Disease Prediction Using Machine Learning

Data Mining and Statistical Learning ISYE 7406 26 Nov, 2023

Group 10 Team Members

Name	GTID	Email Address
Nate Casey	ncasey30	ncasey30@gatech.edu
Amanda Wijntjes	awijntjes3	awijntjes3@gatech.edu
Balaji Marimuthu	bmarimuthu3	bmarimuthu3@gatech.edu
Andoni Sooklaris	asooklaris3	andoni@gatech.edu
Syefira Shofa	sshofa3	sshofa3@gatech.edu

Contents

Abstract

Advancements in machine learning have paved the way for innovative approaches to healthcare, particularly in disease diagnosis. This study focuses on the development of machine learning models for disease prediction to enhance the accuracy of patient diagnoses, with the ultimate objective of improving healthcare outcomes. These models serve as valuable tools for healthcare professionals, aiding in precise and timely assessments. By understanding disease distributions and symptom patterns, targeted and accurate machine learning models are created, aligning with the increasing adoption of value-based healthcare in the medical field.

Our models aim to contribute to this industry shift by building, validating and providing tools that enhance diagnostic accuracy, leading to better patient outcomes and emphasizing the broader objective of value-based healthcare, which prioritizes outcomes and patient satisfaction.

We perform an extensive exploratory data analysis to understand the dataset and its important properties. Various supervised machine learning models are explored, including lasso regression, principal component analysis, k-nearest neighbors, support vector machine with differing assumptions, and random forest to determine their efficacy in accurately predicting disease diagnoses from the given symptom predictors.

Our research underscores the potential of machine learning models in predicting disease diagnoses from symptom indicators. Through extensive experimentation and model evaluation, our findings reveal promising results in using machine learning models for disease prediction, though with certain caveats detailed in the body and conclusion of our report. Future research directions include exploring more complex model architectures and incorporating additional data sources with greater volumes of data to further enhance predictive accuracy and generalizability in real-world clinical settings. These enhancements would be requirements before any modeling tactics detailed in this study could be applied in the real world.

Introduction

Our main objective is to develop supervised machine learning models capable of effectively classifying a specific disease based on their reported symptoms. Before model development, we conducted an in-depth Exploratory Data Analysis (EDA). This analysis aimed to understand the distribution of diseases within the dataset and to provide insights into how symptoms are distributed among patients.

It has become increasingly important for medical professionals to deliver value-based healthcare to their patients. Data-driven insights and models can be used to help increase the accuracy of patient diagnoses, and thereby improve patient outcomes, that will positively impact healthcare and medical care. After conducting EDA, the supervised machine learning models that we explore include lasso regression, principal component analysis, k-nearest neighbor, support vector machine with differing assumptions, linear discriminant analysis, quadratic discriminant analysis, naïve bayes, and random forest.

Problem Statement and Data Sources:

The dataset, obtained from Kaggle, features 132 symptoms that a patient did or did not experience, as well as a medical prognosis. A patient's symptoms serve as predictors, with each symptom being a binary variable: 1 indicating a patient has the symptom, and 0 indicating a patient does not have the symptom. Due to the number of predictors, it is not entirely feasible to display a full sample observation in this report. Example predictors include skin rash, itching, joint pain, shivering, and chills. In the context of this dataset, the prognosis is a multinomial categorical response variable and can be one of 42 diseases. Example diseases include Allergy, AIDS, Diabetes, Fungal Infection, Malaria and Chicken Pox. A patient will have one disease and any number of symptoms. We mapped each disease/prognosis to a number from 1 to forty-two for modeling purposes.

Exploratory Data Analysis (EDA)

EDA is a crucial first step in data mining as it provides insights into the dataset's intricacies, helping us make informed decisions during model development.

Skewness

Table 1. Skewness for symptoms.

