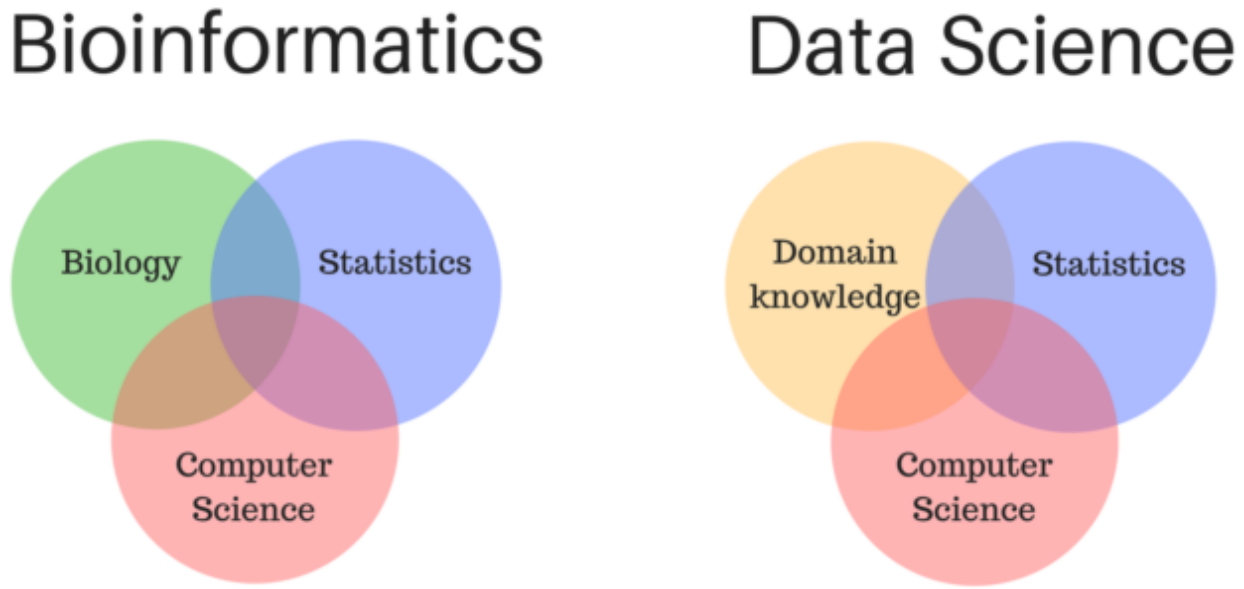
What is Bioinformatics?

At the intersection between Applied Mathematics, Computer Science, and Biological Sciences is a subset of knowledge known as Bioinformatics. The term Bioinformatics was first coined by Paulien Hogeweg and Ben Hesper in 1970 to describe "the study of informatic processes in biotic systems."[1] Coincidentally, the development of DNA and RNA sequencing was co-occurring. Ray Wu established DNA sequencing at Cornell University in 1970.[2] RNA sequencing was developed by Walter Fiers and his coworkers at the University of Ghent, Belgium, in 1972.[3]

Even though the term Data Science was coined in 1996,[4] after the name Bioinformatics was envisioned, Bioinformatics is now considered an offshoot of the field of Data Science. What separates these two pursuits is the domain knowledge that each focus on. Data Science now takes on such topics as Business, Sales, Marketing and includes Science related topics. While Bioinformatics has grown to encompass such ideas as Genomics,[5] Proteomics,[6] Metabolomics,[7] protein structure,[8] Chemical biology[9] and even Systems Biology.[10]



[11]

[1] Paulien Hogeweg, "The Roots of Bioinformatics in Theoretical Biology", PLoS Comput Biol. 2011 Mar; 7(3): e1002021, doi: 10.1371/journal.pcbi.1002021

[2] https://web.archive.org/web/20090304121126/http://www.mbg.cornell.edu/faculty-staff/faculty/wu.cfm

[3] Min Jou W, Haegeman G, Ysebaert M, Fiers W (May 1972). "Nucleotide sequence of the gene coding for the bacteriophage MS2 coat protein". Nature. 237 (5350): 82–8. Bibcode:1972Natur.237...82J. doi:10.1038/237082a0.

