Erin Turner – ST10296341
PROG8411
Part 2

02 June 2023

Table of Contents

Table of Figures

Introduction

This analysis aims to predict and differentiate between Covid, Allergy, Flu and Cold symptoms. This will be done through K-Nearest Neighbors Classifier and Naïve Bayes Classifier and the accuracies of these models will be determined. The dataset used is called the Covid classifier using Machine Learning, and it is based off the Mayo Clinic site comparing the symptoms of Covid, Flu, Allergy and Cold (Mayo Clinic Staff, 2023).

This dataset includes 20 different symptoms and shows what category it usually belongs to (Covid, Flu, Allergy, Cold). These symptoms range from muscle aches to vomiting and diarrhea.

K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a supervised learning algorithm used for classification of data (Jain, 2022). Discrete values are used, and the aim is to determine the nearest neighbor (data point) to a query point. To do this, the algorithm needs to determine the Euclidean distance between all of the data points, to best find the decision boundaries (Jain, 2022). The Manhattan distance can also be used and measures the absolute value between two points (Jain, 2022). The Minkowski and Hamming distance respectively are the generalized form of the two above and is typically used for strings (Jain, 2022). Determining the optimal KNN algorithm, the k-value needs to be balanced. This value defines how many neighbors will be checked in order to determine a specific query point. Lower k-values usually lead to high variance and low bias and vice versa (Jain, 2022).

The application of KNN is endless and can be applied for data pre-processing where there are missing values (IBM, 2023a). KNN in finance is a powerful tool to help assess risk and highlight forecasting. Some advantages are that it is easy to implement and has only a few hyper parameters, such as the k-value and a distance metric (IBM, 2023a). However, there are some disadvantages which include the curse of dimensionality, where high-dimensional data causes an increase in errors. The KNN algorithm is also prone to overfitting, where lower k-values can ‘smooth out’ the prediction values (IBM, 2023a).

Naïve Bayes Algorithm

This algorithm is another supervised machine learning algorithm which is used for classification. This algorithm is based on the Bayes’ Theorem, which allows us to ‘invert’ the conditional properties of the probability of an event happening given another event has occurred (IBM, 2023b). This classifier assumes that the predictors are independent to other features in the model and contribute equally to the outcome (IBM, 2023b). To evaluate how well this algorithm fits the data, a confusion matrix can be plotted (IBM, 2023b). There are three types of Naïve Bayes Classifiers, the GaussianNB, MultinomialNB, and the BernoulliNB. In this case we shall use the BernoulliNB as it is used to classify Boolean variables (IBM, 2023b).

Some advantages of this algorithm include the simplistic nature of its parameter estimates and the easy classification of high-dimensional data (IBM, 2023b). However, there are some disadvantages that include its unrealistic core assumption that all features are independent and may provide incorrect classifications (IBM, 2023b). Some applications include spam filtering and document classification, such as image and text classification (IBM, 2023b).

The Analysis

Libraries

Seaborn

A visualization library based on Matplotlib that provides additional plot types (Waskom, M. 2023).

Matplotlib

A visualization library used to create 2D plots and charts (DataFlair. 2023).

NumPy

A library for numerical computing in Python. It provides tools for working with arrays, matrices, and other structures (DataFlair. 2023).

Pandas

A library for data manipulation and analysis. It provides tools for handling and organizing data in tabular form (DataFlair. 2023).

KNeighborsClassifier

A supervised learning algorithm that is used for classification and regression (IBM, 2023b).

SkLearn.train_test_split

Used to estimate the performance of a machine learning algorithm. This is part of the Scikit-learn library and allows one to split the data into a train and a test group. The train set is used to fit the machine learning model to the date, whereas the test set is used to evaluate that fit (Brownlee, 2020)

Confusion Matrix

This is a table used to classify models and show errors that the model has made. There are four categories, true positive (correctly identified correct prediction value), false positive (doesn't classify a correct prediction value), true negative (classifies a false prediction value as true), false negative (correctly identified a false value) (W3schools, 2023).

CategoricalNB

This is a type of Naïve Bayes algorithm and is part of the Sklearn library (Eurostat, 2014). It is used to classify categorical variables (Runebook, 2023). Its main parameter is alphafloat, a smoothing parameter used for smoothing a trend through weighted averages of the observations (Eurostat, 2014).

Accuracy score

This function is from the Sklearn library and calculates the accuracy for a set of predicted labels against a set of true labels (Ashar, 2023). The parameters include y_true, being the correct labels, and y_pred, being the predicted labels returned by the classifier (Scikit Learn, 2023c).

Classification report

This function builds a report displaying the main classification metrics, such as f1 score and precision (Scikit Learn, 2023d). This precision score should be as high as possible, as it shows how often the True predicted value is actually True (Chouinard, 2022). The f1 score measures the precision and recall at the same time, finding the harmonic mean. Recall is another metric and is the sensitivity or fraction of correctly identified positive predictions (Chouinard, 2022).

Random forest classifier

This is an estimator that suits a number of decision trees on many sub-samples of the dataset. This is used to control over-fitting (Scikit Learn, 2023b). The parameters include n_estimators, max_depth, and max_leaf_nodes, among others (Scikit Learn, 2023b).

GridSearchCV

This is used to find the optimal parameters for a model based on a set. It is also known as a cross-validation technique and calculates the score for each combination of parameters, determining the best one (Great Learning Team, 2023).

KMeans

This is a clustering technique that is able to group data based off similar features. Some parameters include n_clusters, verbose and random_state, among others. This function is from the Sklearn library and is one of the fastest (Scikit Learn, 2023a).