Symptom	Skewness
Foul Smell of Urine	6.7248268
Ulcers on Tongue	6.5246558
Dischromic Patches	6.5246558
Nodal Skin Eruptions	6.5246558
Shivering	6.5246558
Watering from Eyes	6.5246558
Spotting Urination	6.5246558
Extra Marital Contacts	6.5246558
Muscle Wasting	6.5246558
Patches in Throat	6.5246558

In our dataset, each of our predictor variables is a logical indicator, the values for which can only be 1 or 0. To quantify the observed logical distribution for each predictor, we utilize skewness as a statistical metric, providing insights into the asymmetry and shape of the binary variable distributions. Multiple predictors having the same skewness may have a consistent pattern in the spread, suggesting a uniformity in the distribution of binary values and potential similarities in their impact on the predictive model. A subset of 10 predictors/symptoms is shown in Table 1.

Correlation

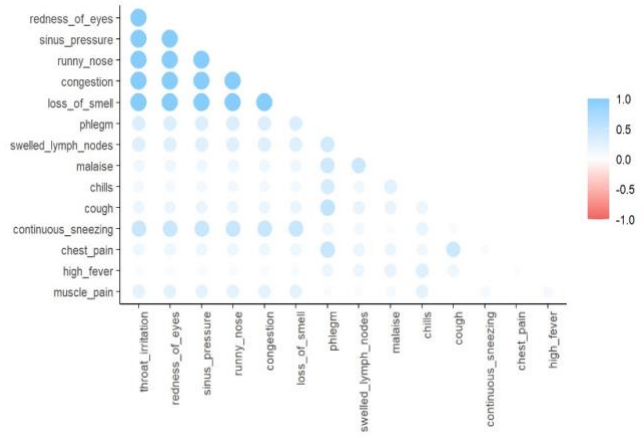

Figure 1: Correlation plot of symptoms.

In our dataset with 132 predictors, displaying the correlations among all variables would be impractical. Therefore, we've filtered to showcase only the highest absolute correlations. The correlation coefficient 'r' serves as a metric to gauge the strength and direction of associations in the dataset, and it is crucial in assessing potential linear relationships between continuous variables. The correlation plot visually represents how closely two variables co-vary, with +1 indicating a perfect positive correlation, 0 representing no correlation, and -1 representing a perfect negative correlation. For instance, throat irritation and redness of eyes are highly positively correlated and so are sinus pressure and a runny nose. Figure 1 shows there are no strong negative correlations in our data set.

Data Cleaning

Perfectly Correlated Variable & Multicollinearity

As described above, a perfectly correlated variable is one where the relationship between it and another variable is characterized by a correlation coefficient of 1. This indicates a precise linear association, where one variable can be expressed as a constant multiple of the other, leading to a redundancy of information between the two variables. Perfectly correlated variables in machine learning modeling can be problematic because they introduce multicollinearity, making it challenging for models to distinguish the individual impact of each variable. This can lead to unstable model coefficients, reduced interpretability, and potential performance issues, as the model may struggle to generalize well to new, unseen data. Thus, we only kept one of the variables for each set of perfectly correlated variables.

The following sets were perfectly correlated:

- Throat irritation, redness of eyes, sinus pressure, runny nose, congestion, loss of smell
    - These symptoms are often correlated because they commonly result from shared causes such as respiratory infections, allergic reactions, or specific health conditions, indicating a systemic response to external factors affecting the respiratory and immune systems.
- Brittle nails, swollen extremities, enlarged thyroid
    - These symptoms are correlated as they can collectively indicate an underlying health condition, such as hypothyroidism, where a dysfunctional thyroid gland affects both nail and tissue health, resulting in characteristic manifestations.

Linearly Dependent Variables

We also removed linearly dependent variables; predictors that are linearly dependent on other variables contribute redundant information in a dataset, leading to multicollinearity issues in linear regression models and making it difficult for the model to estimate unique coefficients for each predictor, potentially impacting the model's stability and interpretability

The symptoms that are linearly dependent on other variables are the following: polyuria, receiving unsterile injections, stomach bleeding, palpitations. These symptoms could be linearly dependent if they are associated with a common underlying condition, such as diabetes or complications from unsterile injections, suggesting a proportional relationship in their occurrence.

