Introduction

At the intersection between Applied Mathematics, Computer Science and an understanding of Biological Sciences is subset of knowledge known as Bioinformatics. Some find Bioinformatics and its more general title Data Science difficult to define. The most ubiquitous pictograph is indeed the intersection of three fields of study. See Figure 1. Data Science is simply the overall title when more general domain knowledge is discussed. Other fields you may come across would be, the study of business or marketing begets Business Intelligence. The study of chemistry begets the field Chemoinformatics.[1] [2] Health or Healthcare begets the field Healthcare-Informatics.[3]

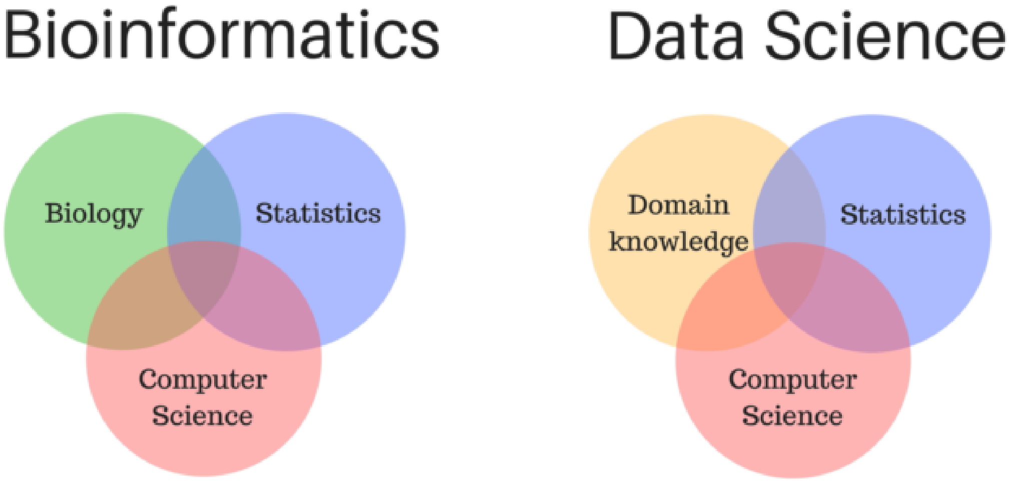

Figure 1: Venn Diagrams of Bioinformatics Vs Data Science

[4]

What is Predictive Modeling?

The term ‘Predictive Modeling’ should bring to mind work in the computer science field also called Machine Learning (ML), Artificial Intelligence (AI), Data Mining, Knowledge discovery in databases (KDD), and possibly even encompassing Big Data as well.

> Indeed, these associations are appropriate and the methods implied by these terms are an integral piece of the predictive modeling process. But predictive modeling encompasses much more than the tools and techniques for uncovering patterns within data. The practice of predictive modeling defines the process of developing a model in a way that we can understand and quantify the model’s prediction accuracy on future, yet-to-be-seen data.[5]

[1] https://www.acs.org/content/acs/en/careers/college-to-career/chemistry-careers/cheminformatics.html

[2] https://jcheminf.biomedcentral.com/

[3] https://www.usnews.com/education/best-graduate-schools/articles/2014/03/26/consider-pursuing-a-career-in-health-informatics

[4] http://omgenomics.com/what-is-bioinformatics/

[5] Max Kuhn, Kjell Johnson, Applied Predictive Modeling, Springer, ISBN:978-1-4614-6848-6, 2013

— Applied Predictive Modeling

As a heads-up, I use `Predictive Modeling` and `Machine Learning` interchangably in this document.

In the booklet entitled, "The Elements of Data Analytic Style" [6] there is an interesting checklist for the uninitiated into the realm of science report writing and indeed scientific thinking. A shorter more succinct listing of the steps, which I prefer, and is described by Roger Peng in his book, The Art Of Data Science. The book lists what he describes as the "Epicycle of Analysis." [7]

The Epicycle of Analysis

1. Stating and refining the question
2. Exploring the data
3. Building formal statistical models
4. Interpreting the results
5. Communicating the results

Therefore let us start by posing a question;

- Is there a correlation between the data points which are outliers from principal component analysis (PCA) and 6 types of predictive modeling?

This experiment is interested in determining if PCA would provide information on the false-positives and false-negatives that were an inevitable part of model building and optimization. The six predictive models that have chosen for this work are Logistic Regression, Support Vector Machines (SVM) (linear, polynomial, and radial basis function kernels), Random Forest and a Neural Network which uses Auto-encoding.

