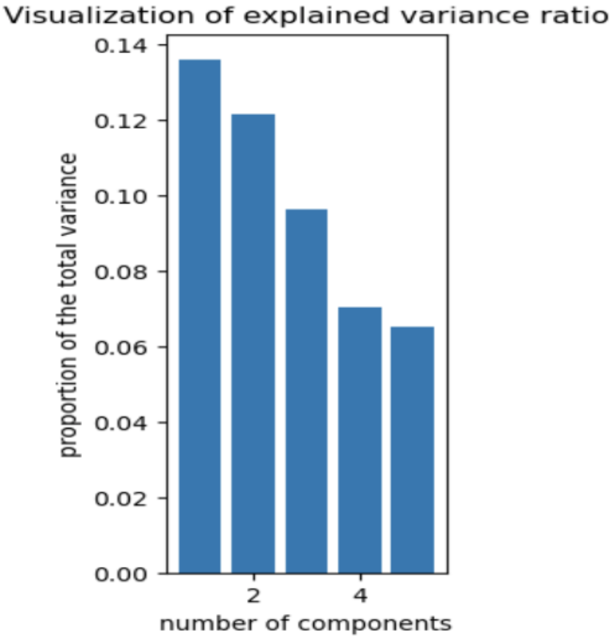


Fig. 5. PCA Explained Variance

The plot shows that each component adds a meaningful amount of explanation, with each having a proportion above 0.05. The first three components especially performed well, which would be useful when performing our PCA + Logistic Regression modeling technique, as it gives us a good number of components to use later on.

K-Means was employed to see if there were any pre-existing groups within the data that we should be wary of when modeling. Since it is unsupervised, it would identify patterns or trends in the dataset on its own. We first used the elbow method to find the optimal number of clusters, which is where one sees that adding more clusters doesn't meaningfully improve performance. The MetabolicSyndrome variable was of course not included in the algorithm as that would provide a natural group to cluster on and conflate with what we were looking for.

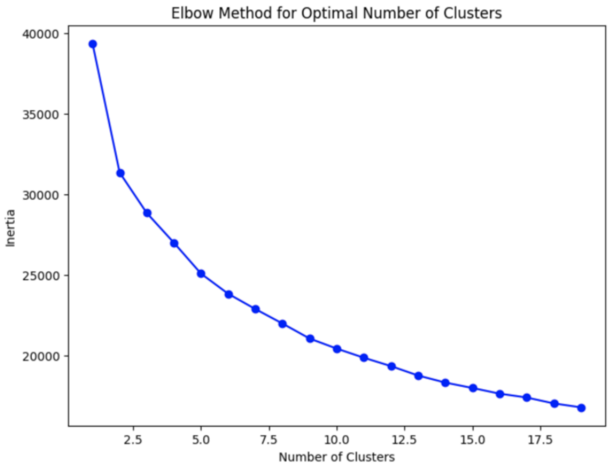


Fig. 6. K-Means Elbow Analysis

While different interpretations of the elbow analysis are possible, we chose 5 as the optimal number of clusters because it was a good trade-off of having too many clusters while also performing well. We see the curve start to flatten out after 5 clusters in Figure 6, further confirming our analysis. To prepare for clustering, we converted the categorical variables to numeric form and imputed any missing values with the means for numeric features and the most frequent observation for categorical features. The data was scaled and then we applied K-Means clustering to the dataset with the number of clusters set to 5 and found no real trends or groups within the data.

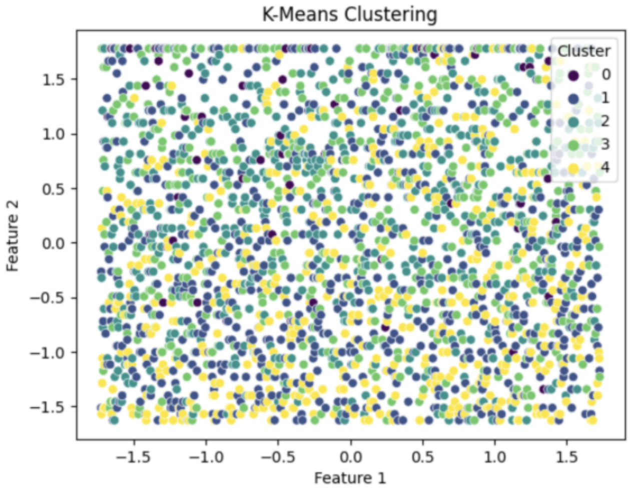


Fig. 7. K-Means Clustering (5 Clusters)

The resulting plot, Figure 7, shows what was described in that the clusters show no patterns which affirmed that we had a good dataset to continue with. We also tested the clustering algorithm with the number of clusters set to 2, to see if they would perhaps cluster along the MetabolicSyndrome variable and if it already had a trend present within the data.