Variance Inflation Factor (VIF)

We used VIF to address further issues with multicollinearity. VIF in machine learning quantifies the extent of multicollinearity among predictor variables, with high VIF values indicating inflated variance in regression coefficients due to strong correlations between predictors. Any variables with a VIF value over 10 were removed.

Proposed Methodologies

Based on our dataset, we are focused on determining the most predictive symptoms for accurate disease classification utilizing supervised machine learning models. Ultimately, we would like to explain the variability in the response by leveraging mathematical combinations of the predictors. To achieve this goal, we plan to utilize a range of data mining and statistical learning methods.

Because we are working with a large dataset of 132 independent variables, the below listed methodologies were used for our model prediction.

Principal Component Analysis (PCA)

PCA is a technique in machine learning that reduces the dimensions of data, transforming it into a simpler representation. It highlights key patterns and captures the most important variance by identifying independent principal components.

Cross Validation

5-fold cross-validation with 3 repeats involves splitting the dataset into 5 subsets, iteratively using 4 for training and 1 for validation; this process is repeated 3 times, providing a more robust evaluation of a model's performance by assessing it across different combinations of training and validation sets.

KNN

KNN (k-Nearest Neighbors) is a machine learning algorithm that classifies based on the majority class of the k-nearest data points in the feature space, making decisions locally by considering the closest neighbors in the dataset.

Random Forest

Random Forest is an ensemble learning algorithm that constructs multiple decision trees during training and outputs the mode (classification) of the individual trees, providing robust and accurate predictions while mitigating overfitting.

LASSO Regression

Lasso regression, a powerful machine learning technique for feature selection, introduces a penalty term (lambda, denoted as λ) to encourage the model to reduce less relevant feature coefficients to zero. This prevents overfitting by excluding irrelevant features. Lasso regression stands out for feature selection, contributing to accurate machine learning models. Balancing high accuracy with guarding against overfitting is crucial for its successful real-world application.

Support Vector Machine

A Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It works by finding the optimal hyperplane that best separates different classes in the feature space, maximizing the margin between data points of different classes.

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a classification technique that seeks linear combinations of features to maximize the separation between classes while minimizing within-class variance. It accomplishes this by projecting the data onto a lower-dimensional space defined by discriminant functions, making it an effective method for distinguishing between multiple classes.

Quadratic Discriminant Analysis (QDA)

QDA is a variant of LDA that allows for different covariance matrices for each class, providing greater flexibility in capturing the shape of class boundaries. QDA models the distribution of each class using quadratic decision boundaries, accommodating scenarios where classes have distinct variances.

Naïve Bayes

Naïve Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes feature independence within classes, meaning that the presence or absence of a particular feature does not influence the presence of other features. Despite its simplistic

assumption, Naïve Bayes is computationally efficient and often performs well in practice, particularly in situations where feature independence is a reasonable approximation.

Analysis and Results

Table 2. Summary of modeling testing error.

Model	Testing Error	Testing Error PCA
Lasso Regression	0	NA
K Nearest Neighbor	0.046	0.131
LDA	0.051	0.049
Naïve Bayes	0.087	NA
Random Forest	0.039	NA
Support Vector Machine (Gaussian "radial")	0.041	0.931
Support Vector Machine (Polynomial)	0.051	0.931
Support Vector Machine (Linear SVM)	0.039	0.921
Support Vector Machine (Gaussian-kernel SVM)	0.041	0.932

Principal Component Analysis

Principal Component Analysis was utilized post data cleaning, which reduced our predictors from 132 to 48. The chart to the right represents the cumulative proportion of variance explained per added principal component. 24 principal components were kept of the 48 that were created since retaining only the principal components that explain a significant proportion of the variance helps in avoiding overfitting. This was enough to explain 91% of the variance in the data. Note that principal components 25-48 contributed 1% or less of the variance in the data each.