Pickle

This serializes or de-serializes a Python object, such as a model (Python, 2023). One is able to save their model, with it having been trained, to then reload it back into the notebook to use for other data (Python, 2023).

Data Analysis

Pre-Processing

Through the use of Jupyter Notebook, this analysis will move through the different stages from pre-processing to evaluating the models' performances. The COVID classifier using Machine Learning dataset is imported using pandas. To evaluate and process the data, we first need to understand the dtype of each feature. As all the features are in integer format except for TYPE, which is in object format, we can move on to determining how large each feature section is. This is to ensure that the dataset is balanced. Null values need to be removed or replaced, therefore, .isna is used to determine if there are any null values in each column.

Exploratory Analysis

To better understand what the data looks like, count-plots are used to show the enumeration of each feature/symptom and what it relates to. 1 indicates that this feature is present, while a 0 indicates that it is not present. For example, the first plot seen below shows the number of cough features seen for each Type (Allergy, Cold, Covid, Flu). We can see that when a cough is present, this usually indicates an allergy or the flu. The final plot shows the amount of data relating to each Type. We can see that there are very few data points that relate to Cold and Covid, making this dataset unbalanced. This will be dealt with in the next section (under sampling).

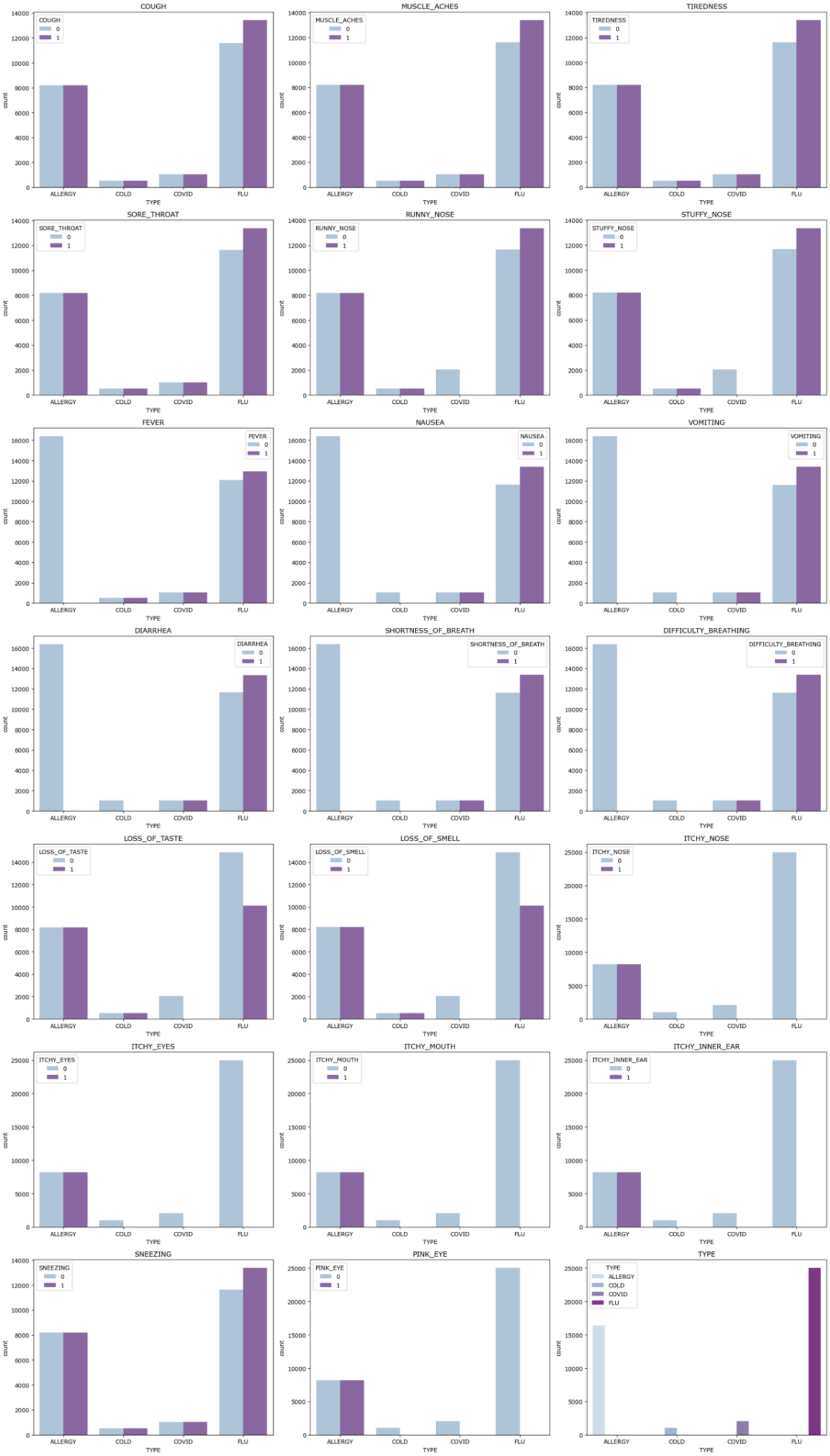
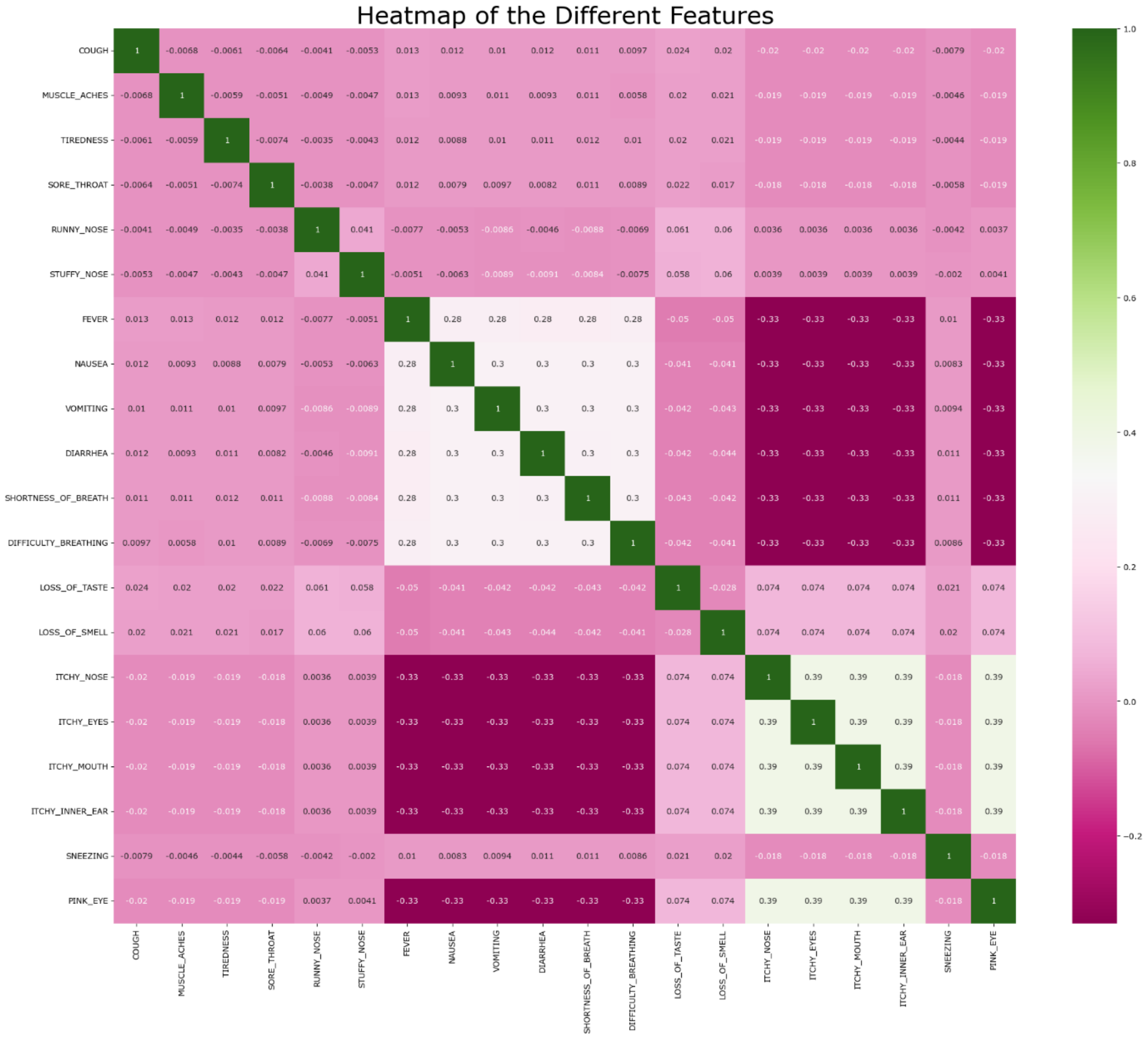


