CMSC 435
Submitted: 11/21/19
Group: *Chain Gang*

Protein Classification Project

Submitted by:
Michael Fitzgerald, Jacob Unterman, Michael Poblacion, Gavin Alberghini, Angel De La Rosa

1. Description of Predictive Model Design

1.1. Motivation

As a team we decided to embrace the iterative approach asked of us in this project. We started by building simplistic preprocessing methods and comparing them against basic models in order to build up a solid preprocessing foundation. Then we began to swap out model types to see where we could get big performance increases. Once our team decided on a single best preprocessing method and model, we began to hybridize our preprocessing methods in order to boost performance. After this step we switched to ensemble learners which allowed us to achieve our best results. At each of these checkpoints a leading design was selected and defined in section 1.4 below.

1.2. Pre-Processing Methods

1.2.1. CharCount Method

The CharCount method is a Python script that was implemented by our team. The script decodes protein sequence data into counts of amino acids for each sample. These counts were put into a table and used as features for training.

1.2.2. SubString Matching Method

The SubString Matching method is a Python script that was implemented by our team. It looks at each sequence within a protein class and searches other sequences for a pair of length N substrings. Then a count is generated representing the number of times that this substring appears. This information was used to determine if substrings were valuable features to utilize.

1.2.3. N-grams Method

The N-Grams method is a Python script that was implemented by our team. N-Grams represents each sequence as a histogram of occurrences of a length N substring. This method generated a table which we utilized as features for training.

1.2.4. PROFEAT Method

The PROFEAT method is a web server that allowed protein sequence data to be uploaded. It then returns a list of descriptors that our team utilized as features for training. This generated list however, was labeled by index number, so there was no meaning the team could extract from features. PROFEAT is accessible at:
http://bidd2.nus.edu.sg/cgi-bin/profeat2016/protein/profnew.cgi

1.2.5. PFeature Method

The PFeature method is a web server that allowed protein sequence data to be uploaded. Of the various processing abilities, our team selected to utilize both standard and advanced physico-chemical properties as features for our training data. PFeature is accessible at: https://webs.iiitd.edu.in/raghava/pfeature/index.html

1.2.5.1. Standard Physico-chemical properties

- **1.2.5.1.1. Positively Charged**
- **1.2.5.1.2. Polarity**
- **1.2.5.1.3. Cyclic**
- **1.2.5.1.4. Basicity**
- **1.2.5.1.5. Hydrophilicity**
- **1.2.5.1.6. Sulphur Content**
- **1.2.5.1.7. Large**
- **1.2.5.1.8. Negatively Charged**
- **1.2.5.1.9. Non Polarity**
- **1.2.5.1.10. Aromaticity**
- **1.2.5.1.11. Neutral (ph)**
- **1.2.5.1.12. Neutral**
- **1.2.5.1.13. Tiny**
- **1.2.5.1.14. Neutral Charged**
- **1.2.5.1.15. Aliphaticity**
- **1.2.5.1.16. Acidity**
- **1.2.5.1.17. Hydrophobicity**
- **1.2.5.1.18. Hydroxylic**
- **1.2.5.1.19. Small**

1.2.5.2. Advanced Physico-chemical properties

- **1.2.5.2.1. Z1**
- **1.2.5.2.2. Z2**
- **1.2.5.2.3. Z3**
- **1.2.5.2.4. Z4**
- **1.2.5.2.5. Z5**

1.2.6. Feature Set Reduction

Weight by SVM was used in RapidMiner. This operator outputs a set of coefficients calculated by an SVM and represents the attribute weights. The attributes which were assigned the least weight values (<.02) were removed. 5 features were deemed as noisy through the use of weight by SVM (Aromaticity, Neutral Charged, A, C, and z2).

1.3. Noticeable Model Trends

1.3.1. Decision Tree

Utilized as a base model for our preprocessing comparisons. Decision Trees were consistent, but had comparatively low accuracy and MCC when compared to our later designs.

1.3.2. SVM

Consistently better performing than Decision Tree. Multiple variations of SVM models were attempted.

1.3.3. kNN, Neural Nets

No significant knowledge learned, regardless of parameters.

1.3.4. Random Forest

Best reported accuracy and MCC for our team. No other attempted ensemble method could perform better.

1.4. Design Selections

1.4.1. Design 1

- **1.4.1.1. PreProcessing**: CharCount
- **1.4.1.2. Model**: Decision Tree
- **1.4.1.3. Evaluation**: 5-fold Cross-Validation
- **1.4.1.4. Criterion:** Gain Ratio
- **1.4.1.5. Maximal Depth:** 20
- **1.4.1.6. Pruning:** Yes
- **1.4.1.7. Confidence:** 0.2
- **1.4.1.8. Pre-pruning:** No