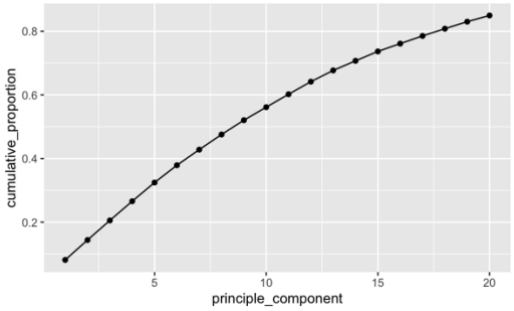

Figure 2: Cumulative proportion of variance explained per added principal component

The first principal component represented 8% of the data. The chart to the right showcases the variables that contributed the most to the first PCA. The first Principal Component is like the biggest piece in the puzzle, representing the main pattern. Understanding which variables of the data contribute a lot to this big piece helps to figure out the most

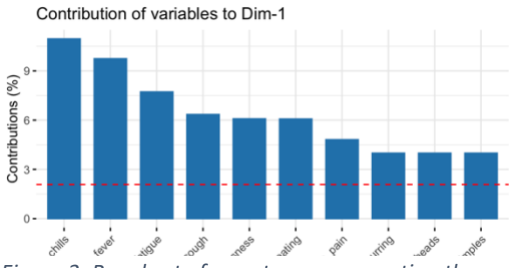

Figure 3: Bar chart of symptoms representing the

important factors influencing the overall picture. Thus, chills, high fever and fatigue contribute the most to the variation in the data. The variation in a dataset is crucial as it encapsulates the diversity of values, enabling pattern recognition, informing decision-making, and influencing the accuracy of models and analyses.

Lasso Regression

In our analysis, cross-validation identified an optimal lambda of 0 among 100 lambdas, suggesting no penalty on feature coefficients. A lambda of 0 implies a model without regularization, akin to standard linear regression. While this closely fits the 80% training and 20% test split, caution is needed to avoid overfitting, especially with noisy datasets or high feature-to-observation ratios. Different features were selected for each classification in a multinomial dataset, highlighting the risk of overfitting.

The test error was 0, resulting in 100% accuracy. This could stem from minimal differences in true features within a prognosis class and an abundance of features per class with little overlap, making classification easier. On a new, unseen dataset, the model will most likely overfit with numerous features, emphasizing the importance of feature selection. Future steps involve evaluating patient populations and demographics, particularly focusing on classes with more observations for improved model generalizability. Given the potential overfitting risks associated with Lasso regression and a lambda of 0, exploring alternative regularization techniques, such as Ridge regression, is worthwhile. Ridge applies a different penalty on coefficients, providing a balance between model complexity and generalization.

K-Nearest Neighbor (KNN)

For KNN, we utilized the Jaccard distance, measuring the dissimilarity between two sets by comparing the size of their intersection to the size of their union. In KNN, using Jaccard distance as a similarity metric allows the algorithm to focus on the similarity of the set elements, which can be particularly relevant when dealing with categorical features or instances where the presence or absence of items matters more than their numeric values. Implementing the Jaccard distance for k-nearest neighbors (KNN) can be useful in scenarios where the data consists of sets or binary attributes.

We utilized the KNN with the Jaccard distance on 24 principal components with 5 fold cross validation. The chart to the right showcases that k = 3 is the optimal model when using root mean square error (RMSE) as the key performance indicator (KPI). Since we can see that as k gets smaller, RMSE is also lower, there may be problems with the dataset. A small k can be problematic as it introduces high sensitivity to noise and outliers, making the model prone to overfitting and leading to less robust predictions.

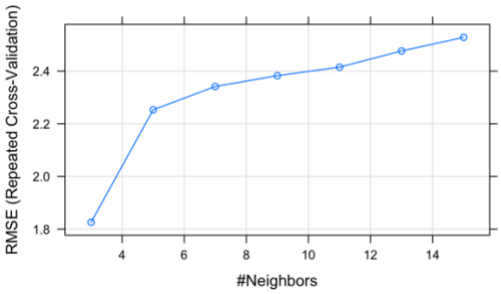

Figure 4: Elbow char for KNN using RMSE.