Figure 1: Plots of each feature and what TYPE they belong to.

This is not able to give us the best understanding of how the different features/symptoms interact with one another, therefore a heat map was constructed. A value of one indicates a perfect, positive correlation, and a value of minus one indicates a perfect, negative correlation (Yi, 2021). Values closer to zero indicate very little correlation between the two variables. In the figure below, the correlations seen are quite low (Yi, 2021). However, there are features that are definitely more correlated than others, such as Itchy_Nose, Itchy_Eyes, Itchy_Mouth, Itchy_Inner_Ear, and Pink_Eye. This can easily be seen by the change in colours, indicating possible patterns (Yi, 2021). Fever, nausea, vomiting, diarrhea, shortness of breath and difficulty breathing are also all more or less correlated at a score of 0.3. Overall, these scores are very low and indicate low correlations.



	COUGH	MUSCLE_ACHES	TIREDNESS	SORE_THROAT	RUNNY_NOSE	STUFFY_NOSE	FEVER	NAUSEA	VOMITING	DIARRHEA	SHORTNESS_OF_BREATH	DIFFICULTY_BREATHING	LOSS_OF_TASTE	LOSS_OF_SMELL	ITCHY_NOSE	ITCHY_EYES	ITCHY_MOUTH	ITCHY_INNER_EAR	SNEEZING	PINK_EYE
COUGH	1	-0.0068	-0.0061	-0.0064	-0.0041	-0.0053	0.013	0.012	0.01	0.012	0.011	0.0097	0.024	0.02	-0.02	-0.02	-0.02	-0.02	-0.0079	-0.02
MUSCLE_ACHES	-0.0068	1	-0.0059	-0.0051	-0.0049	-0.0047	0.013	0.0093	0.011	0.0093	0.011	0.0058	0.02	0.021	-0.019	-0.019	-0.019	-0.019	-0.0046	-0.019
TIREDNESS	-0.0061	-0.0059	1	-0.0074	-0.0035	-0.0043	0.012	0.0088	0.01	0.011	0.012	0.01	0.02	0.021	-0.019	-0.019	-0.019	-0.019	-0.0044	-0.019
SORE_THROAT	-0.0064	-0.0051	-0.0074	1	-0.0038	-0.0047	0.012	0.0079	0.0097	0.0082	0.011	0.0089	0.022	0.017	-0.018	-0.018	-0.018	-0.018	-0.0058	-0.019
RUNNY_NOSE	-0.0041	-0.0049	-0.0035	-0.0038	1	0.041	-0.0077	-0.0053	-0.0086	-0.0046	-0.0088	-0.0069	0.061	0.06	0.0036	0.0036	0.0036	0.0036	-0.0042	0.0037
STUFFY_NOSE	-0.0053	-0.0047	-0.0043	-0.0047	0.041	1	-0.0051	-0.0063	-0.0089	-0.0091	-0.0084	-0.0075	0.058	0.06	0.0039	0.0039	0.0039	0.0039	-0.002	0.0041
FEVER	0.013	0.013	0.012	0.012	-0.0077	-0.0051	1	0.28	0.28	0.28	0.28	0.28	-0.05	-0.05	-0.33	-0.33	-0.33	-0.33	0.01	-0.33
NAUSEA	0.012	0.0093	0.0088	0.0079	-0.0053	-0.0063	0.28	1	0.3	0.3	0.3	0.3	-0.041	-0.041	-0.33	-0.33	-0.33	-0.33	0.0083	-0.33
VOMITING	0.01	0.011	0.01	0.0097	-0.0086	-0.0089	0.28	0.3	1	0.3	0.3	0.3	-0.042	-0.043	-0.33	-0.33	-0.33	-0.33	0.0094	-0.33
DIARRHEA	0.012	0.0093	0.011	0.0082	-0.0046	-0.0091	0.28	0.3	0.3	1	0.3	0.3	-0.042	-0.044	-0.33	-0.33	-0.33	-0.33	0.011	-0.33
SHORTNESS_OF_BREATH	0.011	0.011	0.012	0.011	-0.0088	-0.0084	0.28	0.3	0.3	0.3	1	0.3	-0.043	-0.042	-0.33	-0.33	-0.33	-0.33	0.011	-0.33
DIFFICULTY_BREATHING	0.0097	0.0058	0.01	0.0089	-0.0069	-0.0075	0.28	0.3	0.3	0.3	0.3	1	-0.042	-0.041	-0.33	-0.33	-0.33	-0.33	0.0086	-0.33
LOSS_OF_TASTE	0.024	0.02	0.02	0.022	0.061	0.058	-0.05	-0.041	-0.042	-0.042	-0.043	-0.042	1	-0.028	0.074	0.074	0.074	0.074	0.021	0.074
LOSS_OF_SMELL	0.02	0.021	0.021	0.017	0.06	0.06	-0.05	-0.041	-0.043	-0.044	-0.042	-0.041	-0.028	1	0.074	0.074	0.074	0.074	0.02	0.074
ITCHY_NOSE	-0.02	-0.019	-0.019	-0.018	0.0036	0.0039	-0.33	-0.33	-0.33	-0.33	-0.33	-0.33	0.074	0.074	1	0.39	0.39	0.39	-0.018	0.39
ITCHY_EYES	-0.02	-0.019	-0.019	-0.018	0.0036	0.0039	-0.33	-0.33	-0.33	-0.33	-0.33	-0.33	0.074	0.074	0.39	1	0.39	0.39	-0.018	0.39
ITCHY_MOUTH	-0.02	-0.019	-0.019	-0.018	0.0036	0.0039	-0.33	-0.33	-0.33	-0.33	-0.33	-0.33	0.074	0.074	0.39	0.39	1	0.39	-0.018	0.39
ITCHY_INNER_EAR	-0.02	-0.019	-0.019	-0.018	0.0036	0.0039	-0.33	-0.33	-0.33	-0.33	-0.33	-0.33	0.074	0.074	0.39	0.39	0.39	1	-0.018	0.39
SNEEZING	-0.0079	-0.0046	-0.0044	-0.0058	-0.0042	-0.002	0.01	0.0083	0.0094	0.011	0.011	0.0086	0.021	0.02	-0.018	-0.018	-0.018	-0.018	1	-0.018
PINK_EYE	-0.02	-0.019	-0.019	-0.019	0.0037	0.0041	-0.33	-0.33	-0.33	-0.33	-0.33	-0.33	0.074	0.074	0.39	0.39	0.39	0.39	-0.018	1

Figure 2: A heatmap of the correlations between each feature.

Figure 3 is a cluster map, and indicates a hierarchal clustered heat map, also known as a dendrogram heat map (Bock, 2023).. This measures the hierarchal structure of the features to best show clusters (Bock, 2023). Small links or structures indicate a higher level of similarity between features. We can see below that sore throat, cough, muscle aches, tiredness and sneezing are more correlated/linked to fever, diarrhea, nausea, difficulty breathing, vomiting and shortness of breath than to being itchy.

This is because the branches link to one tree/cluster, whereas less similar features are usually found in different trees/clusters.