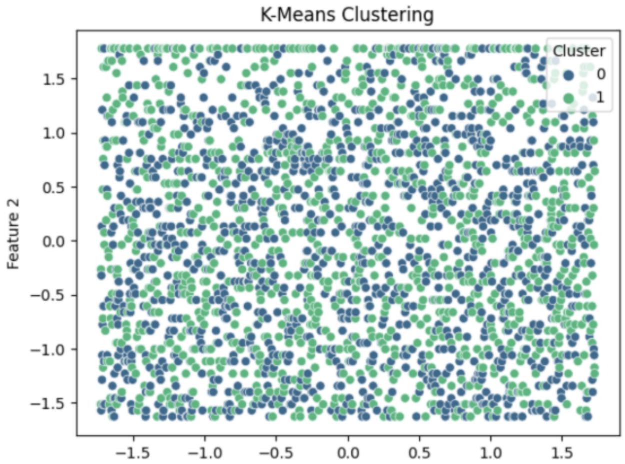


Fig. 8. K-Means Clustering (2 Clusters)

Once again, there is no clear association between the two groups, as shown in Figure 8. If anything, the clustering seems to be random, showing that there are no pre-existing groups within our dataset and that we chose a good dataset to continue with for modeling metabolic syndrome. All these steps, from the initial visuals to clustering, deepened our understanding of the dataset and how the features interacted with themselves and our outcome variable. We now felt ready to begin processing the dataset for modeling.

A. Data Preprocessing

As part of our project, we performed a series of data preprocessing steps to ensure the quality, relevance, and accuracy of our dataset. These steps included the following:

1) Handling Missing Values: Missing values are a common issue in datasets and can significantly impact the accuracy and performance of our models. To address this, we scanned our dataset for any missing values present and then used sklearn's SimpleImputer to update missing values.

For continuous variables with missing values, we opted for imputation using the mean or median of that column. For categorical variables with missing values, we employed mode imputation, where we replaced missing values with the most frequent value in that column. For our neural network model, we dropped missing values entirely instead of the techniques mentioned.

2) Handling Categorical Data: Our dataset contained categorical features, which are variables that have discrete and unordered values. To represent these features numerically, we performed one-hot encoding using SciKit Learn's LabelEncoder function in the preprocessing class. Label encoding involves creating a binary variable for each category of a categorical feature. For example, our target variable MetabolicSyndrome was encoded to have a binary value of 1 if metabolic syndrome was present and a binary value of 0 if it was not.

3) Feature Scaling: Several of our features had different scales, which can affect the performance of some machine learning algorithms. To address this, we employed feature scaling to bring all features to a similar scale. We used SciKit Learn's StandardScaler function from the preprocessing class to perform feature scaling. This class standardizes features by removing the mean and scaling to unit variance.

4) Training Test Split: Finally, we split our data into training and testing splits, with an 80% training and 20% testing split because our dataset included enough observations to split at the well-established ratio. We also employed KFold cross-validation during the modeling process to detect overfitting and add another baseline to compare our models at, choosing the default five folds to run the validation on. Sklearn's StatifiedKFold function was utilized to complete the cross-validation.

After completing these data preprocessing steps our dataset was prepared for modeling and further analysis on Metabolic Syndrome.

B. Models

We employed multiple models to classify Metabolic Syndrome. Each model was chosen based on its suitability for the task and its potential to achieve high accuracy. We utilized Logistic Regression, PCA into Logistic Regression, Random Forest, XGBoost, and a Neural Network.