It is common for Data Scientists to test their data sets for feature importance and feature selection. One test that has interested this researcher is Principal component analysis. It can be an extremely useful tool. PCA is an unsupervised machine learning technique which "reduces data by geometrically projecting them onto lower dimensions called principal components (PCs), with the goal of finding the best summary of the data using a limited number of PCs." [8] However, the results that it provides may not be immediately intuitive to the lay person.

How do the advantages and dis-advantages of using PCA compare with other machine learning techniques. The advantages are numerable and include dimensionality reduction, filtering out noise inherent in the data and it may preserve the global structure of the data. Does the global and graphical structure of the data produced by the first two principal components provide any insights into how the predictive models of Logistic Regression, Neural Networks utilizing auto-encoders, Support Vector Machines and Random Forest. In essence, is PCA sufficiently similar to any of the applied mathematics tools of more advanced approaches. Also, this work is to simply teach myself machine learning or predictive modeling techniques.

The data for this study was derived from the Uniprot database. From the Uniprot database was queried for two protein groups. The first group was Myoglobin and the second was a control group comprised of human proteins not related to Hemoglobin or Myoglobin. See Figure 2 There have been a group of papers that are striving to classify types of proteins by their amino acid structure alone. The simplest classification procedures involve using the percent amino acid composition. This is calculated by using the count of a amino acid over the total number in that protein.

[6] Jeff Leek, The Elements of Data Analytic Style, A guide for people who want to analyze data., Leanpub Books, http://leanpub.com/datastyle, 2015

[7] Roger D. Peng and Elizabeth Matsui, The Art of Data Science, A Guide for Anyone Who Works with Data, Leanpub Books, http://leanpub.com/artofdatascience, 2015

[8] Jake Lever, Martin Krzywinski, Naomi Altman, Principal component analysis, Nature Methods, Vol.14 No.7, JulyY 2017, 641-2

Percent Amino Acid Composition:

$$\%AAC_X \; = \; \frac{N_{Amino\ Acid\ X}}{Total\ N\ of\ AA}$$

The experimental approach carried out Exploratory Data Analysis to determine if the features were skewed and needed to be transformed. In a random system where amino acids were chosen at random one would expect the percent amino acid composition to be close to 5%. However this is far from the case for the Myoglobin proteins or the control protein samples.

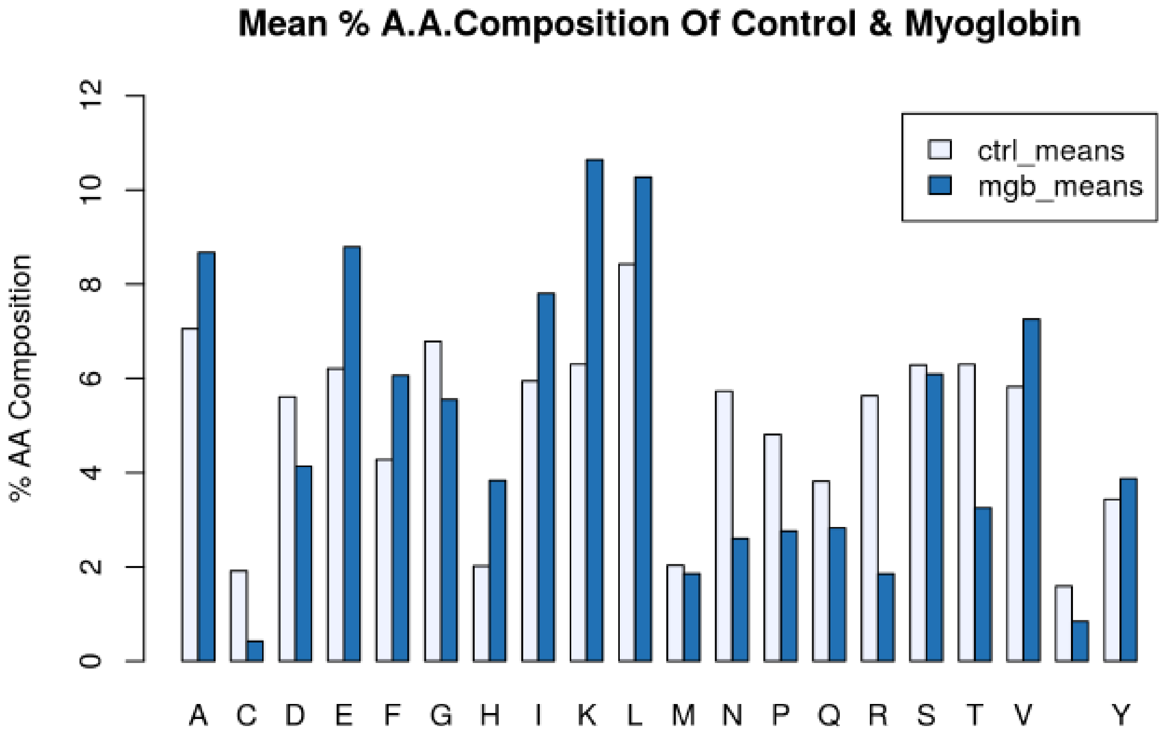

Figure 2: Mean % Amino Acid Compositions for Control & Myoglobin

Predictive Modeling

In general, there are four types of Predictive Modeling or machine learning approaches