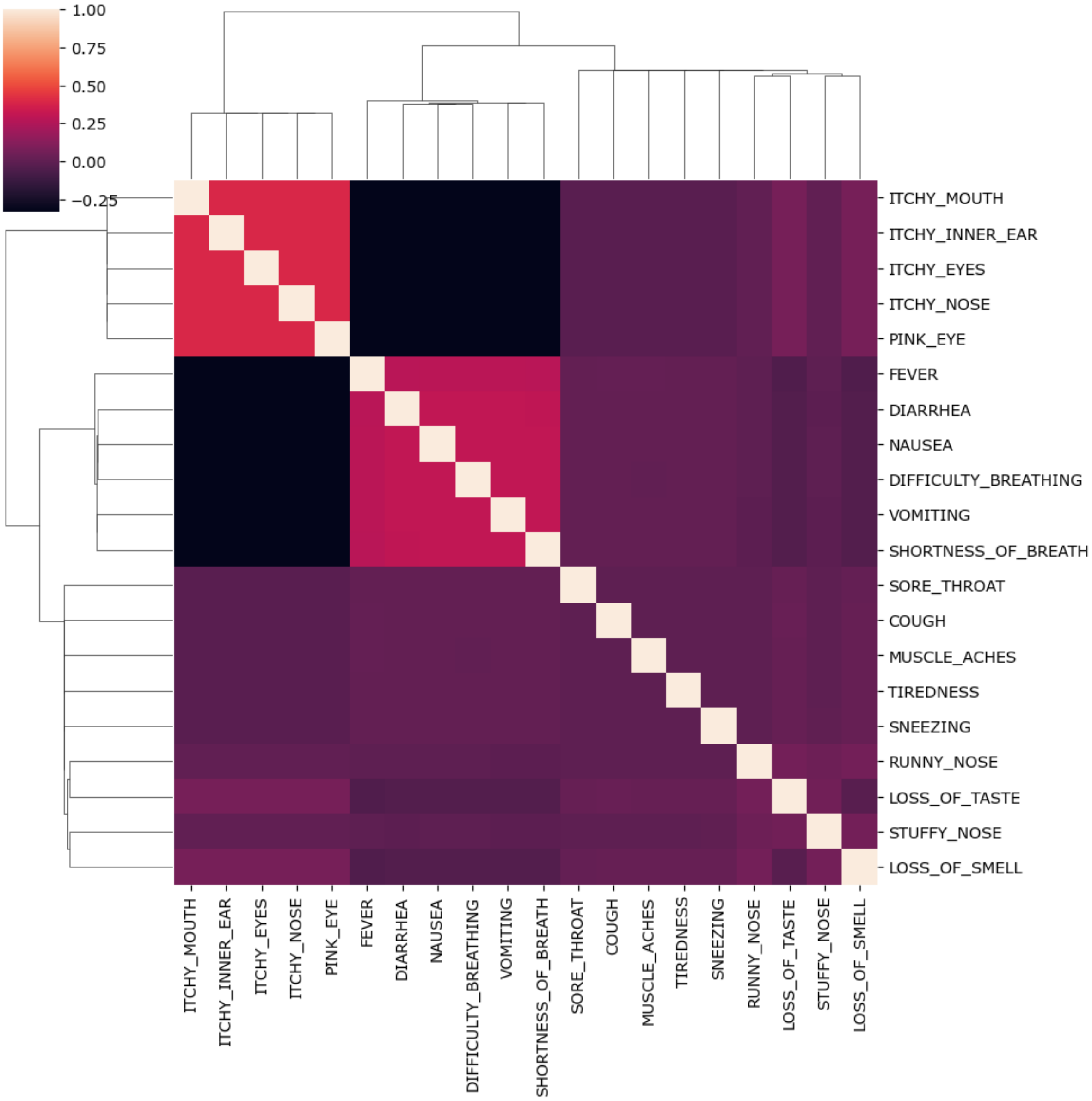


Figure 3: Hierarchal clustered heat map of each feature.

Train and Test

To train and test the different models, the dataset must be split into a training and a testing set. The 70% train and 30% test split has been implemented. To properly define the X and y data, iloc is used. For X, all of the rows and columns are chosen except for the very last column and for y, all rows are chosen and only the last column (TYPE). This separates the independent from the dependent variables.

K-Nearest Neighbors Hyper-Parameter Tuning

The first model to test on the dataset is the K-Nearest Neighbors algorithm. First, hyper-parameter tuning needs to be done to determine the best parameters for the model. To start, we define an array from 1 to 99, skipping all the even numbers, and pass this to the GridSearchCV as n_neighbors. These are then looped through, trained and tested for each iteration of neighbors until 99. The n_neighbor with the best test accuracy score is then automatically used as the parameter to train the dataset. In this case the best parameter was n_neighbors = 43.

To properly understand why n_neighbors of 43 was chosen, a classification report is generated. This report consists of the precision score, the percentage of correct, positive predictions compared to the total positive predictions. The recall score is the percentage of correct, positive predictions compared to the total actual positives. F1 scores is the weighted harmonic mean of the precision and recall scores (Zach, 2022). If it is closer to one, the better the model is. The accuracy is then measured as how well the model knows the data to perform predictions on it (Zach, 2022).

In this case, the classification report for the predicted scores based on the train dataset has a high score. The F1 scores are all close to one, except for Cold and Covid, this could be due to under sampling which will be corrected in the next section. The accuracy, however, is very high at 93%. The classification report for the test set also has an accuracy of 93% yet is has extremely low F1 scores for Cold and Covid.

Cold and Covid are Under-sampled

As seen above, Cold and Covid had very low F1 scores, this needs to be investigated. We determine the number of samples for each TYPE and can clearly see that there are only 2048 and 1024 samples for Covid and Cold respectively. This is very low compared to 25000 and 16381 samples for Flu and Allergy respectively. To visualize this better, a pie chart is constructed. Cold and Covid together only make up 6.91% of all the samples. Random under sampling is used to balance the majority and minority classes. This is where some data points, at random, are deleted in the majority classes. In this case data from Flu was deleted and data was added to Allergy, Cold and Covid.