1) Logistic Regression: Logistic Regression is a statistical method used for binary classification problems. It models the probability of a given instance belonging to a particular class. In the context of predicting Metabolic Syndrome, logistic regression provides a simple yet effective baseline model. We selected this model due to its interpretability and simplicity. Given our familiarity with it from past projects, we used sklearn's implementation. We applied a grid search to tune parameters such as penalty, C, and solver. By utilizing k-fold cross-validation, we were able to identify and address overfitting. The model achieved a cross-validation accuracy of 0.845 and a test accuracy of 0.825. The corresponding F1 score was 0.704.

2) PCA + Logistic Regression: In addition to regular Logistic Regression, we also employed PCA in Logistic Regression. Principal Component Analysis (PCA) is a dimensionality reduction technique. We decided to include this model to assess the impact of dimensionality reduction on the overall performance, reducing our dataset down to 3 components. It allowed us to determine if reducing the number of features improved the classification results. The model achieved a cross-validation accuracy of 0.845 and a test accuracy of 0.815. The corresponding F1 score was 0.698.

3) XGBoost: XGBoost (Extreme Gradient Boosting) is another ensemble learning method that combines multiple weak learners to create a strong predictive model. We chose this model for its ability to handle large datasets and its flexibility in hyperparameter tuning. We used the XGBoost package in Python to implement the model. The model was tuned using parameters such as learning rate, depth, estimators, and min_child_weight. It achieved a cross-validation accuracy of 0.878 and a test accuracy of 0.884. The corresponding F1 score was 0.817.

4) Random Forest: Random Forest is an ensemble learning method that uses multiple decision trees to make predictions. We selected this model because it has proven to be effective in various classification tasks, especially when dealing with complex data relationships. Sklearn's random forest model was our choice again due to our familiarity with the package. The grid search allowed us to tune the number of estimators, depth, min_samples_split, and min_samples_leaf. The model achieved a cross-validation accuracy of 0.876 and a test accuracy of 0.870. The corresponding F1 score was 0.788.

5) Neural Network: Neural Networks are an interconnected set of nodes which each individually work similarly to logistic regression. We chose to develop a neural network because they show a great degree of prowess in solving all kinds of different problems, classification included. Prior research on similar disease classification data sets suggested that a shallow network architecture would perform well [9].

From this, we then tested a variety of network architectures with either 1 or 2 hidden layers each with anywhere from 10 to 100 neurons in each layer. After some trial and error, we found that networks with 1 hidden layer tended to perform better on our data set. We then ran a grid search to find the best-performing network. The parameters of the grid search were as follows: hidden size, epochs, and learning rate. Hidden size ranged from 10 to 100, epochs also ranged from 10 to 100, and learning rate ranged from 0.001 to 1. The grid search found that the best parameters were ten hidden layers, one hundred epochs, and a learning rate of 0.001. Finally, we experimented with a few different optimizers: adam, sgd, and adadelta finding the best results with adam. Our best-performing model achieved 0.862 cross-validation accuracy, 0.914 test accuracy, and 0.910 test F1 score.

C. Model Tuning

Model tuning is an essential step in the data mining process as it allows us to optimize the performance of our models. The objective of model tuning is to find the best combination of hyperparameters or features that maximize the accuracy or performance of the models.

In our project, we employed hyperparameter tuning, where we systematically explored different combinations of hyperparameters to find the optimal values using sklearn's GridSearchCV function. For example, in logistic regression, we tuned parameters such as penalty, C, and solver. For random forest and XGBoost, we tuned parameters like the number of estimators, depth, and minimum samples split/leaf. In the neural network, we tuned the learning rate, optimizer, hidden layers, and epochs. These were all done through loops on a parameter grid for each model that stored different values for the parameters we wanted to optimize.

Model tuning had a significant impact on the performance of our models. Through the tuning process, we were able to identify the best set of hyperparameters or features that maximized the models' accuracy. This resulted in improved performance in terms of cross-validation accuracy, overall accuracy, and F1 score.

VI. Results

A. *Logistic Regression*

Now that we have described our preprocessing, the different models, and the tuning taken to improve then, we will start discussing our results, beginning with the logistic regression model. This was the simplest model we employed, yet its performance was admirable. Figure 9 shows it had a high amount of true positives and true negatives, resulting in an accuracy value of 82.5%. Relative to the other models which will be discussed shortly, logistic regression did struggle with false negatives, predicting 57 cases of metabolic syndrome as no metabolic syndrome. However, its cross-validation accuracy was similarly high to that of accuracy, at 84.5%.

