Machine Learning Assignment - COVID19 vs Influenza

Benjamin Dunn

December 21, 2021

Contents

1 Introduction

For this assignment I was tasked in evaluating the effectiveness of different machine learning classifiers for patient data in which patients either had COVID-19 or influenza. I have chosen to cover five classifiers within WEKA to achieve this goal. These classifiers are as follows:

- A bayesian method - Naive Bayes
- A k-Nearest Neighbors method - IBk
- A decision tree method - J48
- A support vector machine method - LibSVM
- A neural network method - Multilayer Perceptron

1.1 A Note on Kappa Statistic

The kappa statistic is usually used to evaluate how well a classifier is performing on a piece of data. According to Cohen's original paper[1], kappa values less than 0 indicate no agreement, 0.01 to 0.20 as none to slight agreement, 0.21 to 0.40 as fair agreement, 0.41 to 0.60 as moderate agreement, 0.61 to 0.80 as substantial agreement, and 0.81 to 1.00 as almost perfect agreement.

However while this may be generally applicable, for medical research we have to raise our tolerances for what is acceptable since small classification errors could have large real world impacts. This is made clear in Interrater reliability: The kappa statistic[2] "For a clinical laboratory, having 40% of the sample evaluations being wrong would be an extremely serious quality problem."

For this reason we will be more stringent with kappa statistic results in this assignment, following the outline in Interrater reliability: The kappa statistic[2] where kappa values from 0.01 to 0.20 indicate no agreement, 0.21 to 0.39 as minimal agreement, 0.40 to 0.59 as weak agreement, 0.60 to 0.79 as moderate agreement, and 0.80 to 0.90 as strong agreement, and anything above 0.90 as almost perfect agreement.

1.2 A note on auROC

Area under the ROC curve is considered an effective way to evaluate overall accuracy of a classifier. According to Receiver Operating Characteristic Curve in Diagnostic Test Assessment[3], "...an AUC of 0.5 suggests no discrimination (i.e., ability to diagnose patients with and without the disease or condition based on the test), 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered excellent, and more than 0.9 is considered outstanding." So anything close to 0.5 would suggest the classifier has a poor ability to discern between patients with COVID-19/Influenza. The closer it gets to 1.0 the better it is.

2 Task 1 - Investigation of Classifier Performance

2.1 Naive Bayes

Naive Bayes performs reasonably well with a total correctly classified instances of 77%. More importantly it correctly classifies 701 of 742 COVID-19 positive tests. While it does incorrectly evaluate 299 cases of flu as COVID-19, the potential severe health impacts of COVID-19 is greater then that of the influenza virus and correctly assessing COVID-19 true positives is more important then rates of false positives. Obviously having high TP rates for both COVID-19 and the flu is important for efficiently allocating

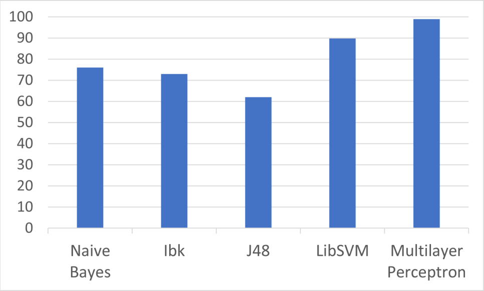


Figure 1: Percent correct classifiers

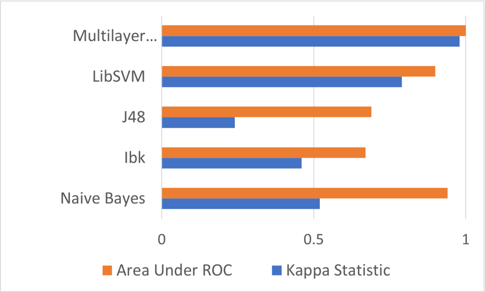


Figure 2: ROC and Kappa statistics

medical resources, these results are potentially very useful for simply catching the vast majority of COVID cases.

For the given dataset Naive Bayes returns a weak kappa statistic of 0.54. Though again depending on what we need to consider this may be disregarded if our goal is to identify as many positive COVID cases as possible.

The ROC scores highly but its important to note that this is due to the high rates of TPs for COVID since it is the positive class in our data and high rate of FNs for influenza. So in this case ROC doesn't necessarily speak to the quality of the classifier.

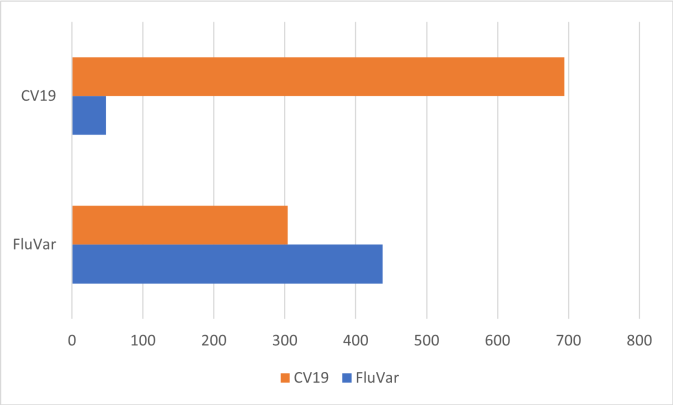


Figure 3: Confusion Matrix Naive Bayes

2.2 IBk

IBk performs well with a total correctly classified instances of 74%. Like Naive Bayes it manages a very high rate of true positive COVID tests, correctly classifying all 742 cases. Again like Naive Bayes results, a low true positive rate for correctly classifying influenza could be forgiven depending on the goal.

For the given dataset IBk returns a weak kappa statistic of 0.49. Again, depending on what we need to consider this may be disregarded if our goal is to identify as many positive COVID cases as possible.

The ROC for IBk fails to achieve an acceptable score of 0.7 and suggests that the classifier isn't particularly trustworthy in accurately identifying between COVID and Influenza.