With this more comprehensive dataset, the GridSearchCV needs to be constructed again for KNN.

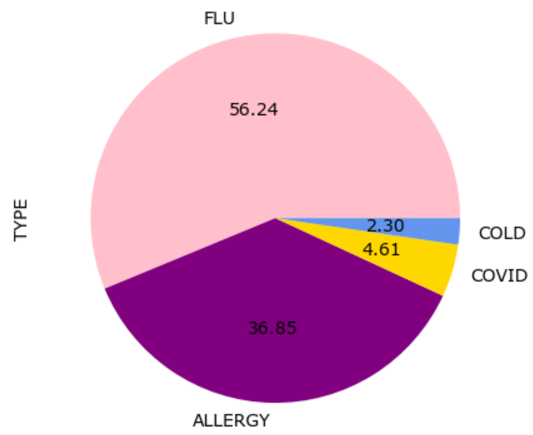


Figure 4: Pie chart of the number of samples of each Type - before under sampling

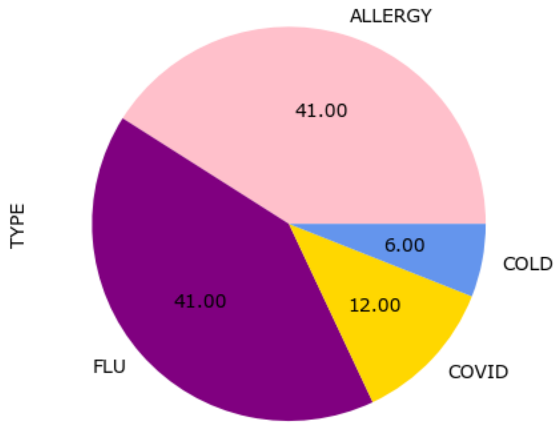


Figure 5: Pie chart of the number of samples of each Type - after under sampling

K-Nearest Neighbors after Under-sampling

The GridSearchCV is performed again and fitted to the X_train and y_train data. We can see that the parameter of n_neighbors = 45 was found to be the best now that the dataset has been balanced. The trained model is then saved as model1 through Pickle.

Testing Model1

To test model1, the model is reloaded back into the notebook and is given the X_test set to predict the y-values. The accuracy score is calculated for y_pred as 90.37%. To further understand how well the model was fitted, a classification report and confusion matrix is outputted. For y_pred_knn_train, model1 predicts the X_train set and gives an accuracy score of 94%. The confusion matrix shows true positive, true negative, false positive and false negative values. The middle diagonal blocks show true positive values, meaning these values were correctly predicted, there are 11 226. The false negative values would add up to be 59, while the false positive values would be 120. Finally, there would be 6982 true negatives. One can see that the majority of the predicted values were predicted correctly. The same process goes for the KNN Test Matrix, where there are a total of 4815 values are correctly predicted.

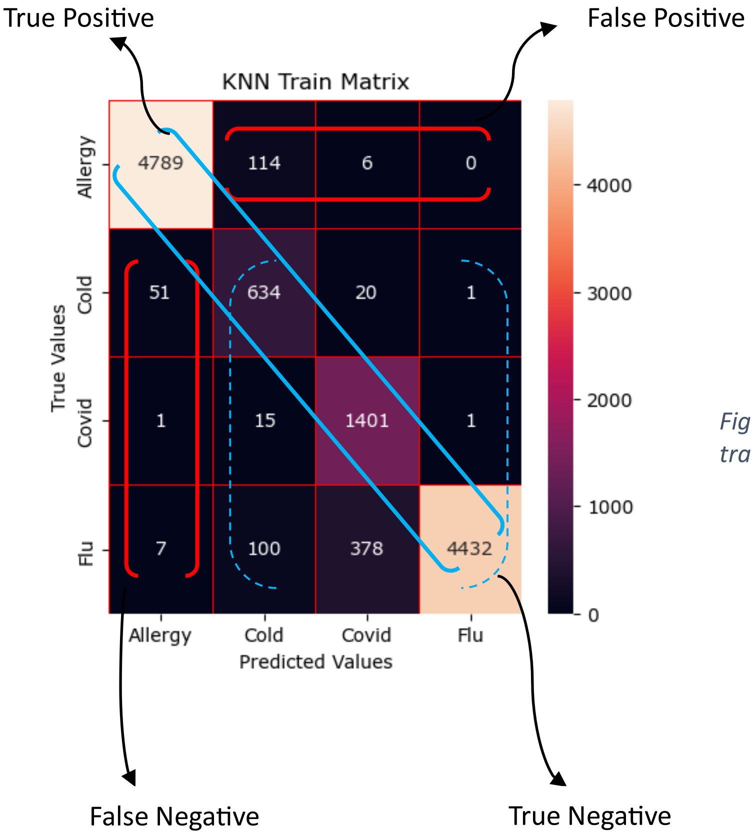


Figure 6: Confusion matrix of the KNN model for the training set.