1. Supervised,
2. Unsupervised,
3. Reinforcement,
4. Semi-Supervised.

For the sake of this document only the first two approaches (Supervised & Unsupervised learning) will be discussed.

Supervised Learning

In supervised learning, data consists of observations X_i (where X may be a matrix of values) that also contains a corresponding label, y_i. The label y may be anyone of C classes. In our case of a binary classifier, we have {‘Is myoglobin’, ‘Is control’}.

Data set: $(X_1, y_1), (X_2, y_2),\ ...,\ (X_N, y_N);\quad y \in \{1, ...,\ C\}$, where C is the number of classes

A machine learning algorithm determines a pattern from the input information and groups this with its necessary title or classification.

One example might be that we require a machine which separates red widgets from blue widgets. One predictive algorithm is called K-Nearest Neighbor (K-NN). K-NN looks at an unknown object and then proceeds to calculate the distance (most commonly the euclidean distance) to the K nearest neighbors. If we consider the figure below and choose $K = 3$, we would find a circumstance as shown. In the dark solid black on the K-Nearest-Neighbor figure we find that the green widget is nearest to two red widgets and one blue widget. In the voting process K-NN algorithm (2 reds vs 1 blue) means that the consignment of our unknown green object is red.

In order for the K-NN algorithm to function optimally the data most be complete with a set of features and a label of each item. Without the corresponding label a data scientist would need a different criteria to track the widgets.

Six of the seven algorithms that this report investigates are supervised. Logit, support vector machines, Random Forest and the neural network that I have chosen require labels for the classification process.

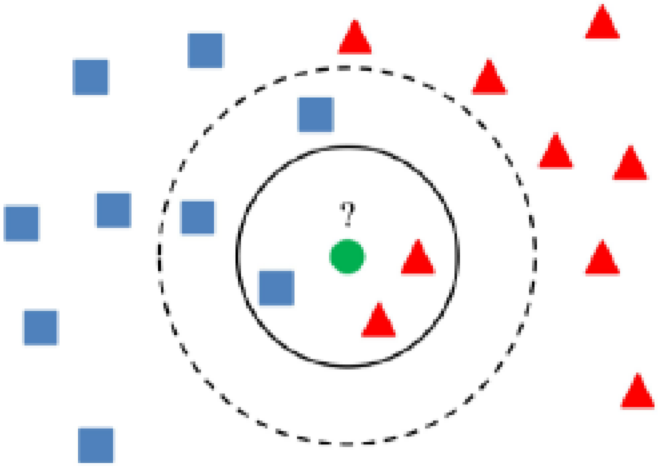

Figure 3: K-Nearest-Neighbor

[9]

What is a shallow learner?

Let us investigate the K-NN algorithm and figure a little further. If we change our value of K to 5 then we see a different result. By using $K = 5$ we will now consider the out dashed-black line. This larger K value contains three blue widgets and two red widgets. If we are now asked to vote our choice, we find that 3 blue beats the 2 red and we assign the unknown a BLUE widget. This assignment is the opposite of the inner circle.

If a researcher were to use K-NN then the algorithm would have to test many possible K values and compare the results, then choose the K with the highest accuracy. However, this is where K-NN falters. To use K-NN the computer algorithm needs to keep all data points it used for its initial training (accuracy testing). Any new unknowns could be conceivably tested against any or all the previous data points. The K-NN does use a generalized rule that would make future assignment quick on the contrary it must memorize all the points

[9] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

for the algorithm to work. K-NN cannot delete the points until it is complete. It is true that the algorithm is simple but not efficient. Matter and fact, as the number of feature dimensions increases this causes the complexity (also known as Big O) to rise. The complexity of K-NN is: $O(K - NN) \;\propto\; nkd$.

Where n is the number of observations, k is the number of nearest neighbors it must check and d is the number of dimensions.[10]

Given that K-NN tends to ‘memorize’ its data to complete its task it is considered a lazy and a shallow learner. Lazy indicates that the decision is left to the moment a new point is learned of predicted. If we were to use a more generalized rule, such as **{Blue** for $(x <= 5)$**}** this would be more dynamic and deeper approach by comparison.