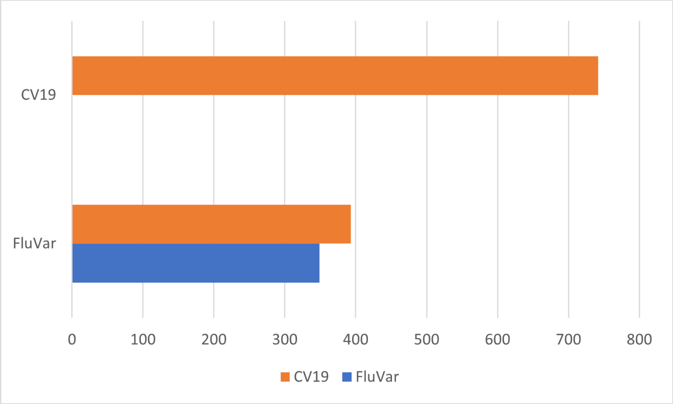


Figure 4: Confusion Matrix IBk

2.3 J48

J48 came in with the lowest rate of correctly classified instances only achieving 63%. It performs very poorly when it comes to correctly identifying COVID cases at 57% of the time.

For the given dataset J48 returns a minimal kappa statistic of 0.26 giving us a strong indicator that we do not want to be using J48.

While J48 does manage to score a ROC of 0.7 not being able to reliably classify COVID cases is still a real issue, and so can be disregarded.

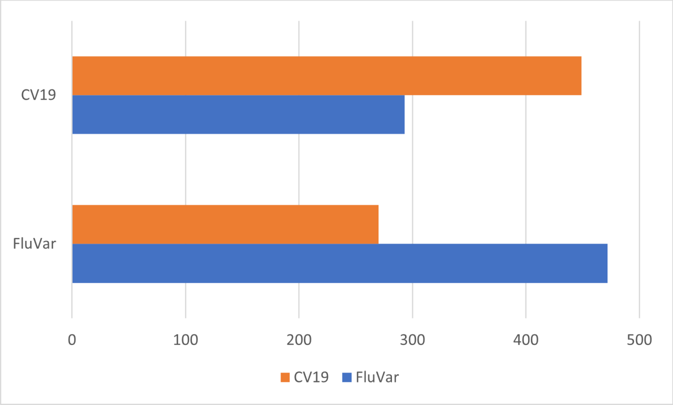


Figure 5: Confusion Matrix J48

2.4 LibSVM

LibSVM achieves a high rate of correct classification at 89%. It both achieves an almost perfect classification of influenza at 99%, and a high rate of COVID classification at 79%. If the goal was to efficiently use resources to treat COVID then based on these results LibSVM wouldn't be a bad choice.

For the given dataset LibSVM returns a moderate, almost strong kappa statistic of 0.79. This is the strongest kappa statistic so far and matches strongly with the high rates of classification for both COVID and influenza.

Further to this the ROC score of 0.89 further suggests this method is accurate and trustworthy.

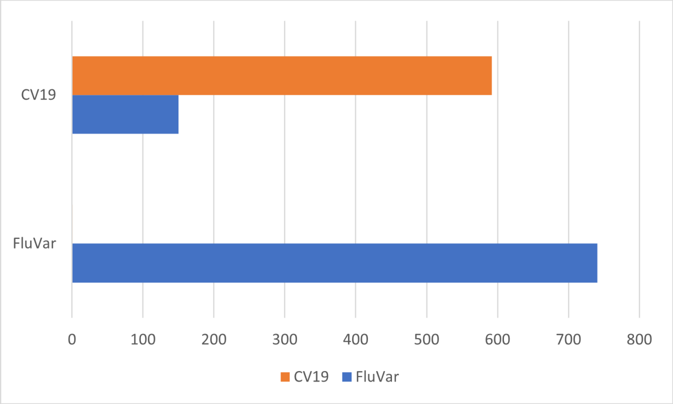


Figure 6: Confusion Matrix LibSVM

2.5 Multilayer Perceptron

The use of a multilayer perceptron neural network achieved an almost perfect classification rate across both classes. The main negative with this method is the long learning time, depending on the scenario this could be an issue.

For the given dataset Multilayer Perceptron returns an almost perfect kappa statistic of 0.97.

The ROC score of 1.0 further suggests this is a reliable and accurate way to discern between the two classes.

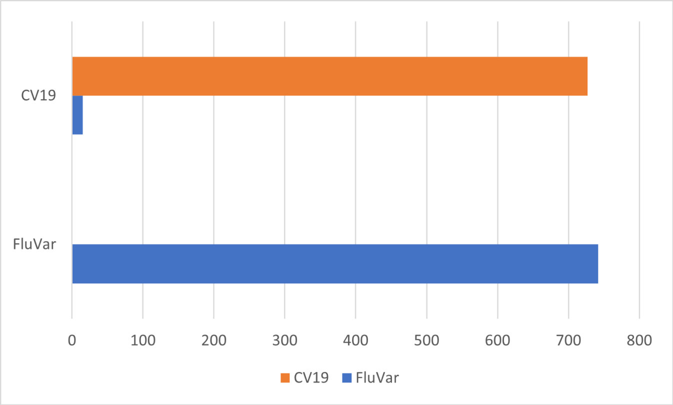


Figure 7: Confusion Matrix Multilayer Perceptron

2.6 Comparing Classifiers

To compare the results further I made use of the experimenter WEKA tool and further sought to validate the learners with a ten fold cross validation - a standard way of further making efficient use of data, using data instances in both testing and training.

Based on the results, the the multilayer perceptron classifier out performs all other classifiers in every category for which we are concerned. Notably however it takes considerably more time to produce results compared to other classifiers. With almost perfect classification results, and almost perfect kappa and ROC static results its hard to argue against the reliability of this learner.