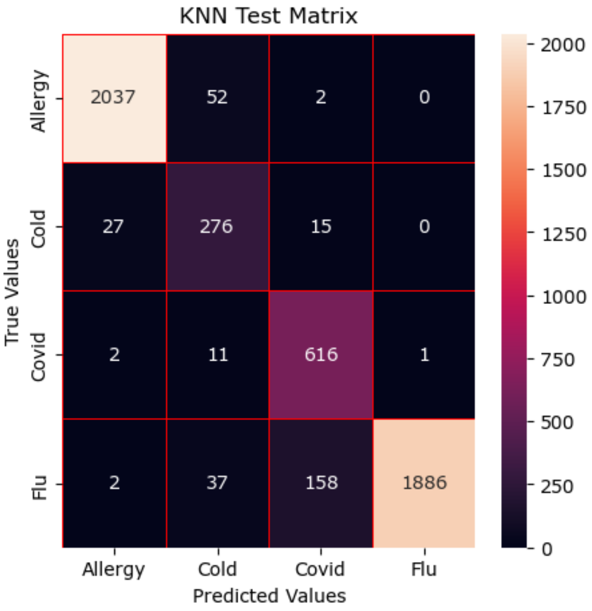


Figure 7: Confusion matrix of the KNN model for the testing set.

Naïve Bayes Hyper-Parameter Tuning

This is the second model applied to the dataset. To get the most accurate predictions, the model must be tuned for the best hyper-parameters. In this case, the best alpha values have to be compared. An array is made of alpha values, from 0.1 to 1.0 (skipping numbers by 0.1 each time). This is then fed into the GridSearchCV to find the best CategoricalNB alpha parameter. Five folds of the training set were looped through, and an alpha value of 0.1 was found to give the highest test accuracy score. This trained model was then saved as model2.

Testing Model2

To accurately test model2, the model was reloaded and used on the test dataset. A classification report was generated on model2's predicted values on the training set. A high accuracy score was seen at 95% and the F1 scores were all above 83%. A confusion matrix was then visualized, and it was found that there were 11 298 true positive values and 7041 true negative values. For the test set, the pickled model2 was used to predict, and the accuracy score, from the classification report, was found to be 94%. This is 1% lower than the training set accuracy. A confusion matrix was computed and found 4840 true positives and 3029 true negatives.
https://www.v7labs.com/blog/confusion-matrix-guide

Comparing Models

To compare both the KNN and Naïve Bayes algorithms, we need to look at the grid search KNN and grid search NB accuracy scores. The KNN score is 93.58%, while the CategoricalNB score is 94.48%. These two scores are very similar, yet the CategoricalNB model definitely has the higher accuracy. Moreover, the number of true positives for the trained KNN model was 11 226 compared to 11 298 for the trained Naïve Bayes algorithm. For the test set, there were 4815 KNN true positives and 4840 Naïve Bayes true positives. These are very small differences, however, there were more false positives for the Naïve Bayes algorithm than for the KNN model. Therefore, through all of the evaluation tests, Naïve Bayes is the more accurate model to use in this case.