[4] https://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/#76fa330155cf

[5] https://www.genome.gov/about-genomics/fact-sheets/A-Brief-Guide-to-Genomics

[6] https://www.ebi.ac.uk/training/online/course/proteomics-introduction-ebi-resources/what-proteomics

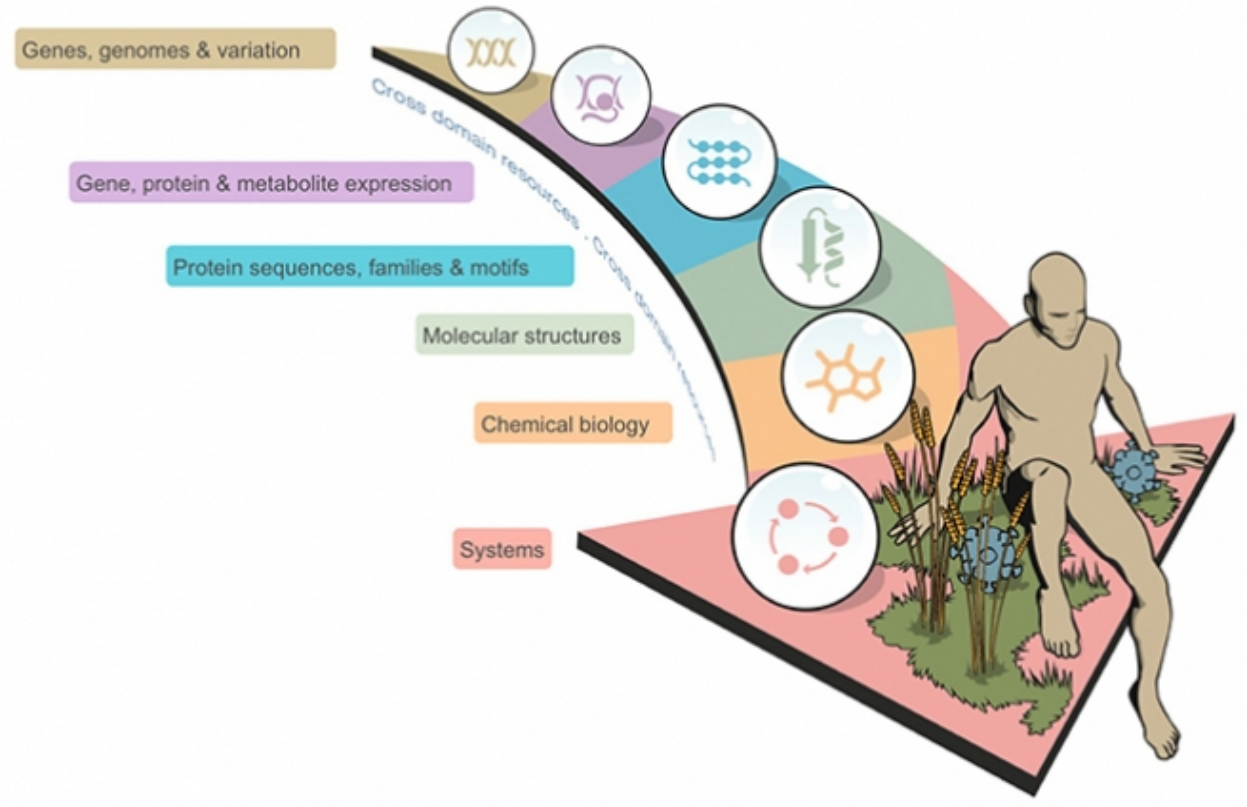
[7] https://www.ebi.ac.uk/training/online/course/introduction-metabolomics/what-metabolomics

[8] https://www.ncbi.nlm.nih.gov/books/NBK26830/

[9] https://pubs.acs.org/journal/acbcct#

[10] https://irp.nih.gov/catalyst/v19i6/systems-biology-as-defined-by-nih

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



[12]

Ultimately, Bioinformatics and Data Science imply a process as well topics. Roger Peng in his book, "The Art Of Data Science," discusses a useful checklist for the uninitiated into the realm of science report writing and, indeed, scientific thinking.[13] Peng describes the "Epicycle of Analysis."