The next most reliable classifier would be the SVM scoring highly in all categories, slightly behind the neural net learner. While not as accurate, it provides results considerably faster and does

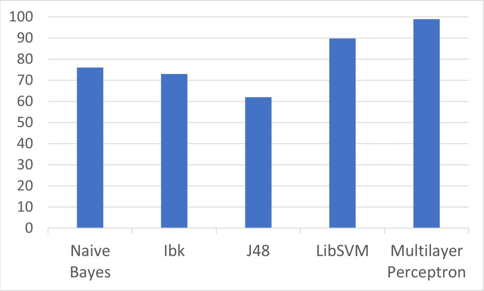


Figure 8: Percent correct classifiers

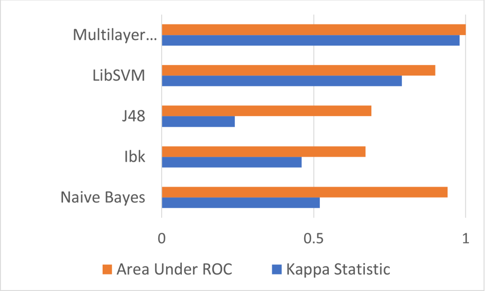


Figure 9: ROC and Kappa statistics

so with little loss in reliability. It should be noted that SVM does fall behind Naive Bayes and IBk for correctly classifying covid cases. This is counterbalanced by its ability to perfectly classify influenza cases which could be useful when being mindful on how to allocate resources.

The remaining three learners have mediocre to poor reliability. J48 performs particularly poorly compared to the other classifiers when looking at how well they can determine positive covid results, arguably the most important metric. Naive Bayes and IBk perform similarly both sharing the tendency to over classify covid cases. If your only goal was to correctly identify as many covid cases as possible this could be acceptable.

3 Task 2 - Tackling the Issue of Missing Data

3.1 Overview of Missing Data

The given dataset has significant amounts of missing data. Only 3 out of the 46 attributes in the dataset (sex, coughing, fever) don't have more then 50% of their values missing. Therefore it is reasonable to assume that such a large incomplete dataset is having an affect on our classifiers ability to classify.

We could remove data instances in which data is missing. While a typically common practice, it is likely to have negative affects on our ability to reliably classify our data as the vast majority of data instances would be removed - at thee very least a filter to remove duplicate instances will be used to remove unnecessary instances. Alternatively we could replace the missing data. Using the WEKA filter *ReplaceMissingValues* we can replace missing values with the mean value for each attribute. Doing so would provide some learners such as IBk and J48 with large improvements on their classification accuracy, since they generally suffer from datasets with large amounts of missing data.

It is important to note that the method of replacing data with a mean can be subject to large amounts of bias if the only data you have exist as outliers in a complete dataset[4].

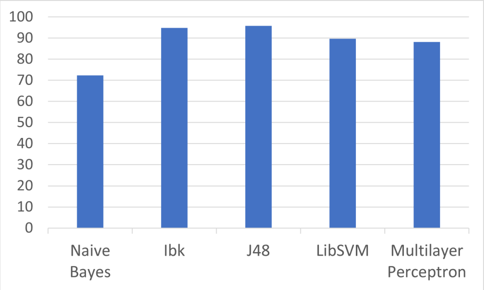


Figure 10: Percent correct classifiers after use of *ReplaceMissingValues*

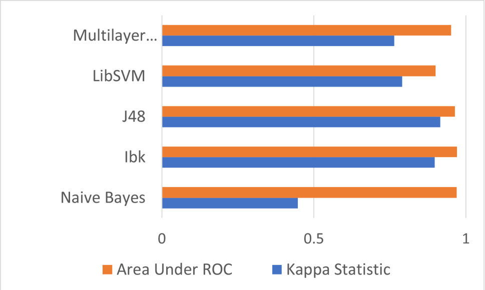


Figure 11: ROC and Kappa statistics after use of *ReplaceMissingValues*

3.2 Analysing Naive Bayes with Missing Data Replaced

Naive Bayes stands out as its Kappa, ROC and percent correctly classified has remained unchanged. This is because the classifier takes into account missing data before any filters are applied. To improve the classifiers performance other methods would need to be applied. Most notably, simply applying the mean across missing values has caused the classifier to over classify flu variants and now fails to correctly classify more then half of covid cases. This method of dealing with the missing data is poorly suited for helping improve the classifier.

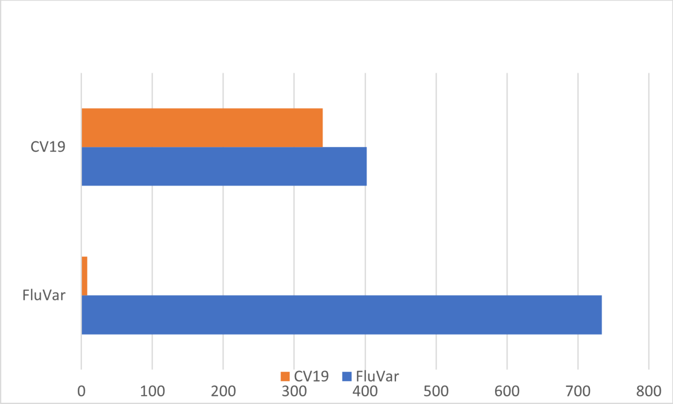


Figure 12: Confusion Matrix Naive Bayes after *ReplaceMissingValues*

3.3 Analysing IBk with Missing Data Replaced

IBk improves considerably with the data replacements. For the given dataset IBk returns a strong kappa statistic of 0.89.

The ROC for IBk excels with a score of 0.97 suggesting that the classifier is now particularly trustworthy in accurately identifying between COVID and Influenza cases.

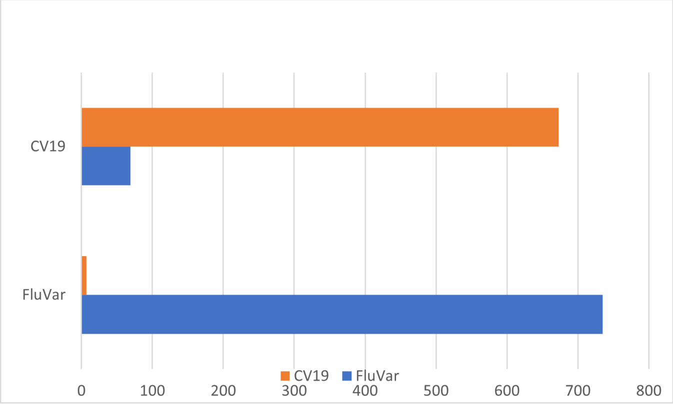


Figure 13: Confusion Matrix IBk after *ReplaceMissingValues*

3.4 Analysing J48 with Missing Data Replaced

J48 also makes significant improvements once data is replaced. It performs exceedingly well when it comes to correctly identifying COVID cases at over 91% of the time.