Unsupervised Learning

By contrast to the supervised learning system, unsupervised learning does not require a label for it to operate.

Data set: $(X_1), (X_2),\ ...,\ (X_N)$ where X may represent a matrix (m observations by n features) of values.

Principal Component Analysis is an example of unsupervised learning, which we will discuss in more detail in chapter 3. The data despite or without its labels are transformed to provide a maximization of the variances in the dataset. Yet another objective of Unsupervised learning is to discover “interesting structures”[11] in the data. There are several methods that structure may show itself these include clustering, knowledge discovery of latent variable or discovering graph structure. In many instances and as a subheading to the a fore mentioned points unsupervised learning is used for dimension reduction or feature selection.

Among the simplest unsupervised learning algorithms is K-means. K-means does not rely on the class labels of the dataset at all. In fact, K-means may be used to determine any number of classes despite any predetermined values. K-means can discover clusters which may then be used in classification or hierarchical feature representation. K-means has several alternative methods but in general calculates the distance (or conversely the similarity) of observations to a mean value of the Kth grouping. The mean value is called the center of mass, the Physics term that provides an excellent analogy since the center of mass is a weighted average. By choosing different number of groupings (values of K much like the K-NN) then comparing the grouping by a measure of accuracy, one example being, mean square error.

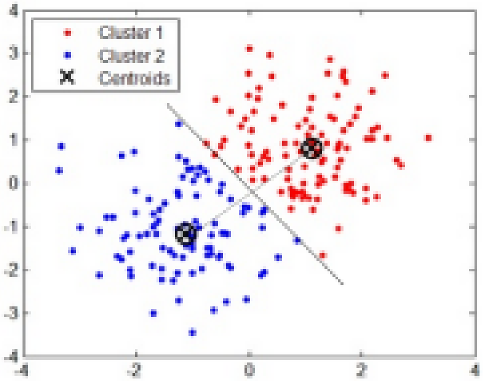

Figure 4: K-Means

[12]

It is easy to see through much or machine learning or predictive modeling if one understands bits of the inner workings of these algorithms,

[10] Olga Veksler, Machine Learning in Computer Vision, http://www.csd.uwo.ca/courses/CS9840a/Lecture2_knn.pdf

[11] Kevin Murphy, Machine learning a probabilistic perspective, 2012, ISBN 978-0-262-01802-9

[12] https://www.slideshare.net/teofili/machine-learning-with-apache-hama/20-KMeans_clustering_20

Four Challenges In Predictive Modeling

To many predictive modeling is a panacea for all sorts of issues. Although it does show promise there are hurdles which need to be overcome. Martin Jaggi[13] has summarized four points which elucidate current problems in the field that need to be tackled.

Problem 1: The vast majority of information in the world in unlabeled so it would be advantageous to have a good Unsupervised machine learning algorithms to use.

Problem 2: Algorithms are very specialized, too specific.

Problem 3: Transfer learning to new environments

Problem 4: Scale, the scale of information is huge in reality and we have computers that work in gigabytes not the Exabytes that humans may have available to them. The scale of distributed Big Data…

The predictive models which are executed in this report will be discusses in further detail in their own sections.

Experimental Procedure

The experimental procedure will be broken into 3 major steps.

Exploratory data Analysis (EDA)

During EDA the data was checked for irregularities, such as missing data, outliers among features, skewness, and visually for normality using QQ-plots. The only irregularity that posed a significant issue was the skewness of the amino acid features. Many of 20 amino acid features had a significant number of outliers as seen by Boxplot analysis, however only three features had skew which might have presented a problem. Dealing with the skew of the AA was important due to the fact that Principal Component Analysis was a major aspect of this experiment.

It was determined earlier that three amino acids (C, F, I) from the single amino acid percent composition should be transformed by using the square root function. The choice of transformations was Natural log, log base 10, squaring (x^2) and using the reciprocal ($1/x$) of the values. The square root transformation lowered the skewness to values of less than 1.0 from highpoints of greater than 2 in all three cases to {-0.102739 $\leq$ skew after transformation $\leq$ 0.3478132}.

Amino Acid	Initial skewness	Skew after square root transform
C, Cysteine	2.538162	0.347813248
F, Phenolalanine	2.128118	-0.102739748
I, Isoleucine	2.192145	0.293474879

Once the three features were transformed this dataset was denoted

`~/00-data/02-aac_dpc_values/c_m_TRANSFORMED.csv` and used throughout the rest of the analysis.