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.

This is also artful shown in Harvard Data Science course taught by Blitzstein et al.[14]

One piece of this process is Reproducible Research.[15] Reproducible Research and replication are linked but it goes beyond the ability of the work to be *independently verified*. The computer code and data must be provided. There locations explicitly spelled out such that other scientists may find and carry on the work and check all its calculations and methodology. Now that computers play such a large role procedurally the smallest error in a spreadsheet may plague the overall conclusion.

[12] https://www.ebi.ac.uk/training/online/course/bioinformatics-terrified/what-bioinformatics-0

[13] 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

[14] Joe Blitzstein, Hanspeter Pfister, Verena Kaynig-Fittkau, Harvard Data Science course CS109, http://cs109.github.io/2015/index.html

[15] Roger Peng, The Real Reason Reproducible Research is Important, https://simplystatistics.org/2014/06/06/the-real-reason-reproducible-research-is-important/

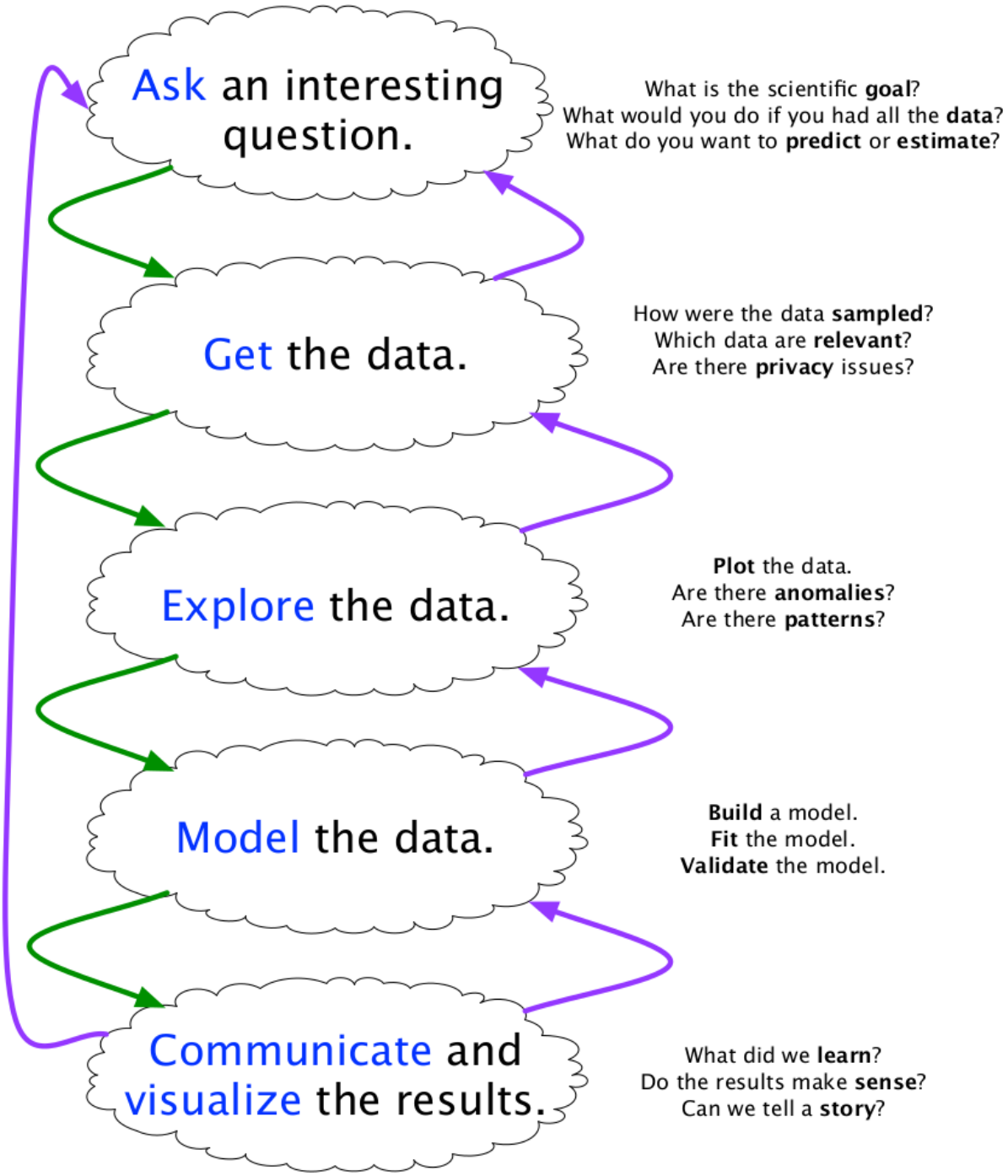


Figure 1: The Data Science Flowchart From Harvard CS109

What is Machine Learning?

> "Machine learning is essentially a form of applied statistics with increased emphasis on the use of computers to statistically estimate complicated functions and a decreased emphasis on proving confidence intervals around these functions"
>
> Ian Goodfellow, et al[16]

What is Predictive Modeling?

Although what this paper discusses is Predictive Modeling, it is under the umbrella of Machine Learning. 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."