For the given dataset J48 returns a kappa statistic of 0.91 giving us a strong indicator that we now want to be using J48.

Furthermore J48 does manage to score a ROC of 0.96 further suggesting the accuracy and reliability of the classifier. When compared to Multilayer Perceptron pre *ReplaceMissingValues* filtering, J48 is now a close second in terms of reliability and accuracy, but evaluates the data significantly faster.

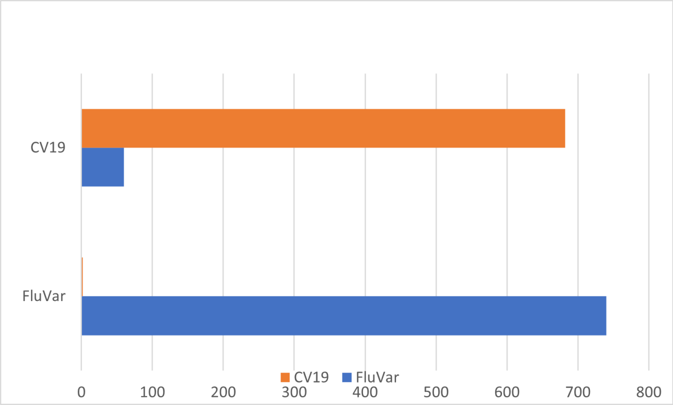


Figure 14: Confusion Matrix J48 after *ReplaceMissingValues*

3.5 Analysing LibSVM with Missing Data Replaced

Like Naive Bayes, SVM's are unaffected in their performance due their ability to deal with missing data. Hence why they are extensively used for missing data handling[4]. To improve the classifiers performance other methods would need to be applied.

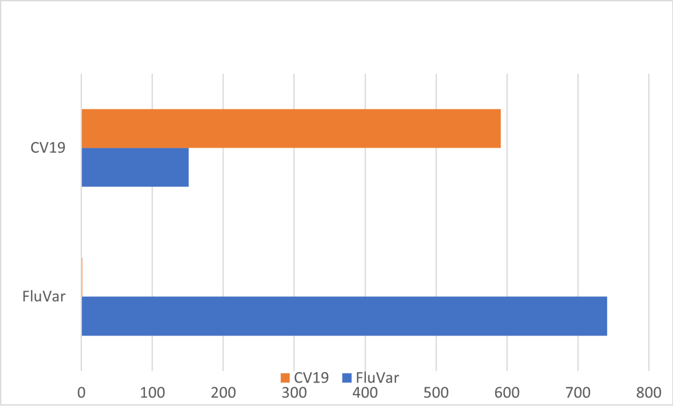


Figure 15: Confusion Matrix LibSVM after *ReplaceMissingValues*

3.6 Analysing Multilayer Perceptron with Missing Data Replaced

The Multiplayer Perceptron classifier is the only classifier to lose performance - this is likely due to biases created by *ReplaceMissingValues*. Since the results pre *ReplaceMissingValues* filtering were almost perfect there is no reason to be using the altered dataset.

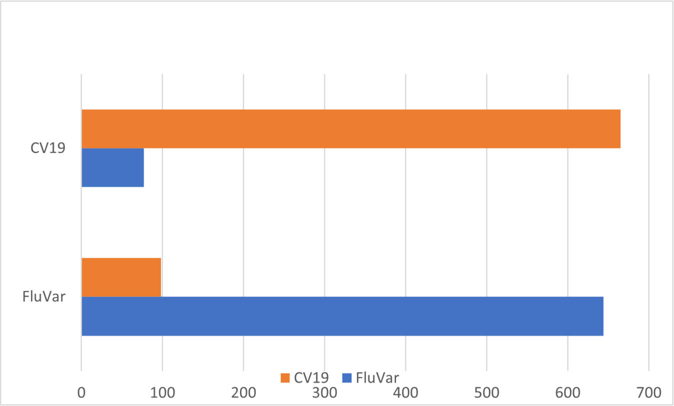


Figure 16: Confusion Matrix Multilayer Perceptron after *ReplaceMissingValues*

4 Task 3 - Feature Selection

Applying some feature selection on our dataset will help reduce the size of the dataset by removing irrelevant attributes that do not contribute in determining the classes of each instance. In some instances this could further improve the performance of our classifiers.

4.1 Evaluating Features

To know which features we will use or discard we need some general guidelines for selecting meaningful features. Features with tiny amounts of data, features that simply repeat information from previous features, and noisy features with large standard deviations are all very likely features with little to no value. Removing these features is likely a good start in any feature selection.

Past the references there are related charts and tables. Blue bar represents the *mean*, orange represents *standard deviation*, grey represents *max*, and yellow represents *min*. Please refer to these when reading this section.

4.1.1 Initial PCR Diagnosis

This feature has a very lopsided distribution between positive and negative. With 95% of the instances that have an initial PCR Diagnosis being positive this feature is likely redundant when it comes to classifying accurately and can be removed.

4.1.2 Neutrophil Categorical

This feature essentially duplicates the feature *Neutrophil*. Hence this one is redundant and can be removed.

4.1.3 Serum Levels Of White Blood Cell Categorical

Just like the other *Categorical* feature this is likely a redundant feature and can be safely removed.

4.1.4 Plateletes

This feature has large maximum outliers and a large standard deviation. This noise is likely causing issues for our classifiers and should be removed.

4.1.5 C Reactive Protein Levels Categorical

Just like the other *Categorical* feature this is likely a redundant feature and can be safely removed.

4.1.6 Eosinophils

This feature is missing in 99% of instances and is unlikely to be useful and will be removed.

4.1.7 Red Blood Cells

This feature is missing in 99% of instances and is unlikely to be useful and will be removed.

4.1.8 Hemoglobin

This feature is missing in 96% of instances and is unlikely to be useful and will be removed.

4.1.9 Procalcionin

This feature is missing in 95% of instances and is unlikely to be useful and will be removed.