1.4.2. Design 2

- **1.4.2.1. PreProcessing**: PROFEAT
- **1.4.2.2. Model**: SVM (Polynomial-Binomial)
- **1.4.2.3. Evaluation**: 5-fold Cross-Validation
- **1.4.2.4. *Kernel Type:*** Polynomial
- **1.4.2.5. *Kernel Degree:*** 1.2
- **1.4.2.6. *C:*** 0.6
- **1.4.2.7. *Convergence Epsilon:*** 0.278
- **1.4.2.8. *L Pos:*** 0.246
- **1.4.2.9. *L Neg:*** 0.468
- **1.4.2.10. *Epsilon:*** 0.624
- **1.4.2.11. *Epsilon Plus:*** 0.2
- **1.4.2.12. *Epsilon Minus:*** 0.36

1.4.3. Design 3

- **1.4.3.1. PreProcessing**: PFeature
- **1.4.3.2. Model**: Random Forest
- **1.4.3.3. Evaluation**: 5-fold Cross-Validation
- **1.4.3.4. *Number of trees:*** 30
- **1.4.3.5. *Criterion:*** Gini Index
- **1.4.3.6. *Max Depth:*** 49

1.4.4. Design 4

- **1.4.4.1. PreProcessing**: PFeature and CharCount
- **1.4.4.2. Model**: SVM (Polynomial-Binomial)
- **1.4.4.3. Evaluation**: 5-fold Cross-Validation
- **1.4.4.4. *Kernel Type:*** Radial
- **1.4.4.5. *Kernel Degree:*** .1
- **1.4.4.6. *C:*** -0.6
- **1.4.4.7. *Convergence Epsilon:*** 9.0 x 10^-4

1.4.5. Best Design

- **1.4.5.1. PreProcessing**: PFeature, CharCount, Feature Set Reduction
- **1.4.5.2. Model**: Random Forest
- **1.4.5.3. Evaluation**: 5-fold Cross-Validation
- **1.4.5.4. *Number of trees:*** 35
- **1.4.5.5. *Criterion:*** Information Gain
- **1.4.5.6. *Max Depth:*** 30

2. Results of predictive models

Outcome	Quality Measure	Baseline Result	Design 1	Design 2	Design 3	Design 4	Best Design
DNA	Sensitivity	6.9	23.08	7.87	55.0	80.0	78.0
	Specificity	99.3	95.29	96.29	95.79	95.94	95.97
	Predictive ACC	95.2	95.17	84.84	95.60	95.86	95.87
	MCC	0.132	0.07	0.06	0.17	.26	.27
RNA	Sensitivity	39.6	45.49	11.69	77.58	80.89	78.83
	Specificity	98.9	95.29	96.29	96.49	96.21	96.40
	Predictive ACC	95.3	93.78	72.60	95.87	95.78	95.85
	MCC	0.501	0.30	0.15	0.56	.54	.55
DRNA	Sensitivity	4.5	14.29	50.00	100.0	66.67	100.0
	Specificity	100	99.76	99.76	99.76	99.77	99.77
	Predictive ACC	99.7	99.69	99.75	99.76	99.76	99.77
	MCC	0.122	0.08	0.15	0.21	.25	.30
nonDRNA	Sensitivity	98.6	91.01	92.13	92.13	91.98	92.22
	Specificity	29.8	51.78	14.68	81.57	86.39	84.97
	Predictive ACC	91.3	89.50	60.58	91.73	91.79	91.95
	MCC	0.428	0.27	0.11	0.45	.46	.47
Average MCC		0.265	0.18	0.12	0.35	.38	.40
Accuracy		90.8	89.07	58.89	91.48	91.60	91.72

3. Conclusions

3.1. Results

As a team, we are satisfied with the results we have been able to achieve. We incrementally increased our class prediction accuracy, overall accuracy, and MCC values. Comparatively, our accuracy is reported closely to the baseline results. However, our MCC value is considerably higher, pointing to a better overall system. The accuracy is a poor measure of the system because of the enormous weight to one class (nonDRNA). Which is why having a higher MCC value is a much better benchmark to compare against.

3.2. Experiences

For the duration of this project, our team spent a large portion of time performing research on potential preprocessing methods for the training data. As something that isn't often experienced in our curriculum, it was interesting to have this new open ended aspect in our project. In terms of model evaluation our team would spend large chunks of time running models. Then we would meet and record the best performing ones in table one. Overall this project is a unique experience that we feel was very valuable in understanding the complexities of Data Science.

3.3. Advantages

One of the key advantages of our model is that it is easily reproducible. This is due to the fact that there is a short build time that doesn't require any complex procedures to build the system. Another advantage is that it is easy to explain how the model (Random Forest) works at a high level. Finally and most importantly, our model has improved accuracy and MCC metrics, making it quantitatively a better system overall.

3.4. Disadvantages

The key disadvantage of our model is the use of web servers to generate feature sets. This is unideal for a couple reasons, the first being that we have no control over when these servers go offline. Another issue is that we don't control the traffic of the webserver and can get slow response times when processing data. The most impactful disadvantage is that some of the features provided by these web servers have no description.

3.5. Summary

Overall, with both pros and cons considered, our team still feels that we have created a competent system. Our results point to a better predictive model not only overall but in each individual class as well. This gives us a strong indication that we are performing well.