>
> Max Kuhn[17]

As an aside, I use `Predictive Modeling` and `Machine Learning` interchangeably in this document.

Predictive Modeling

In general, there are three types of Predictive Modeling or Machine Learning approaches;

1. Supervised,
2. Unsupervised,
3. Reinforcement.

Due to the fact this paper only uses Supervised & Unsupervised learning and for the sake of this brevity, I discuss only the first two types of Predictive Models.

[16] Ian Goodfellow, Yoshua Bengio, Aaron Courville, 'Deep Learning', MIT Press, 2016, http://www.deeplearningbook.org
[17] Max Kuhn, Kjell Johnson, Applied Predictive Modeling, Springer, ISBN:978-1-4614-6848-6, 2013

Supervised Learning

In supervised learning, data consists of observations x_i (where X may be a matrix of real values) AND a corresponding label, y_i. The label y maybe anyone of C classes. In our case of a binary classifier, we have {'Is myoglobin', 'Is control'}.

Data set:

- $(X_1, y_1), (X_2, y_2),\ ...,\ (X_N, y_N) \quad where\ X \in \Re$
- $y \in \{1, ...,\ C\}$, 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 that separates red widgets from blue widgets. One predictive algorithm is called a K-Nearest Neighbor(K-NN) algorithm. 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, the K-NN algorithm (2 reds vs. 1 blue) means that the consignment of our unknown green object is red.

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

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

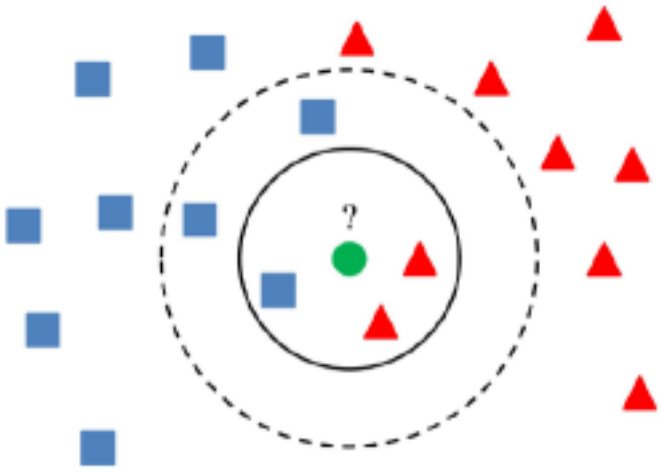


Figure 2: Example of K-Nearest-Neighbor

[18]

What is a shallow learner?