4.1.10 Days to Death

This feature is missing in all but 4 instances and in those it is N/A. It is unlikely to be useful and will be removed.

4.1.11 Days In Incubation

This feature is missing in 95% of instances and is unlikely to be useful and will be removed.

4.1.12 Scan Results

This feature is biased strongly toward positive labels with a weighting of 125 compared to the other labels of 10. Of those other labels almost half of them are N/A. This feature will be removed.

4.1.13 X Ray Results

This feature is missing in 93% of instances and is unlikely to be useful and will be removed.

4.1.14 Smoking Status

This feature is missing in 99% of instances and is unlikely to be useful and will be removed.

4.1.15 Number Affected Lobes

This feature is missing in 98% of instances and is unlikely to be useful and will be removed.

4.1.16 Ground Glass Opacity

This feature has a very lopsided distribution between 'yes' and 'no'. With 91% of the instances that have an Ground Glass Opacity response being 'yes' this feature is likely redundant when it comes to classifying accurately and can be removed.

4.1.17 Asymptomatic

This feature is missing in 98% of instances and is unlikely to be useful and will be removed.

4.1.18 Time Between Admission and Diagnosis

This feature is missing in 94% of instances and is unlikely to be useful and will be removed.

4.1.19 Pregnant

This feature is missing in 92% of instances and is unlikely to be useful and will be removed.

4.1.20 Baby Death

This feature is missing in 95% of instances and is unlikely to be useful and will be removed.

4.1.21 Premature Delivery

This feature is missing in 98% of instances and is unlikely to be useful and will be removed.

4.1.22 Hematocrit

This feature is missing in 99% of instances and is unlikely to be useful and will be removed.

4.1.23 Advanced Partial Thromboplastin Time

This feature is missing in 99% of instances and is unlikely to be useful and will be removed.

4.1.24 Fibrinogen

This feature is missing in 99% of instances and is unlikely to be useful and will be removed.

4.1.25 Urea

This feature is missing in 97% of instances and is unlikely to be useful and will be removed.

4.1.26 Monocytes

This feature is missing in all but one instance and is unlikely to be useful and will be removed.

4.1.27 Basophil

This feature is missing in all but one instance and is unlikely to be useful and will be removed.

4.1.28 Cancer

This feature is missing in all but three instances and is unlikely to be useful and will be removed.

4.1.29 Thrombocytes

This feature is missing in all but one instance and is unlikely to be useful and will be removed.

4.2 Post Feature Selection Analysis

4.2.1 Feature Selection without Missing Data dealt with

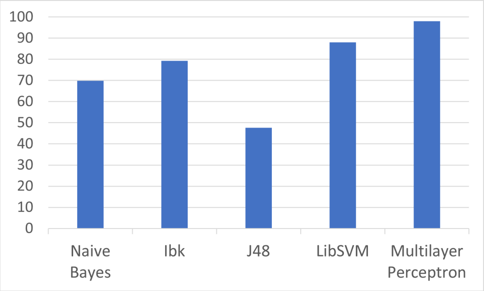


Figure 17: Percent correct classifiers with feature selection

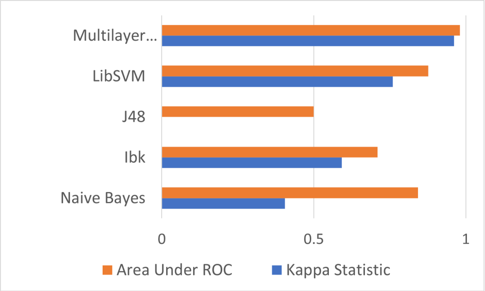


Figure 18: ROC and Kappa statistics with feature selection

When compared to the original classification tests Naive Bayes performs worse in all chosen metrics. Further to this it also can't classify covid cases as well as the original, one if its biggest perks.

IBk however shows an improvement across chosen metrics. This will be due to removing noisy features that throw of the K nearest neighbour calculations.

J48 shows a huge decrease in performance across the chosen metrics, simply classifying all cases as covid. This will be due to removing too many of the features resulting in over pruning.

LibSVM shows almost no change in metrics, suggesting that for LibSVM redundant features were removed, providing similar results with a much smaller dataset.

The Multilayer Perceptron classifier also shows almost no change, again suggesting that for Multilayer Perceptron redundant features have been removed.

4.2.2 Feature Selection with Missing Data dealt with

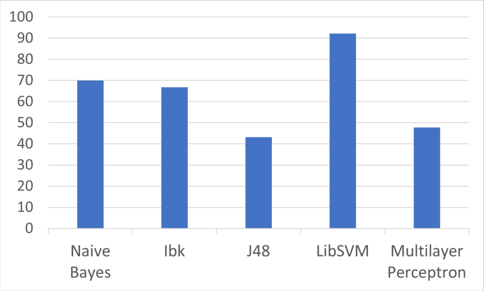


Figure 19: Percent correct classifiers with feature selection and missing data dealt with

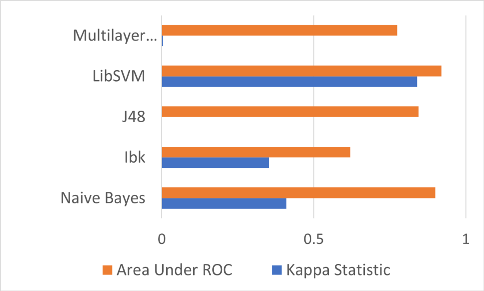


Figure 20: ROC and Kappa statistics with feature selection and missing data dealt with

After feature selection and the missing data is dealt with Naive Bayes classifies with poorer accuracy then the original tests.

IBk and J48 see a noted drop in performance after the missing data is applied to the new feature selected dataset. An improved way to deal with the missing data needs to be found. Given more time a method predicting the missing values could be advantageous over simply using mean for missing data. This is leading to skewed weighting on features, something we tried to remove in feature selection. When a mean is used for the missing data we are reintroducing the issue of skewed data.

LibSVM does however see a small but not statistically significant increase in performance. When compared to all other experimentation on LibSVM this provides the best results and with a much smaller dataset.