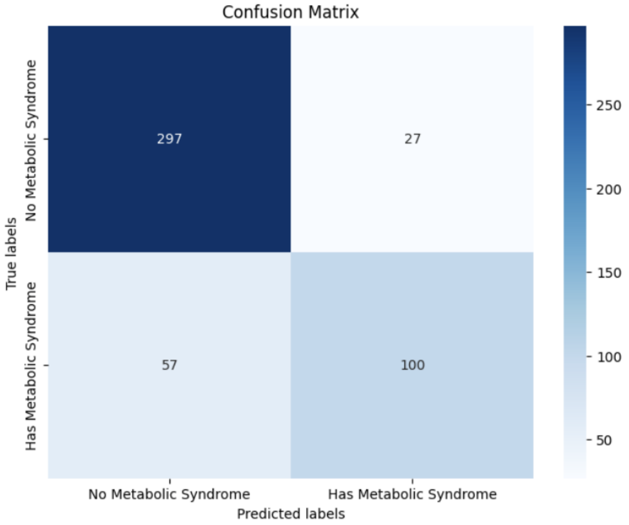


Fig. 9. Logistic Regression Confusion Matrix

To expand our analysis, we utilized F-1 score, which provides a balance of precision and recall for binary classification. This method resulted in a F-1 score of 0.704. Immediately, we noticed that the results exceeded our expectations, specifically the F-1 score. And, this was our simplest method, which led us to intrigue about how our additional methods would perform.

In addition to the model's performance, we also analyzed feature importance to grasp what features the model relied on versus those of less use. In Figure 10, we see that triglycerides, sex, and waist circumference were the three most important features in our model, where variables like UrAlbCr and income were practically useless. In further iterations of our work, we would utilize this to improve feature selection and reduce the dimensionality of our dataset where applicable. We were pleasantly surprised with the results for our first model and excited to see how our next few would perform.

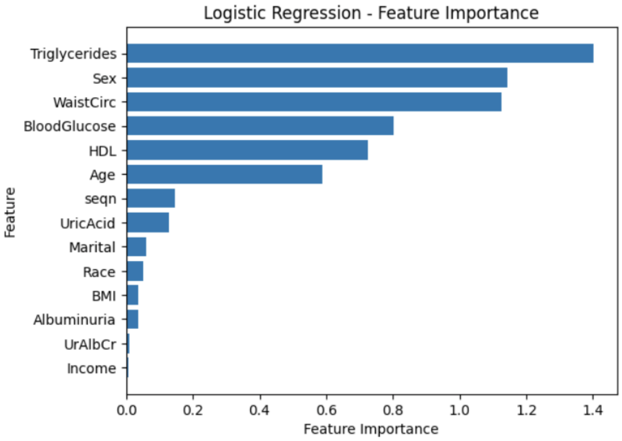


Fig. 10. Logistic Regression Feature Importance

B. *PCA + Logistic Regression*

Moving on to our next model, we utilized principal component analysis to reduce our data and then fed it into a logistic regression model. The idea here was that we might be able to reduce our dataset and still effectively capture the information we needed to create a highly performing-model. Our confusion matrix in Figure 11 shows that this iteration performed slightly worse, with fewer true positives and true negatives than the logistic regression version. It also resulted in more false positives. However, even with those drawbacks, it still had an overall accuracy of 81.5% and a cross-validation accuracy of 84.5%.

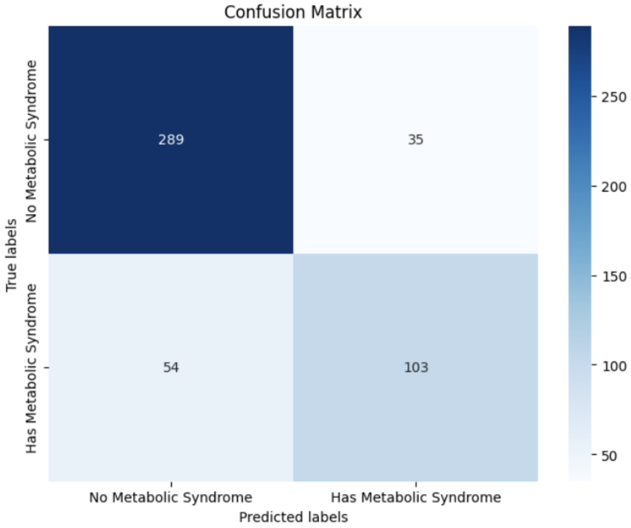


Fig. 11. PCA + Logistic Regression Confusion Matrix

The F-1 score dropped slightly from just logistic regression, going down to 0.698. For the amount of data reduction PCA performed, the evaluation metrics being as high as they are was encouraging, showing similar performance to the logistic regression model. Looking at our feature importance plot in Figure 12, we see that the model relied heavily on