Let us investigate the K-NN algorithm and figure 2.2 (K-Nearest-Neighbor) a little further. If we change our value of K to 5, then we see a different result. By using $K = 5$, we consider the out dashed-black line.

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

This more considerable K value contains three blue widgets and two red widgets. If we ask to vote on 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. The K-NN algorithm needs to keep all data points 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 assignments quick on the contrary. It must memorize all the points 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 $O(\text{K-NN}) \propto nkd$.

Where n is the number of observations, k is the number of nearest neighbors it must check, and d is the number of dimensions.[19]

Given that K-NN tends to 'memorize' its data to complete its task, it is considered a lazy and 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 $(x \leq 5)$} this would be a more dynamic and more in-depth approach by comparison.

Unsupervised Learning

In contrast to the supervised learning system, unsupervised learning does not require or use a **label** or **dependent variable**.

Data set:

- $(X_1), (X_2),\ ...,\ (X_N)$
- **No Y**

where X may represent a matrix of m observations by n features with $\Re$ values.

Principal Component Analysis is an example of unsupervised learning, which we discuss in more detail in chapter 3. The data, despite or without its labels, are transformed to provide maximization of the variances in the dataset. Yet another objective of Unsupervised learning is to discover "interesting structures" in the data.[20] There are several methods that show structure. These include clustering, knowledge discovery of latent variables, or discovering graph structure. In many instances and as a subheading to the aforementioned points, unsupervised learning can be 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. K-means may be used to determine any number of classes despite any predetermined values. K-means can discover clusters later 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 a 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.

[21]

[19] Olga Veksler, Machine Learning in Computer Vision, http://www.csd.uwo.ca/courses/CS9840a/Lecture2_knn.pdf
[20] Kevin Murphy, Machine learning a probabilistic perspective, 2012, ISBN 978-0-262-01802-9
[21] https://www.slideshare.net/teofili/machine-learning-with-apache-hama/20-KMeans_clustering_20

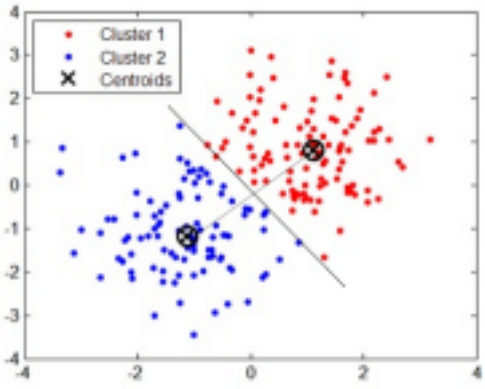


Figure 3: Example of K-Means

Five Challenges In Predictive Modeling

To many predictive modeling is a panacea for all sorts of issues. Although it does show promise, some hurdles need research. Martin Jaggi [22] has summarized four points that elucidate current problems in the field that need research. To this list, I have added one more point, which is commonly called the *Vaiance-Bias Tradeoff*.

Problem 1: The vast majority of information in the world is 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 vast 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 specific predictive models which are executed in this report are discussed in further detail in their own sections.

Problem 5: Bias-Variance Trade-Off.

The ability to generalize is a key idea in predictive modeling. This idea harkens back to freshman classes where one studied Student's t-test and analysis of variance.

$$E\left[\left(y_0 - \hat{f}(x_0)\right)^2\right] = Var(\hat{f}(x_0)) + \left[Bias(\hat{f}(x_0))\right]^2 + Var(\epsilon) \tag{1}$$

The bias-variance dilemma can be stated as follows.[23]

1. Models with too few parameters are inaccurate because of a large bias: they lack flexibility.
2. Models with too many parameters are inaccurate because of a large variance: they are too sensitive to the sample details (changes in the details will produce huge variations).
3. Identifying the best model requires controlling the "model complexity", i.e., the proper architecture and number of parameters, to reach an appropriate compromise between bias and variance.