Finally the Multilayer Pereceptron classifier sees a significant drop in performance. Now the classifier classifies almost everything as a covid case. Again an issue with how missing is being dealt with is causing large issues with the accuracy of the classifier.

5 Task 4 - Summary of Results

A good model for the full dataset using cross-validation would either be the original multilayer perceptron model or the J48 model where missing data has been replaced with the mean of thee respective attributes. If time is not a concern the multilayer perceptron would be the most accurate and reliable, otherwise the J48 model quickly provides strong results with high kappa and ROC statistics ensuring reliable performance.

The method of dealing with the missing data used can be useful but largely fails when used in conjunction with feature selection, often creating skewed feature data. A more sophisticated method - perhaps making predictions on the missing data - would be more useful. Given more time this is something I would focus on.

The most useful features of in the dataset are *Temperature*, *Region*, and *C Reactive Protein Levels*. These features score the highest in terms of information gain once our feature selection is performed, meaning they are more balanced features and therefore typically more helpful for our classifiers [5]. Feature selection when used properly can provide much smaller datasets and remove potential noise helping classifiers perform better. In this case there were many features that were redundant and noisy with little impact upon overall results.

Having reduced datasets provide a potentially more useful model since we know that it retains its reliability when being applied to small datasets. So even if there is a drop in correctly classified cases of covid or influenza, an argument can be made that our model retains its value.

References

[1] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 04 1960.

[2] Mary McHugh. Interrater reliability: The kappa statistic. *Biochemia medica : časopis Hrvatskoga društva medicinskih biokemičara / HDMB*, 22:276–82, 10 2012.

[3] Jayawant N. Mandrekar. Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9):1315–1316, 2010.

[4] Mpoeleng D. Emmanuel T. Maupong T. A survey on missing data in machine learning. *Journal of Big Data*, 8(140):1–37, 2021.

[5] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Education, page 57, 1997.

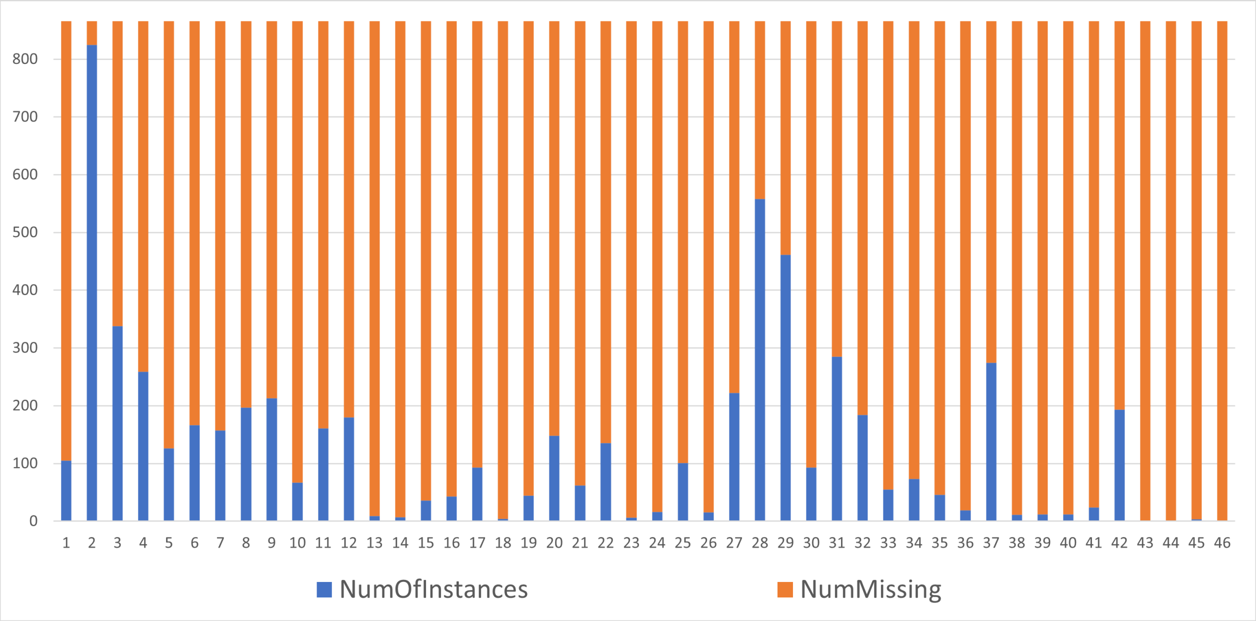


Figure 21: Chart displaying features and the total number of instances with missing data e.g. Feature 1 (Initial PCR Diagnosis) has missing data in 761/866 instances

