CMSC 435 - Introduction to Data Science

Due: December 1st, 2022

Group #01

Brianna Barrett, Eleanor Estwick, Cyaira Hughes, Savannah Nelson, Menelaos Sofroniou

Design Description

The features in all our datasets were generated from Pfeature using the entirety of the amino acid sequences provided in the training file. The first dataset was created using Pfeature's simple amino acid feature generation, in which we used all twenty, generated features. This was done by computing the percent composition of each amino acid in each protein sequence and results in features that represent those percentages. This included descriptors like, but not limited to, the amino acid composition of Alanine, Cysteine, and Aspartic acid.

Our second dataset split the amino acid sequences into two fragments and performed simple amino acid feature generation on both fragments. This turned our initial dataset of twenty features into a new dataset comprising forty features where each feature represents a percent composition of each amino acid in fragment 1 and fragment 2.

For our third dataset we utilized multiple functions for feature generation. Functions included physico-chemical properties of each sequence where desired properties were: positively charged, polarity, cyclic, basicity, hydrophilicity, sulfur content, large, negatively charged, non-polarity, aromaticity, neutral ph, neutral, tiny, neutral charged, aliphaticity, hydrophobicity, hydroxylic and small; structural physico-chemical properties of each sequence where displayed properties were: secondary structure (helix), secondary structure (strands), secondary structure (coil), solvent accessibility (buried), solvent accessibility (exposed), and solvent accessibility (intermediate); physico-chemical properties where the sequence was split in half using same previous desired properties, physico-chemical properties of each sequence removing 5 amino acids from both N and C terminals using same previous desired properties, residue repeats of each sequence, shannon entropy of physico-chemical properties of each sequence using same previous desired properties, simple amino acid composition properties of each sequence, simple amino acid composition properties of each sequence split in half, and simple amino acid composition properties of each sequence removing 5 amino acids from both N and C terminals. This created a final dataset comprised of 200 features.

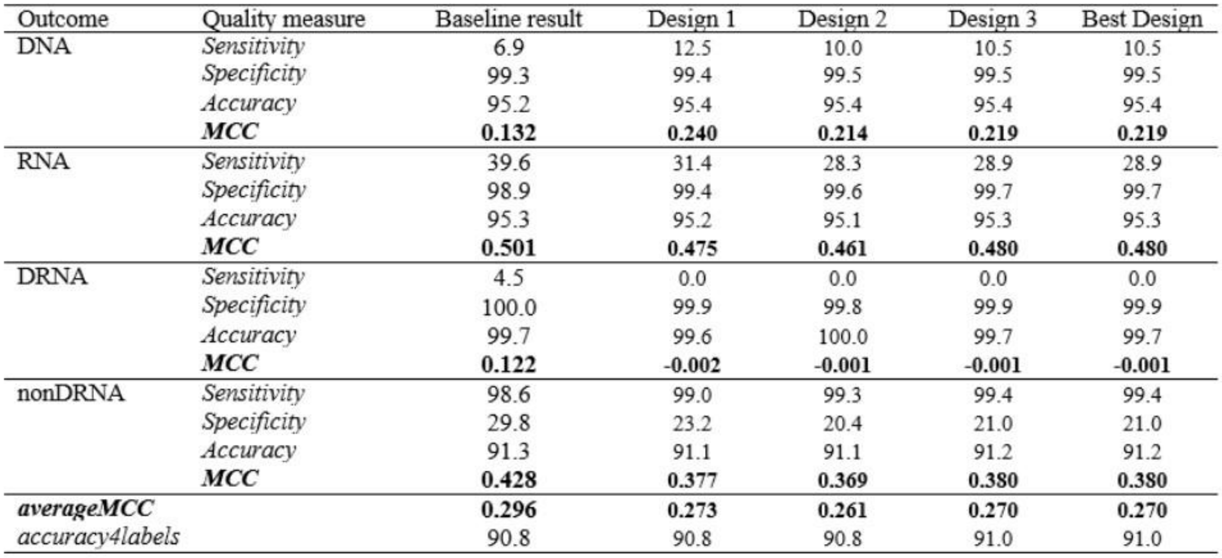
Our first model focused on the **k-NN algorithm** with the parameters k = 5, weighted vote = true, measure types = MixedMeasures and our preferred numerical measures = MixedEuclideanDistance. Our second model used the **k-NN algorithm** with the parameters k = 7, weighted vote = true, measure types= NumericalMeasure and numerical measure to be correlation similarity. Our third model used the **k-NN algorithm** with parameters altered to k = 6, weighted vote = true, measure types = NumericalMeasure and numerical measure = CosineSimilarity.