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

[23] Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning; Data Mining, Inference, and Prediction, https://web.stanford.edu/~hastie/ElemStatLearn/, 2017

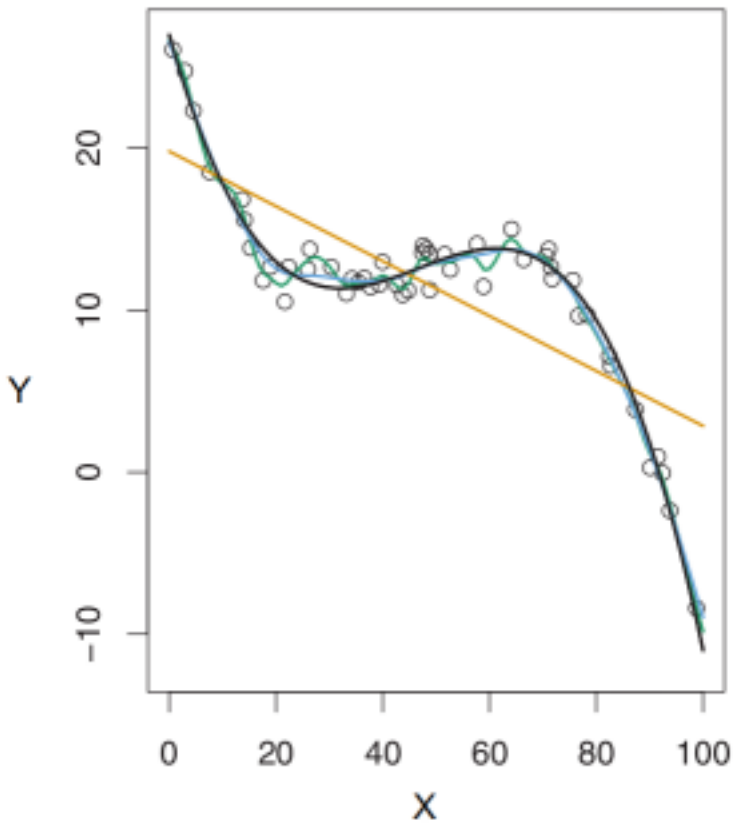


Figure 4: Bias-Variance Tradeoff

One very good example is seen in figure 2.4, *Bias-Variance Tradeoff*. By considering the yellow-orange line we find a simple slope intercept model ($y \propto k \cdot x$) where the variance is high but the bias low and is not flexible enough. This is called underfitting. Looking at the green-blue line we see it follows the data set much more closely, e.g. ($y \propto k \cdot x^8$). Here the variance is very low but common sense tells us that the line is overfit and would not generalize well in a real world setting. Finally leaving us with the black line which does not appear to have too many parameters, ($y \propto k \cdot x^3$).

Research Description

Is there a correlation between the data points, which are outliers from principal component analysis (PCA), and six 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 kernel, and radial basis function kernel, and a Neural Network.

I have studied six different M.L. algorithms using protein amino acid percent composition data from two classes. Class number 1 is my positive control which is a set of Myoglobin proteins, while the second class is a control group of human proteins that do not have Fe binding centers.

Group	Class	Number of Class	Range of Groups
Controls	0 or (-)	1216	1, ..., 1216
Myoglobin	1 or (+)	1124	1217, ..., 2340

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 a 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." [24] However, the results that it provides may not be immediately intuitive to the layperson.

How do the advantages and disadvantages of using PCA compare with other machine learning techniques? The advantages are numerable. They include dimensionality reduction and 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 teach me machine learning or predictive modeling techniques.

The data for this study is 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 1.5, *Percent Amino Acid Composition*. There have been a group of papers that are striving to classify types of proteins by their amino acid structure alone. The most straightforward classification procedures involve using the percent amino acid composition (AAC). The AAC is calculated by using the count of an amino acid over the total number in that protein.