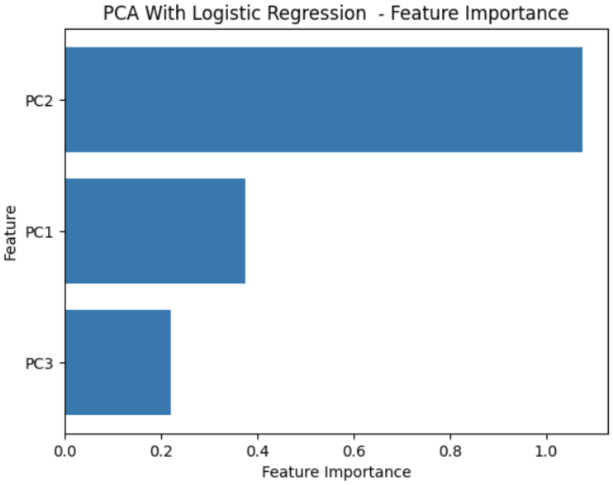


Fig. 12. PCA + Logistic Regression Feature Importance

principal component 2. With more time, we would have liked to further break down the components and see what features went into each. The reliance on component 2 makes us wonder how an iteration of just using one component in a logistic regression model would perform and would be something we would try down the line. Overall, we were encouraged by this combination's results but still wanted to improve on them.

C. XGBoost

XGBoost was chosen as another method because of its generally strong performance but also its ability to capture non-linear patterns in data, as well as capture possible interactions between features. Figure 13 shows the confusion matrix results of our model on the test set, and it performed better than the two logistic regression methods, reaching 300 true positives and 125 true negatives. It also had fewer false positives and false negatives than the prior two methods, resulting in an overall accuracy of 88.4% on the test set. It also achieved a cross-validation accuracy of 87.8% and a F-1 score of 0.817.

XGBoost outperformed our expectations all around. Another perk of XGBoost is it provides an easy way to generate feature importance, which we did as seen in Figure 14. HDL, WaistCirc, and Triglycerides were the three most important features of this model, while it barely relied on UricAcid, Income, and Race. The logistic regression model, while in a different order, held a similar pattern in terms of feature importance, which was reassuring. However, HDL was not in the top three for logistic regression, but it was the predominant feature for XGBoost. These nuanced differences warrant further investigation that we would have loved to examine with more time.

D. Random Forest

Moving on to our random forest model, we employed this because they use ensemble methods on decision trees, providing robust predictions, and minimizing the impact of

