DATA MINING

Objective: Finding the most accurate Covid-19 prediction model.

DATA MINING REPORT

This document primarily focuses on 2 models, Naïve Bayes (NB) and K nearest neighbors. (KNN), to find any correlations within a Covid-19 dataset, to determine the likelihood of Covid-19 classification.

Data Mining – M3DiB

Berkan Baskopru - 2136590
Joris Cornel – 1656167
25 october 2023

Contents

Objective

The objective for this report is to build multiple designs for models that will be tested on their efficiency and effectiveness. We will explain how the different models that we've chosen work. We will also evaluate which model is the best in this context. The dataset chosen to test the models is a Covid-19 dataset. The main goal for the dataset as set by the original publisher of the dataset was to build a machine learning model that, given a Covid-19 patient's current symptom, status, and medical history, will predict whether the patient is in high risk or not. For our own objective we tested tried to find the most accurate model to predict whether a patient does have Covid-19 or not purely based on the available information of the patient (see current symptoms, status, and medical history). How exactly we did this will be further elaborated below.

Furthermore, the original dataset and objective can be found in the following link:

https://www.kaggle.com/datasets/meirnizri/covid19-dataset

Data Preparation and understanding

To understand the dataset, a detailed description was made on the hyperlink above. This also included an explanation of the variables within the dataset. It became clear that the main focus in the dataset, would be the variable classification_final, since this is what we are trying to predict using the models.

In preparation for the actual use of the dataset itself, it was important that any missing values would be excluded from the models, before running the models on the code. While there was no actual missing datapoints in the dataset, the numbers 97 and 99 generally indicates missing values. This was not the case for the variable 'Age' since there was data available for people with 97 or 99 being their age at that moment. In preparation for the use of the data, we cleaned all data where a variable had missing values, by removing the datapoints 97 and 99 for every variable, except 'Age'.

Next to that we also simplified the column classification_final. In the original dataset the numbers 1-3 depict the severity of the patient's sickness. The values 4 and above meant that either the Covid-19 tests were negative or invalid. We simplified this to the values 1 for Covid-19 classified patients and the value 0 for non-Covid-19 or invalid classified patients. This because in our case the objective is not to predict the severity of Covid-19, only to see whether the patient does or does not have it. We also noticed a spike of around 5-10% in accuracy after this change.

Models

K-Nearest Neighbour (k-NN)

The K-Nearest Neighbour model is a quite simple machine learning algorithm. It can be used both as a classification or regression model. It works by classifying data points with attributes that are like each other. It makes predictions by calculating the "k" nearest data points in the training dataset to classify a new data point. In our case classification_final. It uses either classification (based on the majority) or regression (based on the mean).

Since the model is very sensitive to the choice of K, which is a hyper parameter, we tuned it to fit our specific dataset and context as much as possible.

Through a performance evaluation we can test how accurate our model is at guessing whether a patient has Covid-19 or not.

We tuned the 'K' and training set to get the highest possible accuracy. Some different inputs for 'K' got us higher accuracies on the '1' or '0' but took accuracy off the other. With our current inputs we have the most similar accuracy for both while maintaining the highest possible average accuracy.

Naive Bayes

The Naive Bayes model is the second machine learning algorithm we explored in our assignment to predict whether a patient has Covid-19 based on their available information, including current symptoms, status, and medical history. This classification model works as an algorithm which uses certain probability to determine the output. It splits the data into 2 sets: a training set, and a testing set.

In the training set the data will train the Naïve Bayes model, by learning from certain patters, relationships, and structures within the data. It will try to find correlations, in our case between the variables and the target variable: classification_final, which would indicate a covid-19 risk. All this will be done to eventually be able to make accurate predictions on new or unseen data.

The testing set is the other portion of the dataset, which is not used for training the model. This set is used to assess the performance of the model itself and will use the (as previously described) unseen data, to predict the classication_final variable. This way a certain accuracy of the model can also be calculated, which is ultimately an indication of how well the model performs on the dataset.

Others

We have also decided to try some different models to test the accuracy and see how our chosen models compete with others. This was to test if the accuracy is mainly affected by the dataset itself or our chosen models. We tried the Random Forest model, and the logistic Regression model. The models won't be further explained in detail as they were purely to test.