- Percent Amino Acid Composition:

$$\%AAC_X = \frac{N_{Amino\ Acid\ X}}{Total\ N\ of\ AA} \tag{2}$$

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

The Exploratory Data Analysis determines if features were skewed and needed must 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. On top of this the differences between the myoglobin and control proteins can be as high as approximately 5% with the amino acid Lysine, K.

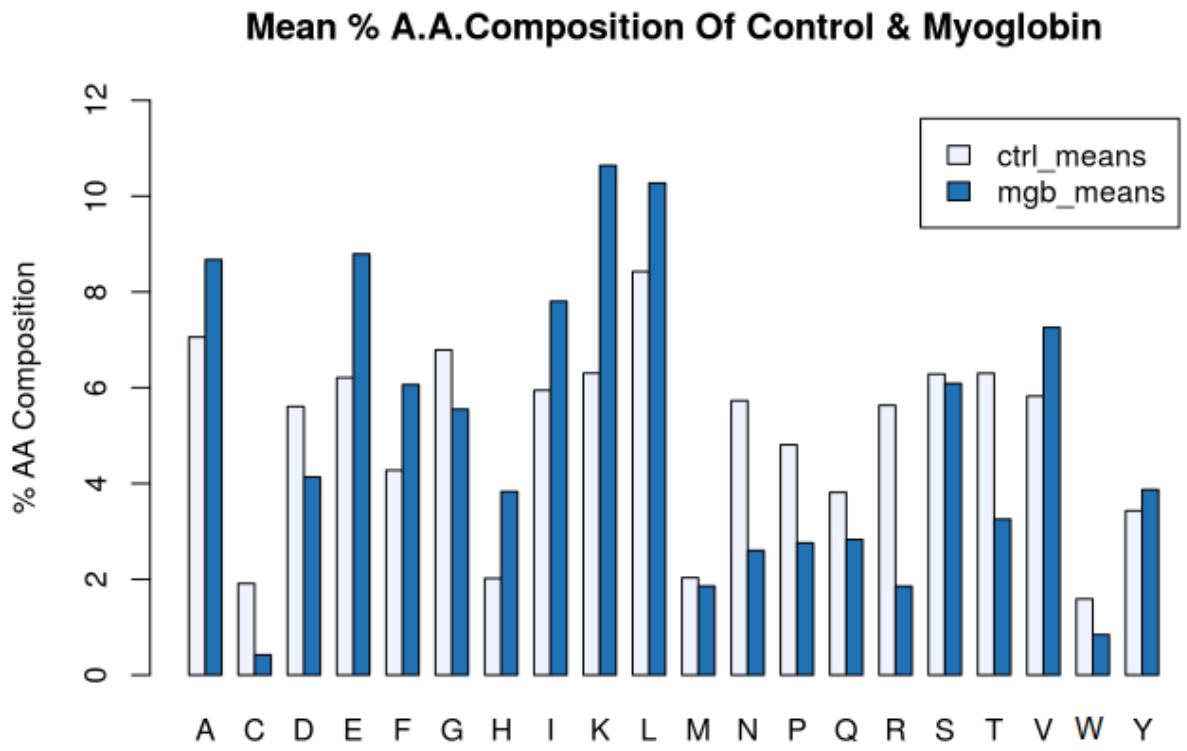


Figure 5: Mean Percent Amino Acid Compositions For Control And Myoglobin

Exploratory Data Analysis (EDA)

During EDA, the data is 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 necessary since Principal Component Analysis was a significant aspect of this experiment.

Testing determined earlier that three amino acids (C, F, I) from the single amino acid percent composition needs transformation 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 high points 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

Three transformations take place for this dataset.

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

All work uses R[25], RStudio[26] and a machine learning library/framework `caret`.[27]

Caret library for R

The R/caret library is attractive to use for many reasons. It currently allows 238 machine learning models that use different options and data structures.[28] The utility of caret is that it organizes the input and output into a standard format making the need for learning only one grammar and syntax. Caret also harmonizes the use of hyper-parameters. Work becomes reproducible. Setting up the training section for caret, for this experiment, can be broken into three parts.