Ultimately when choosing our best model we elected to use our best performing k-NN model, **model three**. We chose to use this algorithm as it was one of the first algorithms introduced to us in class and therefore an algorithm that we all were relatively familiar with going into the project. We used the k parameter to denote the number of k neighbors and found a balance between k equaling 6-7. We chose this value because if we were to use a higher k value we would run the risk of overfitting our data and if we were to use a lower value we would run the risk of having too much noise factored in. For our measure type we found that numerical measures would be the most appropriate as this matched the format of the data in our dataset. Additionally, when determining the numerical measure parameter for the model, it was decided that CosineSimilarity would be used. This was done due to the fact that certain classes in the test dataset appeared more often, or were a majority, compared to other classes. Cosine similarity disregards magnitude. This means that if a measure appears more frequently in the data it won't be considered more frequently for classification, giving more balance to potentially unbalanced datasets. Using these parameters, along with the k-NN algorithm, we finalized our third iteration as our model of choice.

Our team considered several other algorithms during our design process. We attempted models with Random Forests, Gradient Boosted Trees, and Support Vector Machines, however, we ran into a variety of roadblocks with these approaches. For our SVM model attempt, we found that the dataset had to be manipulated intensively to create a format that the algorithm would be able to use accurately after identifying that SVM is typically used for binary classification. To make this approach work, our team attempted to run the model with one label marked as positive and all other as negative for each classification, however later realized this went against project requirements. Comparatively, attempts at a Random Forest model and Gradient Boosted Trees resulted in no DRNA indication which created issues when trying to calculate the corresponding MCC values. Even though the model seemed to have some accuracy improvement with a higher amount of trees we chose to prioritize our MCC values, and pursued a model that utilized the k-NN algorithm instead.

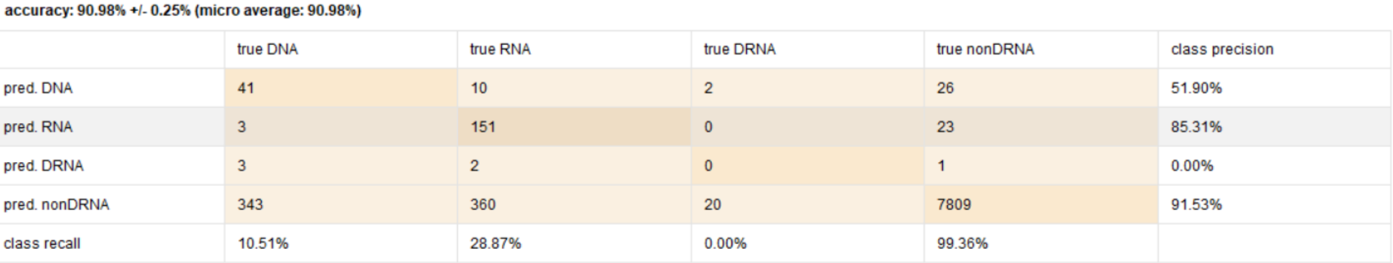
Results:

Table 1. Summary of results based on the 5-fold cross validation on the training dataset.



Outcome	Quality measure	Baseline result	Design 1	Design 2	Design 3	Best Design
DNA	*Sensitivity*	6.9	12.5	10.0	10.5	10.5
	Specificity	99.3	99.4	99.5	99.5	99.5
	Accuracy	95.2	95.4	95.4	95.4	95.4
	MCC	**0.132**	**0.240**	**0.214**	**0.219**	**0.219**
RNA	*Sensitivity*	39.6	31.4	28.3	28.9	28.9
	Specificity	98.9	99.4	99.6	99.7	99.7
	Accuracy	95.3	95.2	95.1	95.3	95.3
	MCC	**0.501**	**0.475**	**0.461**	**0.480**	**0.480**
DRNA	*Sensitivity*	4.5	0.0	0.0	0.0	0.0
	Specificity	100.0	99.9	99.8	99.9	99.9
	Accuracy	99.7	99.6	100.0	99.7	99.7
	MCC	**0.122**	**-0.002**	**-0.001**	**-0.001**	**-0.001**
nonDRNA	*Sensitivity*	98.6	99.0	99.3	99.4	99.4
	Specificity	29.8	23.2	20.4	21.0	21.0
	Accuracy	91.3	91.1	91.1	91.2	91.2
	MCC	**0.428**	**0.377**	**0.369**	**0.380**	**0.380**
averageMCC		**0.296**	**0.273**	**0.261**	**0.270**	**0.270**
accuracy4labels		90.8	90.8	90.8	91.0	91.0