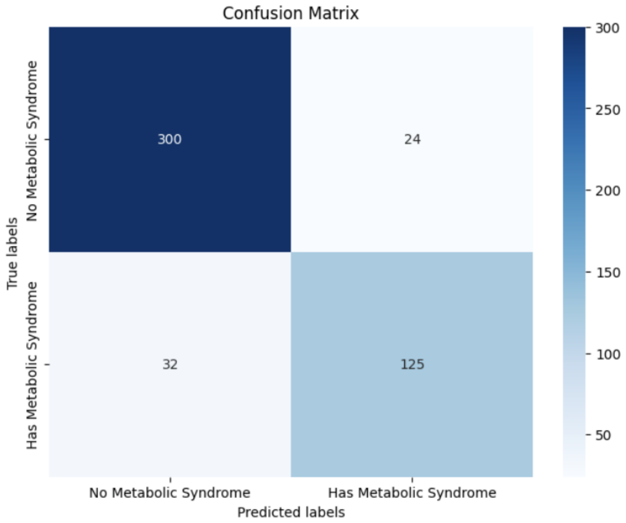


Fig. 13. XGBoost Confusion Matrix

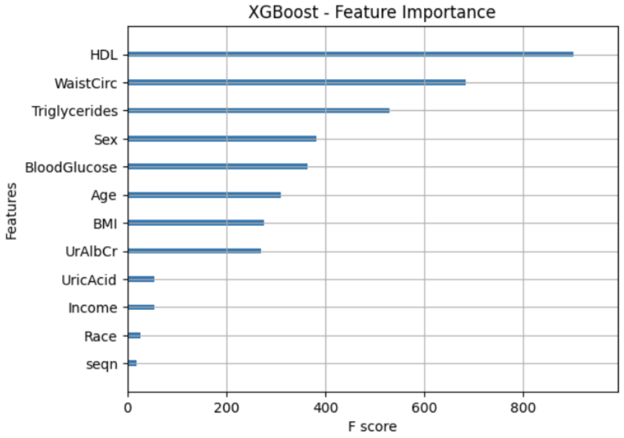


Fig. 14. XGBoost Feature Importance

outliers. Its performance was stronger than the two logistic regression methods, but slightly worse than XGBoost. As seen in Figure 15, it achieved 298 true positives, and 119 true negatives, but had slightly more false positives and false negatives than XGBoost. This resulted in an overall accuracy of 87%, along with a cross-validation accuracy of 87.6%.

The drop-off was more noticeable when analyzing the F-1 score, which was only 0.788. Granted, this meets our initial expectations with flying colors, but we still wanted to find the best model possible.

Random forest also allows for feature importance analysis, which we constructed in Figure 16. Similar to logistic regression, triglycerides were the most important variable in this model. It also relied on blood glucose more than the previous models. One observation worth noting is that it relied on the sex feature much, much less than previous models. Categorical variables in general seemed of less use to the random forest iteration, which would be another item we would like to investigate with more time. Being able

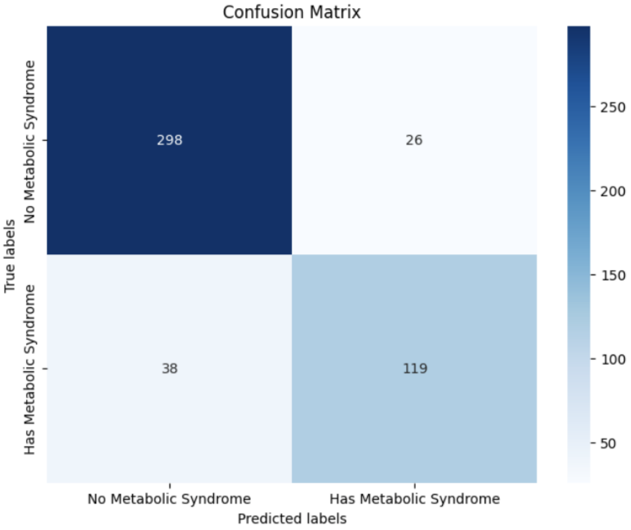


Fig. 15. Random Forest Confusion Matrix

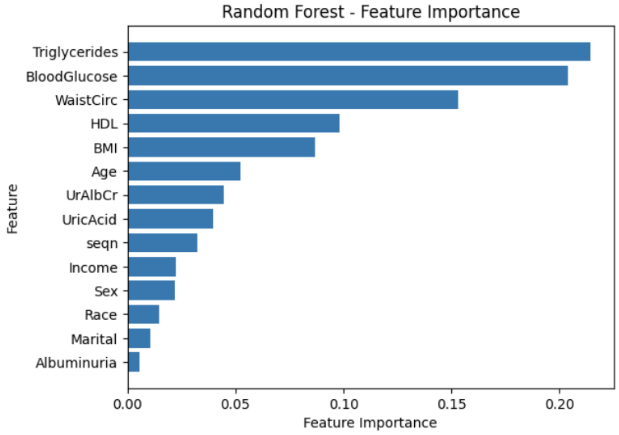


Fig. 16. Random Forest Feature Importance

to reduce the input features to just traits an individual can control, which would not include their race or sex, might lead to doctors being able to give better explanations to patients on what could be causing a disease. It would also provide more actionable advice on how to improve one's health after being detected for metabolic syndrome.

E. Neural Network

Lastly, our neural network implementation was the most complex type of model we utilized. Their scalability and adaptiveness appealed to us, and for good reason. Our confusion matrix in Figure 17 shows the model's strong results. Note that we dropped observations with null values instead of imputing them for the neural network, which is why the raw totals of each are not as high as the previous models. However, the neural network had the highest true positive rate and minimized false positives and negatives the best. This is reflected in its true accuracy on the test set, which was a resounding 91.4%.