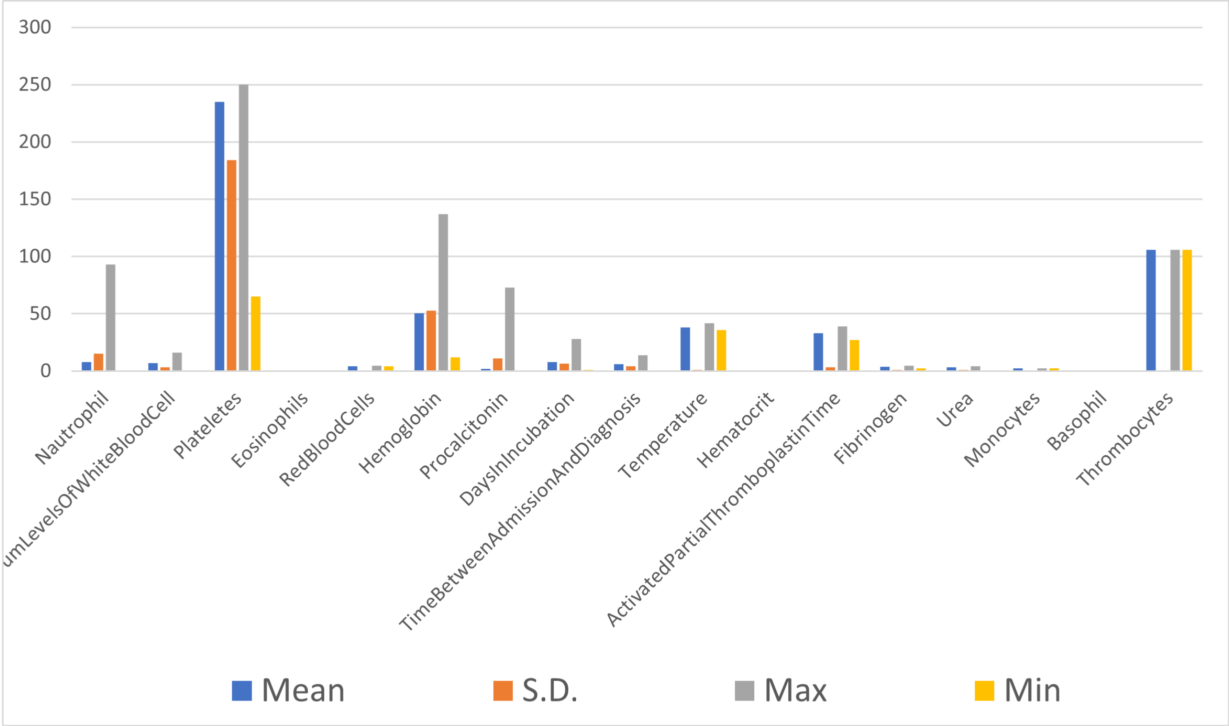
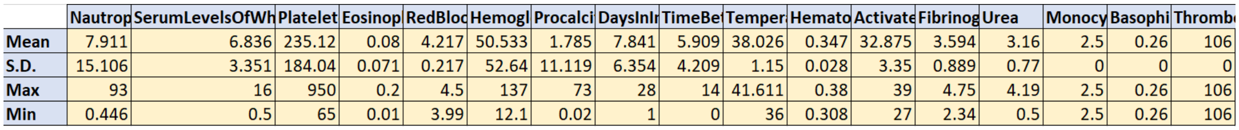


Figure 22: Chart comparing numeric features with *mean*, *standard deviation*, *max*, *minimum* (plateletes max has been reduced to fit chart, should be 950)



	Nautrop	SerumLevelsOfWh	Platelet	Eosinop	RedBloc	Hemogl	Procalci	DaysInI	TimeBet	Temper	Hemato	Activate	Fibrinog	Urea	Monocy	Basophi	Thromb
Mean	7.911	6.836	235.12	0.08	4.217	50.533	1.785	7.841	5.909	38.026	0.347	32.875	3.594	3.16	2.5	0.26	106
S.D.	15.106	3.351	184.04	0.071	0.217	52.64	11.119	6.354	4.209	1.15	0.028	3.35	0.889	0.77	0	0	0
Max	93	16	950	0.2	4.5	137	73	28	14	41.611	0.38	39	4.75	4.19	2.5	0.26	106
Min	0.446	0.5	65	0.01	3.99	12.1	0.02	1	0	36	0.308	27	2.34	0.5	2.5	0.26	106

Figure 23: Feature Table with *mean*, *standard deviation*, *max*, *minimum* breakdown