In the DNA outcome quality measurements our best design has increased sensitivity compared to design 2, an increase in specificity compared to design 1, and similar accuracy compared to all other iterations. For this category, DNA, there is also a notable increase in MCC value from iteration 2. In the RNA outcome quality category our best design has an increase in sensitivity compared to iteration 2, a slight increase in specificity in comparison to iteration 1 and iteration 2, and an increase in accuracy and the MCC value when compared to iteration 1 and iteration 2. The DRNA outcome results show the sensitivity to remain the same in all three iterations. Regarding the other values, our best design possesses an increase in specificity compared to iteration 2 and an increase in accuracy compared to iteration 1. For the MCC in this category, DRNA, an increase is represented in our best design value compared to iteration 1. In the last outcome, nonDRNA, the sensitivity in our best design increases from both iteration 1 and iteration 2, its specificity increases compared to iteration 2, and the accuracy and MCC value of our best design has increased compared with iteration 1 and iteration 2. Finally, when comparing the average MCC and accuracy for all labels between each iteration, it can be noted that there is an increase in the MCC value when comparing the best design to iteration 2 and an overall increase in accuracy for labels for our best design when compared to design 1 and 2.



accuracy: 90.98% +/- 0.25% (micro average: 90.98%)

	true DNA	true RNA	true DRNA	true nonDRNA	class precision
pred. DNA	41	10	2	26	51.90%
pred. RNA	3	151	0	23	85.31%
pred. DRNA	3	2	0	1	0.00%
pred. nonDRNA	343	360	20	7809	91.53%
class recall	10.51%	28.87%	0.00%	99.36%	

Of the 8794 proteins to classify in the blind test dataset, 8 were classified as DRNA, 72 as DNA, 191 as RNA, and 8523 as nonDRNA. Comparatively, the contents of the similarly-sized training dataset consists of 22 DRNA proteins, 391 DNA proteins, 523 RNA proteins, and 7859 nonDRNA proteins.

Knowing that the blind dataset should have similar class proportions as the training dataset, we confirmed our results validity by calculating each class proportion percentage on both datasets: 0.09% were classified as DRNA, 0.8% as DNA, 2.17% as RNA, and 96.91% as nonDRNA for the blind dataset, compared to 0.25% being DRNA, 4.44% being DNA, 5.95% being RNA, and 89.35% being nonDRNA on the known training dataset. Overall, it seems that the model does create a prediction set that approximately follows the same proportions between both sets, with the majority of proteins being classified as nonDRNA, followed by RNA, DNA, and DRNA.

Advantages of our chosen method are it was easy to understand and implement, fast to run, and is able to handle multi-class (non-binary) labels. It also provided the closest MCC to the baseline for RNA, nonDRNA, and nearly DNA, however did not provide good results for DRNA, which was a disadvantage. Our method couldn't capture DRNA strains at all due to our method not being able to classify DNA and RNA feature strands correlating with each other and capturing it as one entity as DRNA. Our model may ignore DRNA's but improves the rate of which we find DNA and RNA which are more prevalent amino acids in the overall dataset.

Conclusions:

In regards to our best design using 5-fold cross validation, the quality of our model is quite biased towards classifying a given sequence as nonDRNA. Comparing sensitivity among DNA, RNA and DRNA to the sensitivity of nonDRNA, it's clear to see that the number of positive instances among nonDRNA that the model was able to correctly identify is significantly higher than in other classes. When comparing specificity, nonDRNA is significantly lower than that of the other classes indicating that the model is mislabeling a large amount of sequences as nonDRNA when it should not. This bias is also reflected within the total accuracy of all labels as most of the training data is within the class nonDRNA and can therefore mean the high total accuracy does not correlate to an accurate model. Most importantly, the low average MCC value of 0.27 suggests a weak, positive relationship between predicted and actual classes in our model.

The outcome of the DNA present in Table 1 shows that the baseline sensitivity and the best design sensitivity has the biggest difference amongst all the quality measures for DNA. Similarly, the sensitivity, specificity, accuracy, and MCC of the DNA also shows increases in value. The quality measures in our best design with the outcome RNA, show a decrease in sensitivity and MCC value, an increase in specificity, and the same measure for accuracy. The DRNA outcome's quality measures indicate a decrease in sensitivity, a decrease in specificity, the same measure for accuracy, and a decrease in MCC value. In our nonDRNA findings we see an increase in sensitivity measure and a decrease in specificity, accuracy, and MCC value.

Conclusively, when averaging the MCC value there is a decrease in comparison with the baseline average of MCC values and an increase in the accuracy for most of the labels.

Overall, we had a positive learning experience during this project. At the start we were a bit unsure about how to approach this particular problem. However, with some research, some questions and constant communication among group mates we were able to establish multiple methods of solution that we all could understand and felt comfortable performing on our own or with the help of other team members. As a group, we all worked well together and supported each other constantly throughout the entire length of the assignment which led to a good team experience.