Its F-1 score was also really high, coming in at 0.910. In terms of performance solely on the test set, this was our best model. However, it did relatively struggle with cross-validation accuracy, which was 86.2%. This is notably much less than its true accuracy and worse than the random forest and XGBoost implementations. We believe this could be due to the model overfitting. Another drawback of neural networks is their lack of interpretability. Their black-box nature and utilization of layers make it hard to decompose predictions and understand how the model arrived at certain values. They also do not offer feature importance values, which in the scope of this project, is a major drawback. While the results were super encouraging, the lack of transparency is something that requires our attention.

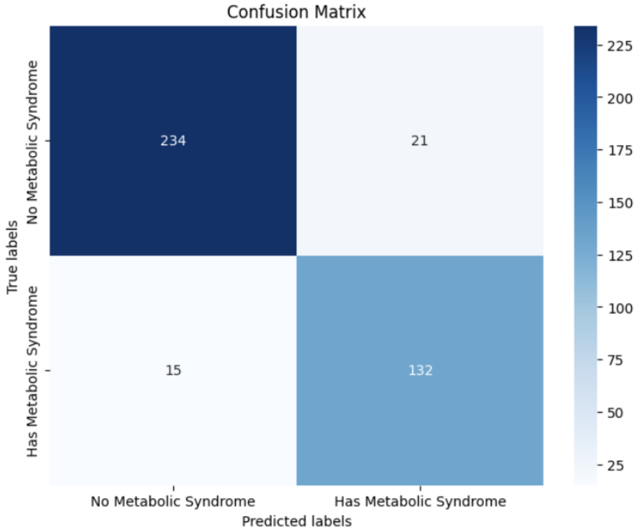
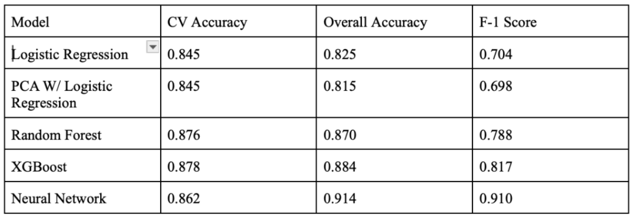


Fig. 17. Neural Network Confusion Matrix

F. Overall

Our overall results left us very satisfied with how this project turned out. Each of the model's cross-validation accuracy, pure accuracy, and F-1 score can be found in Figure 18. We see that the neural network performed the best as a whole, but lagged behind XGBoost and random forest in terms of cross-validation accuracy, indicating possible overfitting.



Model	CV Accuracy	Overall Accuracy	F-1 Score
Logistic Regression	0.845	0.825	0.704
PCA W/ Logistic Regression	0.845	0.815	0.698
Random Forest	0.876	0.870	0.788
XGBoost	0.878	0.884	0.817
Neural Network	0.862	0.914	0.910

Fig. 18. Model Results

XGBoost had the second-highest accuracy and F-1 score and is also a much more interpretable model than a neural network, which is more like a black box. XGBoost would provide a sound method for patients, doctors, and whoever else to understand what is being used to model metabolic syndrome whereas a neural network would be much more complicated to break down and analyze. The small trade-off in accuracy may be worth the advanced interpretability that XGBoost offers. At the same time, there is a strong disparity in the F-1 score that shows the neural network having stronger performance regarding false positives and false negatives, which are of utmost importance with medical diagnoses. Testing each model on larger datasets would be a key next step that could help us truly distinguish which we would recommend. It's possible using both in conjunction could be the best step forward. Nonetheless, our strong results left us quite encouraged by the usability of these models and their potential application.

VII. Conclusion

In conclusion, our project successfully classified metabolic syndrome with high accuracy using various modeling techniques. We built and analyzed results from 5 different models, including a Logistic Regression model, a PCA+Logistic Regression model, a Random Forest model, an XGBoost model, and a Neural Network model. Most notably, the Neural Network model had the highest test accuracy classifying Metabolic Syndrome at 91.4%. However, we found possible overfitting with the Neural Network model as XGBoost had the highest cross-validated accuracy at 87.8%. The interpretability of the XGBoost model compared to the neural network is also a major feature that makes us appreciate the former much more.