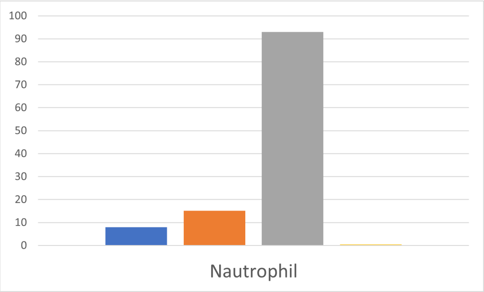


Figure 24: Nautrophil Feature

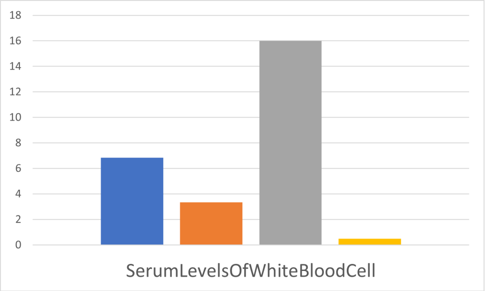


Figure 25: SerumLevelsOfWhiteBloodCell Feature

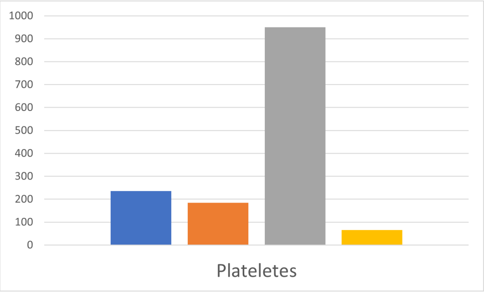


Figure 26: Plateletes Feature

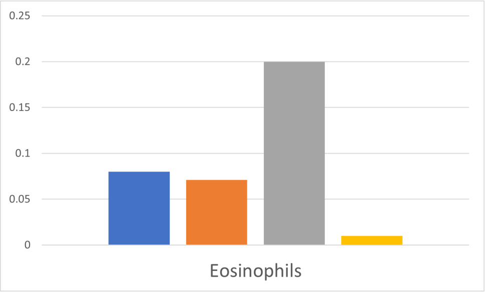


Figure 27: Eosinophils Feature

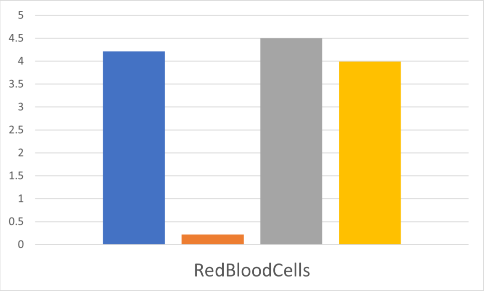


Figure 28: RedBloodCells Feature

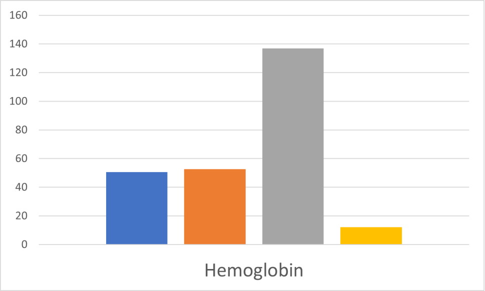


Figure 29: Hemoglobin Feature

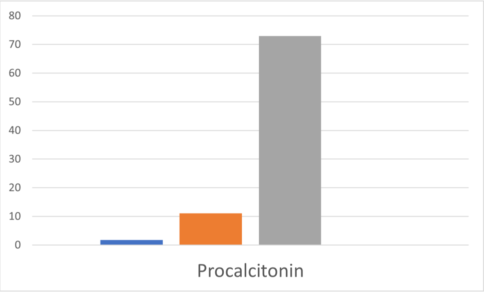


Figure 30: Procalcitonin Feature

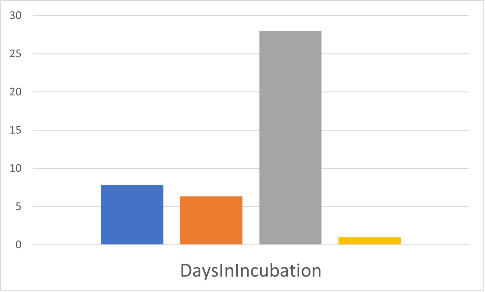


Figure 31: DaysInIncubation Feature

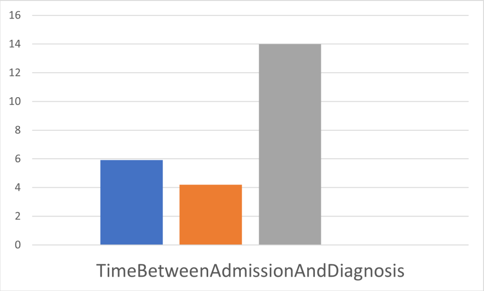


Figure 32: TimeBetween Feature

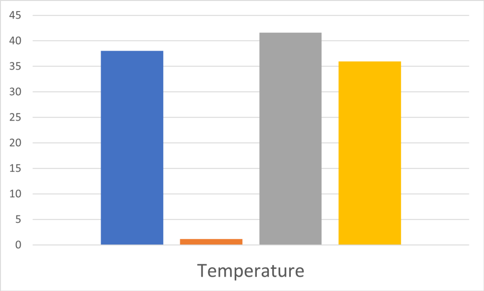


Figure 33: Temperature Feature

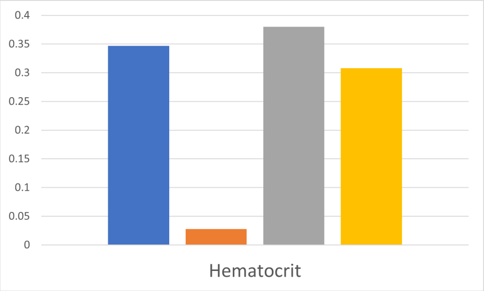


Figure 34: Hematocrit Feature

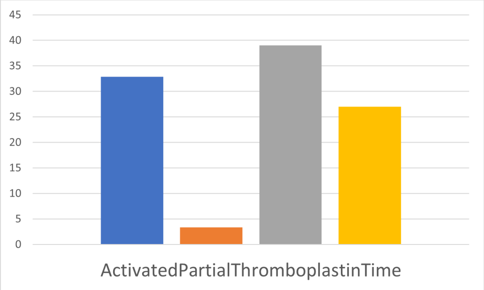


Figure 35: ActivatedPartialThromboplatin Feature

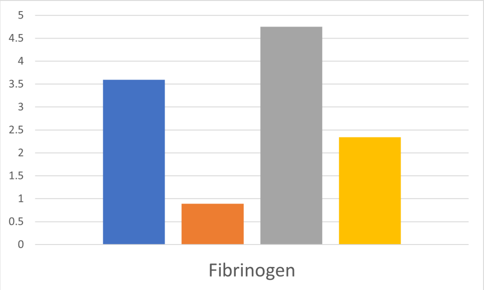


Figure 36: Fibrinogen Feature

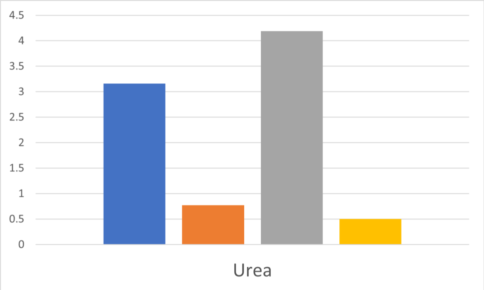


Figure 37: Urea Feature

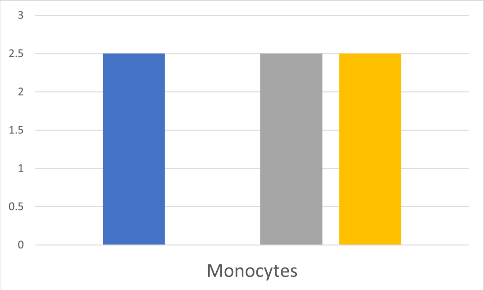


Figure 38: Monocytes Feature

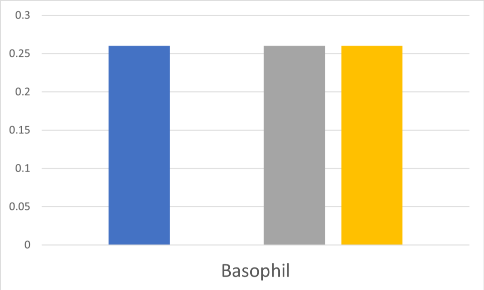


Figure 39: Basophil Feature

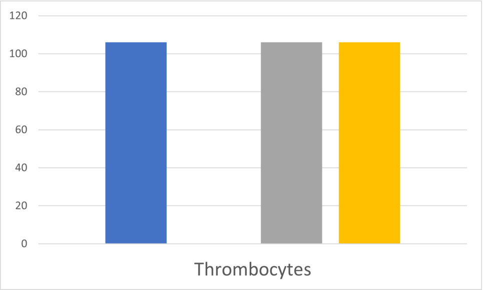


Figure 40: Thrombocytes Feature