Tuning Hyper-parameters

The `tune.grid` command set allows a researcher to experiment by varying the hyper-parameters of the given model to investigate optimum values. Currently, there are no algorithms that allow for the quick and robust tuning of parameters. Instead of searching, a sizable experimental space test searches an n-dimensional grid in a full factorial design if desired.

Although some models have many parameters, the most common one is to search along a cost hyper-parameter.

> "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."[29]

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

Example Cost Function: [30]

$$Cost \;=\; \Big(y_i - \hat{f}(x_i)\Big)^2 \tag{3}$$

It may be important to search the literature to determine if other researchers have used a specific range of optimum value, which may speed a search. For example, C.W. Hsu et al. suggest using a broad range of 20 orders of magnitude of powers of 2,[31]

[25] https://cran.r-project.org/
[26] https://rstudio.com/
[27] http://topepo.github.io/caret/index.html
[28] http://topepo.github.io/caret/available-models.html
[29] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, http://www.deeplearningbook.org, 2016
[30] Roberto Battiti and Mauro Brunato, The LION way. Machine Learning-Intelligent Optimization, LIONlab, University of Trento, Italy", 2017", http://intelligent-optimization.org/LIONbook
[31] Chih-Wei Hsu, et al., A Practical Guide to Support Vector Classification, 2016, http://www.csie.ntu.edu.tw/~cjlin

R Code: Using TuneGrid command to test sequences of number such that the optimal cost function can be sought.

```
# tuneGrid = svmLinearGrid
svmLinearGrid <- expand.grid(C = c(2^(seq(-5, 15, 2))))
```

`expand.grid` will produce a sequence of numbers (in our case) from $2^{-5}, 2^{-3}$ *to* 2^{15} to be tested as values for the cost function.

Then switch to 4 or 5 orders of magnitude with 1/4 log steps,

e.g. cost = `expand.grid(c(2^(seq(1, 5, 0.25))))`

for a finer search gird.

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.[32]

> "The samples are randomly partitioned into k sets of roughly equal size. A model is fit using all samples except the first subset (called the first fold). The held-out samples are used for prediction by the recent model. The performance estimate measures the accuracy of the *out of bag* or *held out* samples. The first subset is returned to the training set, and the 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."[33]

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

R Code: 10 Fold cross-validation repeated 5 times

```
fitControl <- trainControl(method = "repeatedcv",     # Type of Cross-Validation
                           number = 10,               # Number of splits
                           repeats = 5,               # Produce 5 replicates
                           savePredictions = "final") # Saves FP/FN predictions
```

Train command

The training command produces an object of the model. The first line should point out the "formula," which is modeled. The dependent variable is first. The ~ (Tilda sign) indicates a model is called. Then the desired features can be listed or abbreviated with the all (.) sign.

R Code: Train command

```
model_object <- train(Class ~ .,              # READ: Class is modeled by all features.
                      data = training_set,    # Data used
                      trControl = fitControl, # Cross Validation setup
                      method = "svmLinear",   # Use caret ML method
                      tune.Grid = Grid)       # Hyperparameter grid exploration
```

[32] https://machinelearningmastery.com/k-fold-cross-validation/
[33] Max Kuhn, Kjell Johnson, Applied Predictive Modeling, 2013, ISBN 978-1-4614-6848-6

Analysis of results

In binary classification, a two by two contingency table describes predicted versus actual value classifications. This table is also known as a confusion matrix for machine learning students. It is also known as a two by two contingency table in statistics.

2 x 2 Confusion Matrix	Actual = 0	Actual = 1
Predicted = 0	True-Negatives	False-Negatives
Predicted = 1	False-Positives	True-Positives

There are many ways to describe the results further using this confusion matrix. However, Accuracy is used for all comparisons.

$$Accuracy \ = \ \frac{TP + TN}{N_{Total}} \tag{4}$$

The second goal of this experiment is to produce the False Positives and False-Negatives and evaluating these by comparing them to the Principal Component Analysis Biplot of the first two Principal Components.