The results of our analysis have several implications for future research and practical applications. Firstly, the high accuracy achieved by our models suggests that it is possible to accurately classify Metabolic Syndrome using commonly available demographic and biological features. This opens up the possibility of early detection and disease prevention, which could have a significant positive impact on healthcare outcomes, reducing the overall prevalence of some of the secondary diseases including diabetes and cardiovascular disease.

For data scientists, our findings highlight the importance of model tuning and using different techniques to achieve optimal performance. While the neural network model was our best performing, it is much less interpretive than XGBoost, for example, which had similar performance but is much easier to explain to those unfamiliar with the model. This would be useful to keep in mind, especially when working with doctors and patients. Future research could build upon our findings by exploring additional modeling techniques or evaluating the performance of our models on larger and more diverse datasets. For example, metabolic syndrome may have different prevalence rates in different races or income groups. Building out separate models on different minority datasets could help increase accuracy across subclasses of our dataset, and is something we would have liked to try with more time and a more appropriate dataset.

For practitioners and businesses in the healthcare industry, our models provide a practical tool for identifying individuals at risk of Metabolic Syndrome. This could help healthcare professionals prioritize their efforts, target interventions, and improve patient outcomes. Additionally, our models could be incorporated into existing clinical decision support systems to provide real-time risk assessment for individuals. They would greatly reduce the amount of stress put on hospitals and doctors alike in identifying the syndrome, easing their jobs while also providing patients with quicker results. Streamlining the identification process would allow doctors to spend their efforts on more dire needs while not sacrificing the needs of other patients.

In summary, our project achieved high accuracy in classifying Metabolic Syndrome, exceeding our baseline expectation of accuracy of 80% and an F-1 score of around 0.65. The findings of our research have the potential to advance the field of early disease detection and prevention. By automating the classification process, we can save manual labor and improve response times for patients. The models developed in this project provide a framework that can be applied to larger datasets, enabling early detection for larger populations at increased efficiency. We hope that our findings can contribute to the body of knowledge in this field and inspire further innovation in metabolic syndrome classification and further prevention of secondary diseases.

References

[1] "Metabolic Syndrome." *Johns Hopkins Medicine*, https://www.hopkinsmedicine.org/health/conditions-and-diseases/metabolic-syndrome. Accessed 13 December 2023.

[2] "Metabolic Syndrome - What Is Metabolic Syndrome?" *NHLBI*, 18 May 2022, https://www.nhlbi.nih.gov/health/metabolic-syndrome. Accessed 13 December 2023.

[3] Li, Yuqing et al. "Predicting metabolic syndrome by obesity- and lipid-related indices in mid-aged and elderly Chinese: a population-based cross-sectional study." *Frontiers in endocrinology* vol. 14 1201132. 28 Jul. 2023, doi:10.3389/fendo.2023.1201132

[4] Yang, H., Yu, B., OUYang, P. *et al.* Machine learning-aided risk prediction for metabolic syndrome based on 3 years study. *Sci Rep* 12, 2248 (2022). https://doi.org/10.1038/s41598-022-06235-2

[5] Zou, G., Zhong, Q., OUYang, P. *et al.* Predictive analysis of metabolic syndrome based on 5-years continuous physical examination data. *Sci Rep* 13, 9132 (2023). https://doi.org/10.1038/s41598-023-35604-8

[6] Antony, Albert. "Metabolic Syndrome." *Kaggle*, 27 Oct. 2023, www.kaggle.com/datasets/antimoni/metabolic-syndrome.

[7] "sklearn.metrics.f1_score — scikit-learn 1.3.2 documentation." *Scikit-learn*, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html. Accessed 15 December 2023.

[8] "Metabolic Syndrome: What It Is, Causes, Symptoms & Treatment." *Cleveland Clinic*, 13 September 2023, https://my.clevelandclinic.org/health/diseases/10783-metabolic-syndrome. Accessed 15 December 2023.

[9] Ahsan, M. M., Luna, S. A., & Siddique, Z. (2022). Machine-Learning-Based Disease Diagnosis: A Comprehensive Review. *Healthcare (Basel, Switzerland)*, *10*(3), 541. https://doi.org/10.3390/healthcare10030541