The testing error for KNN with PCA is higher than KNN without PCA. This suggests that PCA may not have been advantageous for the specific dataset or problem. Potential reasons for this outcome include information loss during dimensionality reduction, inappropriate selection of the number of principal components, and PCA's assumption of linear relationships not aligning well with the underlying data patterns. PCA's assumption of linear relationships may not perform well on binary predictors as the limited variability, discrete nature, and potential nonlinear associations of binary variables may hinder the effectiveness of PCA in capturing essential patterns. Additionally, the curse of dimensionality and the sensitivity of KNN to local data density might play a role. It highlights the importance of carefully assessing the characteristics of the dataset, considering the limitations of PCA, and potentially exploring alternative dimensionality reduction methods or hyperparameter tuning to enhance model performance.

Random Forest

The next model we tested was random forest, which was tied with support vector machine (SVM) for our best performing model. To find the best model, we iterated through different parameters including the number of trees and the depth of each to find the random forest that performed the best against testing data. The optimal number of variables to sample at each split was 6, and the best number of trees was 300.

In addition, we looked at the importance of each feature in the random forest which can be seen in the Figure 5 below.

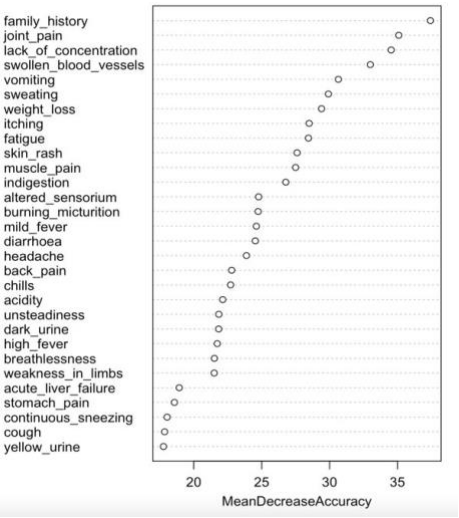

Figure 5: Importance of features in random forest.

Importance examines how much each feature changes accuracy within the resultant leaves of the tree. As can be seen above, there are a few features that contribute much more to these calculations than others. We fitted a model with only the five most important features to compare to the full model, and it performed similarly. Because it is a simpler model with similar performance, we chose this reduced model as the final random forest model.

Overall, this model showed many of the same issues that we found with the other models. Specifically, it produced a testing error of 3.9% which is unrealistically low. Such a low testing error rate raises concerns of very high overfitting which would need to be addressed before deploying this model in any way.

Support Vector Machine

qof the fitted support vector machine models, the linear support vector machine with the "vanilladot" kernel yielded the lowest testing error. A linear SVM uses a linear decision boundary, often referred to as a hyperplane, to separate different classes in the feature space. The linear SVM might have performed best among the kernels tried for several reasons. The classes in our dataset are well-separated by a linear boundary, so a linear kernel may excel in finding a hyperplane that effectively separates the classes. Additionally, a simpler linear model may generalize better and avoid overfitting, especially considering the high dimensionality of the feature space.

We observed the testing errors with PCA showed drastically high testing errors. It could indicate that reducing the number of features using PCA didn't help the SVM perform better. PCA might have removed important information needed for accurate predictions. In this case,

keeping all the original features helped the SVM see and understand the picture better, resulting in more accurate predictions.

LDA, Naïve Bayes, and QDA

When fitting QDA, the "rank deficiency in group 1" error occurred. This error typically indicates an issue with the covariance matrix estimation which can occur when there is a lack of variability in one of the classes, causing the covariance matrix to be singular or nearly singular. A similar problem was observed when fitting a lasso regression.

In our dataset, the superior performance of Linear Discriminant Analysis (LDA) compared to Naive Bayes suggests that the assumption of equal covariances across classes, a critical assumption in LDA, aligns well with the underlying data distribution. This alignment likely contributes to more accurate classification, as LDA effectively models the relationship between features and classes based on the specific characteristics of our dataset.