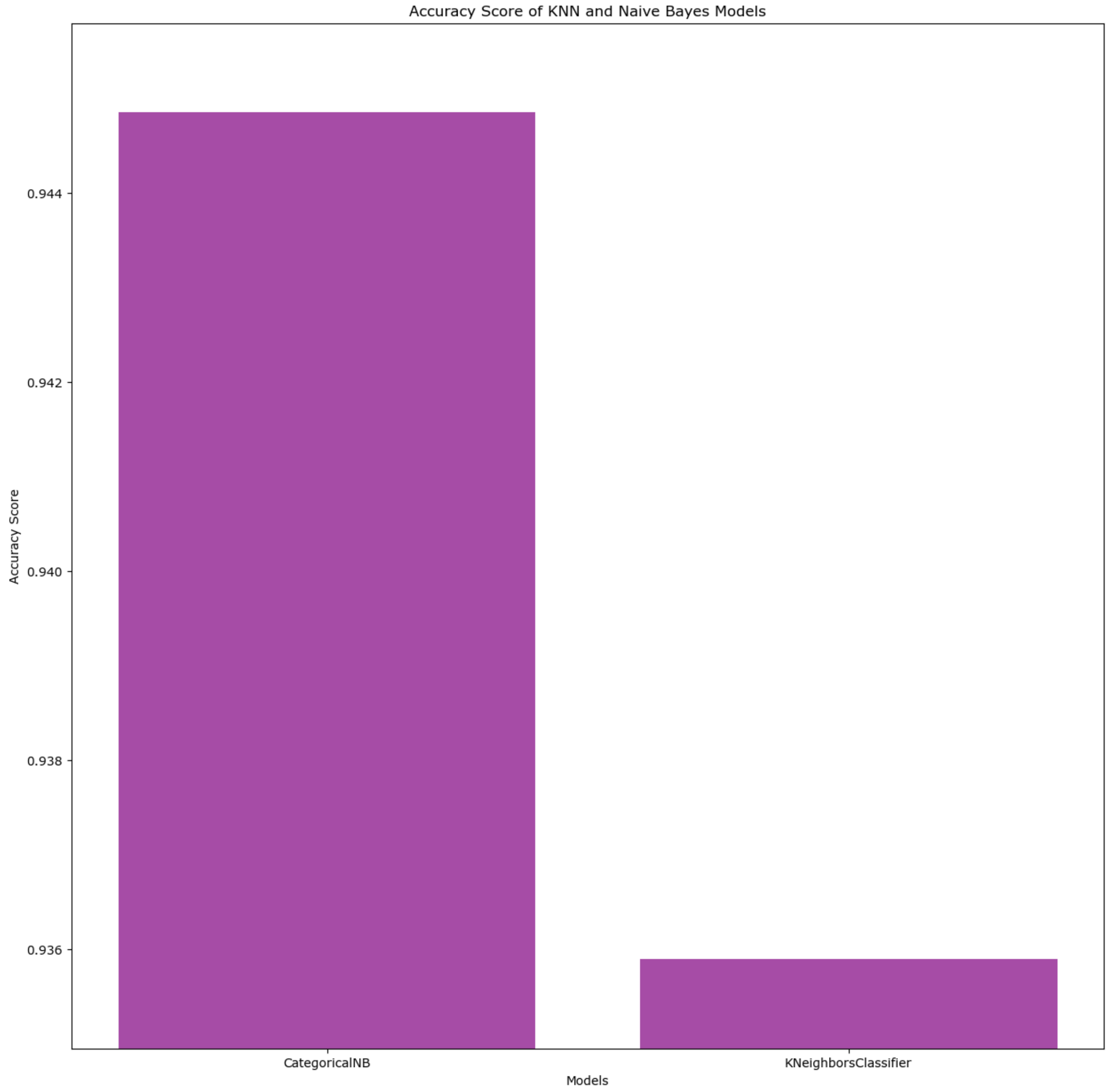


Figure 8: Comparing the accuracy score for the KNN model and Naive Bayes model.

References

Ashar, T. 2023. What is the accuracy_score function in Sklearn? Educative. [Online]. Available at: https://www.educative.io/answers/what-is-the-accuracyscore-function-in-sklearn [Accessed on 1 June 2023].

Bock, T. 2023. What is a Dendrogram? DisplayR. [Blog]. Available at: https://www.displayr.com/what-is-dendrogram/#:~:text=A%20dendrogram%20is%20a%20diagram,to%20allocate%20objects%20to%20clusters [Accessed on 1 June 2023].

Brownlee, J. 2020. Train-Test Split for Evaluating Machine Learning Algorithms. Machine Learning Mastery. 24 July 2020. [Blog]. Available at: https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/ [Accessed on 1 June 2023].

Chouinard, J.C. 2022. How to use Classification Report in Scikit-learn (Python). JC Chouinard. 5 May 2022. [Blog]. Available at: https://www.jcchouinard.com/classification-report-in-scikit-learn/ [Accessed on 1 June 2023].

DataFair. 2023. Python Libraries – Python Standard Libraries & List of Important Libraries. [Online]. Available at: https://data-flair.training/blogs/python-libraries/# [Accessed on 21 April 2023].

Eurostat. 2014. Glossary:Smoothing. Eurostat. 23 December 2014. [Online] available at: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Glossary:Smoothing#:~:text=Smoothing%20refers%20to%20estimating%20a,to%20partially%20offset%20each%20other [Accessed on 1 June 2023].

Great Learning Team. 2023. Hyperparameter Tuning with GridSearchCV. GreatLearning. 30 May 2023. [Online]. Available at: https://www.mygreatlearning.com/blog/gridsearchcv/#:~:text=GridSearchCV%20is%20a%20technique%20for,parameter%20values%2C%20predictions%20are%20made [Accessed on 1 June 2023].

Mayo Clinic Staff. 2023. COVID-19, cold, allergies and the flu: What are the differences? Mayo Clinic. [Online]. Available at: https://www.mayoclinic.org/diseases-conditions/coronavirus/in-depth/covid-19-cold-flu-and-allergies-differences/art-20503981 [Accessed on 30 May 2023].

IBM. 2023a. Learn how Naïve Bayes classifiers uses principles of probability to perform classification tasks. IBM. [Online]. Available at: https://www.ibm.com/topics/naive-bayes#:~:text=The%20Naïve%20Bayes%20classifier%20is,a%20given%20class%20or%20category [Accessed on 30 May 2023].

IBM. 2023b. What is the k-nearest neighbors algorithm? IBM. [Online]. Available at: https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point [Accessed on 29 May 2023].

Jain, V. 2022. Introduction to KNN Algorithms. Analytics Vidya. 31 January 2022. [Blog]. Available at: https://www.analyticsvidhya.com/blog/2022/01/introduction-to-knn-algorithms/ [Accessed on 1 June 2023].

Python. 2023. Pickle-Python object serialization. Python. [Online]. Available at: https://docs.python.org/3/library/pickle.html#:~:text="Pickling"%20is%20the%20process%20whereby,back%20into%20an%20object%20hierarchy [Accessed on 1 June 2023].

Runebook. 2023. Sklearn.naive_bayes.CategoricalNB. [Online]. Available at: https://runebook.dev/en/docs/scikit_learn/modules/generated/sklearn.naive_bayes.categoricalnb [Accessed on 30 May 2023].

Scikit Learn. 2023a. Sklearn.cluster.KMeans. Scikit Learn. [Online]. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html [Accessed on 1 June 2023].

Scikit Learn. 2023b. Sklearn.ensemble.RandomForestClassifier. Scikit Learn. [Online]. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html [Accessed on 1 June 2023].

Scikit Learn. 2023c. Sklearn.metrics.accuracy_score. Scikit Learn. [Online]. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html [Accessed on 1 June 2023].

Scikit Learn. 2023d. Sklearn.metrics.classification_report. Scikit Learn. [Online]. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html [Accessed on 1 June 2023].

W3schools. 2023. Machine Learning – Confusion Matrix. W3schools. [Online]. Available at: https://www.w3schools.com/python/python_ml_confusion_matrix.asp [Accessed on 1 June 2023].

Yi, M. 2021. A Complete Guide to Heatmaps. ChartIO. [Blog]. Available at: https://chartio.com/learn/charts/heatmap-complete-guide/#:~:text=Heatmaps%20are%20used%20to%20show,for%20one%20or%20both%20variables [Accessed on 2 June 2023].

Zach. 2022. How to Interpret the Classification Report in sklearn (with examples). Stratology. 09 May 2022. [Blog]. Available at: https://www.statology.org/sklearn-classification-report/ [Accessed on 2 June 2023].