It was decided early in the work that R[14], RStudio[15] and its machine learning library / framework `caret`[16] would be used

Caret library for R

[13] https://www.machinelearning.ai/machine-learning/4-big-challenges-in-machine-learning-ft-martin-jaggi-2/

[14] https://cran.r-project.org/

[15] https://rstudio.com/

[16] http://topepo.github.io/caret/index.html

The R/caret library is attractive to use for many reasons. It currently allows some 238 machine learning models which have been written using disparate options and data structures. [17] The utility of caret is that it organizes the input and output into a common format making the need for learning only one grammar and syntax. caret also harmonizes the use of hyperparamters such that they are more easily repeated.

Data Partitioning

Other functions, used in this work, built into caret is a data partitioning subroutine. The partitioning is carried out such that it attempts to avoid imbalanced data. In short, the program balances the number of positive samples to negative samples, etc. Within this experiment data partitioning was set to 80 parts for training versus 20 parts for testing.

```
# Partition data into training and testing sets
set.seed(1000)
index <- createDataPartition(c_m_TRANSFORMED.csv$Class, p = 0.8, list = FALSE)

training_set <- c_m_9aa[ index, ]
test_set     <- c_m_9aa[-index, ]
```

Training the Predictive model

Setting up the training section for caret, for this experiment, can be broken into three parts.

Tuning Hyperparameters

The `tune.grid` command set allows a reasearcher to experiment by varying the hyperparameters of the given model to investigate optimum values. Currently there are no algorithms which allow for the quick and robust tuning of parameters. Instead searching the experiemental space in a methodical fashion is easily carried by defining an n-dimensional grid to search along.

Although some models have many parameters the most common one is to search along the cost hyperparameter.

> The function we want to minimize or maximize is called the objective function or criterion. When we are minimizing it, we may also call it the cost function, loss function , or error function. [18]

The cost function (a term derived from business modeling, i.e. optimizing the cost) is an estimate as how well a models predicted value fits from the actual value. A typical cost function is the square error function which is the difference of the actual value minus the estimated function squared.

Example Cost Function [19]

$$Cost \ = \ \Big(y_i - \hat{f}(x_i)\Big)^2$$

It may be important to search the literature to determine if other researchers have used a specific range of optimum value which may speed your own search. For example, C.W. Hsu at al suggest using a broad range of 20 orders of magnitude of powers of 2,

e.g. `cost =` $2^{-5}, 2^{-3}, ..., 2^{15}$ for an intial gross search then switching to

4 or 5 orders of magnitude with 1/4 log steps.

e.g. `cost =` $2^1, 2^{1.25}, ..., 2^5$ for a fine search for unknown SVM using a radial basis function. [20]

[17] http://topepo.github.io/caret/available-models.html

[18] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, http://www.deeplearningbook.org, 2016

[19] Roberto Battiti and Mauro Brunato, The LION way. Machine Learning-Intelligent Optimization, LIONlab, University of Trento, Italy“, 2017”, http://intelligent-optimization.org/LIONbook

[20] Chih-Wei Hsu, et al, A Practical Guide to Support Vector Classification, 2016, http://www.csie.ntu.edu.tw/~cjlin

k-Fold Cross validation of results

Another valuable option that caret has is the ability to cross validate results.

Cross-validation is a statistical method used to estimate the skill of machine learning models.[21]

> The samples are randomly partitioned into k sets of roughly equal size. A model is fit using the all samples except the first subset (called the first fold). The held-out samples are predicted by this model and used to estimate performance measures. The first subset is returned to the training set and procedure repeats with the second subset held out, and so on. The k resampled estimates of performance are summarized (usually with the mean and standard error) and used to understand the relationship between the tuning parameter(s) and model utility.[22]

Cross-validation has the advantage of using the entire dataset for training and for testing, increasing the opportunity that more training samples will produce a better model.

Example R/caret code:

```
## 10 fold Cross Validation repeated 5 times
fitControl <- trainControl(method = "repeatedcv",
                           number = 10,
                           repeats = 5,
                           savePredictions = "final")
```

Train command

Example Train command

```
model_object <- train(Class ~ .,
                      data = training_set,
                      trControl = fitControl,
                      method = "svmLinear",
                      tune.Grid = grid)
```

Analysis of results

- Compile CM output (Accuracy, MCC, etc.) for comparing 10 models.
- No matter how much data we have, it is very unlikely that we will see those exact examples again at test time.

[21] https://machinelearningmastery.com/k-fold-cross-validation/

[22] Max Kuhn, Kjell Johnson, Applied Predictive Modeling, 2013, ISBN 978-1-4614-6848-6