Furthermore, the observation that LDA provides results while Quadratic Discriminant Analysis (QDA) throws an error in our results may indicate challenges in our dataset. QDA is more sensitive to data irregularities, especially in situations with a limited sample size or highly correlated features, potentially resulting in issues such as rank deficiency in covariance matrices. The resilience of LDA to these challenges makes it a dependable choice when the assumptions of equal covariances and sample size limitations align with the unique features of our dataset.

Conclusion

Overall, each of the models built on this data performs at an incredibly high rate with the worst testing error being only 15% and with most models being around 5%. These values raise concerns of further correlated terms within the data, specifically that a predictor is very highly correlated with the dependent variable that we are trying to predict and there is limited variability within classes in our dataset. Although we employed multiple different methods—PCA, Lasso, etc.—to attempt to address this concern, the performance of the models is still seemingly, unrealistically good. Before moving forward with any of these models, more work would need to be done to understand the relationships in the data and why the models are performing so well. Based on our results and errors when fitting some of the models or attempting cross validating, it is most likely because there is a lack of variability in our classes causing the covariance matrix to be singular or nearly singular.

The data set with which we worked was relatively small, and a larger data set could help to alleviate some of these issues. This would allow us to expose our models to a larger variety of symptomologies and resultant diagnoses and would make it easier to trust the results of our models. In a real-life situation, symptoms and diagnoses of patients could be logged as they came to the healthcare facility and then used to continue training and testing the model.

Our analysis shows that machine learning models can help predict diseases using symptoms, which can be used to improve patient outcomes. These models serve as valuable tools for healthcare professionals, aiding in precise and timely assessments. By understanding disease distributions and symptom patterns, targeted and accurate machine learning models can be

created. In the future, we plan to explore more advanced models and include more data to make the predictions even more accurate in real-world situations. Before using the methods, we discuss in real life, we need to make improvements such as broadening the dataset.

Lessons Learned - Project

One lesson learned is that reducing the number of predictors was helpful. Our original 132 predictors added a lot of noise and issues with multicollinearity to our data and models. There was no reduction in predictive power after removing the majority of predictors. When applying our methodologies to real patients, care would need to be taken to ensure that a critical symptom is not removed in that particular setting, but in the scope of our analysis, reducing the number of predictors was very beneficial.

PCA's performance can vary across different machine learning algorithms due to their underlying assumptions and sensitivity to different data structures. Because PCA performed poorly on SVM but well on KNN and exceptionally on Linear Discriminant Analysis (LDA), it may be because SVM relies heavily on capturing complex decision boundaries, and the linear transformations introduced by PCA might not effectively represent these non-linear structures. In contrast, KNN is more flexible and adaptable to local patterns, benefiting from dimensionality reduction without losing critical information. LDA, being a supervised method, may exploit the class information present in the data, aligning well with PCA's linear transformations and leading to superior performance.

We would select the random forest model based on the reduced 48 predictor data set as our final recommendation, but that recommendation would come with some concerns, as mentioned above. This was the best performing model, but the testing error is unrealistically low. We would **not** recommend this model be deployed in an actual healthcare setting until it had been exposed to much more data and refined. In a setting such as healthcare, incorrect diagnoses are far too detrimental to deploy a model that does not command very high confidence.

Lessons Learned - Course

Overall, this course was a fantastic overview of a large number of machine learning and data mining methodologies. The format of the class exposed us to many different ways of working with data, and the homework assignments were helpful in allowing us to be creative and to apply the concepts to real-world data. Additionally, peer assessments provided a great opportunity to receive feedback and further build our skill sets. It was beneficial to observe others' approaches and learn from them.

For future iterations of this course, it would be very helpful to see examples with code of expanding models to multinomial situations. For example, the code for logistic regression only demonstrated a binomial dependent variable. Although we understand that these simple examples are best for learning, it would be nice to have available more information on more complex implementations. Additionally, there were times that points were deducted from homework

assignments and projects for not using a concept, but that concept had never been covered or prescribed in class—t-test and Wald tests for selecting model.