Outputs

K-nearest neighbours

Accuracy: 0.64

Classification report:

	precision	recall	f1-score	support
0	0.64	0.48	0.55	7240
1	0.63	0.77	0.70	8395
accuracy			0.64	15635
macro avg	0.64	0.63	0.62	15635
weighted avg	0.64	0.64	0.63	15635

Naïve Bayes

Accuracy: 0.5901503038055644

Classification Report:

	precision	recall	f1-score	support
0	0.59	0.41	0.48	14566
1	0.59	0.75	0.66	16704
accuracy			0.59	31270
macro avg	0.59	0.58	0.57	31270
weighted avg	0.59	0.59	0.58	31270

Number of mislabeled points out of a total 31270 points: 12816
Percentage of misclassified points: 40.98%

Other models

Random Forest Classifier Results:

Accuracy: 0.5957501280081925

Classification Report:

	precision	recall	f1-score	support
0	0.57	0.53	0.55	14493
1	0.62	0.65	0.63	16755
accuracy			0.60	31248
macro avg	0.59	0.59	0.59	31248
weighted avg	0.59	0.60	0.59	31248

Number of mislabeled points out of a total 31248 points: 12632

Percentage of misclassified points: 40.42%

Logistic Regression Classifier Results:

Accuracy: 0.6028225806451613

Classification Report:

	precision	recall	f1-score	support
0	0.59	0.48	0.53	14493
1	0.61	0.71	0.66	16755
accuracy			0.60	31248
macro avg	0.60	0.59	0.59	31248
weighted avg	0.60	0.60	0.60	31248

Number of mislabeled points out of a total 31248 points: 12411

Percentage of misclassified points: 39.72%

Conclusion

The objective was very clear, we wanted to identify the best model for predicting any covid-19 risk based on many variables within our dataset. We set our focus to 2 models, KNN (K-Nearest neighbours), and NB (Naïve Bayes). Out of these two, the KNN model was the one with the highest accuracy, with 64% (0.64). To determine if any other models were more suited for the dataset, we tried the Random Forest model, and the Logistic regression model. The accuracy results of these were very similar to each other, as well as the Naïve Bayes model, however, the KNN model is still the best performing of the 4.

Even though the KNN model is the best performing model, we would not recommend anyone to try to make accurate predictions using this model, as the accuracy is only 64%.

To improve the accuracy of the models, certain changes could be made to the dataset itself, such as:

1. Creating new variables within the dataset itself. You could try adding variables which might have an impact on the classification_final variable.
2. The dataset could be expanded with data from a more diverse range of people.

To predict whether a patient has Covid-19 or not with this dataset more accurately, one would have to build a model from the ground up or combine models together to meet this context specific needs. This could be a point to further research in the future.

Reflection

Berkan Baskopru

Personally, I have had a small background in coding. With HTML5, CSS, Java and a couple of others. But never in Python. However, I always wanted to try it out as it is the perfect language for business students to learn for the future. Since data will only become more prevalent.

I noticed that its very similar to other coding languages. However, a lot is being done via packages and no raw coding is necessary in most cases. It showed me how easy it can be to code a couple models to calculate specific things to gain insights from data that might have never come up otherwise. Coding the models was made very easy with the help of the packages and ChatGPT. Also being able to test your code quickly to see what is wrong with it and fix the errors is very helpful too. It's a skill you really learn by a lot of doing.

I noticed how much maths is involved and how deep the calculations and algorithms can get. Personally, I love data and maths so puzzling with models to try and answer a complex problem like whether a patient has covid or not just with the help of data and no actual analysis being involved really made me more and more interested in learning everything in the world of data science. For future projects I now also have a solid foundation to build up on.

Joris Cornel

With this assignment I was introduced to the world of coding within python, along with using data to make certain predictions. Since I had never coded before, except for VBA in Excel, it took a little time to get used to python itself. On many cases it is comparable with writing macro's in excel, except for the way certain code was written, and certain packages had to be imported.

It is also a big relieve, that we can nowadays rely on language models such as ChatGPT to find out certain things, which can be specifically answered to our situation. This way any errors within the code were easily spotted, and fixed. I also found it very helpful that you didn't have to start on the barebones when creating a piece of code. I feel like I mostly learned from generating a certain code with ChatGPT, only to try to understand what it does after. This way it was relatively easy to understand how to write the required code specifically for our dataset, and what the possibilities there were.

This assignment, along with any other assignments related to coding within python, enabled a certain interest for me, to explore the possibilities within python, and any other programming languages. In other words, I quite liked working in python in this assignment.