Compare Classification Performance of Multiple Machine Learning Methods on High Dimensional Genetics data

Ziyan Xia[1] and Yanlin Li[1]

[1] *Department of Statistics and Data Science, Carnegie Mellon University*
[1] *Department of Statistics and Data Science, Carnegie Mellon University*

Abstract

Detecting whether genes are small Open Reading Frames (smORFs) is important but difficult and costly process in biology experiments. It will be of great help if we could use machine learning classification models to predict whether genes are smORFs based on genes' features. We tested classification models including Random Forest (RF), iterative Random Forest (iRF) and XGBoost models on a genetic dataset from a online bioinformatics database and there are only a small portion of genes are smORFs in this dataset. After evaluating their predictive classification performance using ROC Curve, AUC and Precision, we found XGBoost models perform worst and iRF always performs better than RF in AUC. There's still future work to do regarding how to choose a appropriate criterion and oversampling methods for such an imbalanced dataset.

1 Introduction

Small Open Reading Frames (smORFs) are important sources of putative peptides previously dismissed as being non-functional or junk DNA. As detecting whether a gene is smORF or not is fairly difficult and costly, we would like to develop a classification model that could classified genes as whether or not smORF based on relevant features. Using the Flybase data (See Page 2-3), we designed an experiment to evaluate the prediction performance of these classification models including Random Forest, iterative Random Forest and XGBoost models. To develop a appropriate classification model, we are especially interested in the following questions:

Is the data appropriate to do classification performance comparison? Imbalanced data is always a big problem because quite a lot classification models in machine learning is under the assumption that classes are of equal weights. It is important for us to know whether the dataset is appropriate to do the classification performance comparison.

How do we get the classification results to compare? If we decide to use an imbalanced dataset to do model performance evaluation and comparison, how should we get the comparable results for each methods?

Except for classification performance, is there something else that we can compare these models for? Except for the predictive classification performance comparison, are other useful things that we can extract from these models to compare? If so, What is the meaning of compare them?

2 Data

FlyBase is an online bioinformatics database and the primary repository of genetic and molecular data for the insect family Drosophilidae(Thurmond, 2019). The data we used from FlyBase is a combination of RNAi data, ChIP-Seq data, RBP data, and smORF data of 14,006 genes from model organism Drosophila

	Drosophila Genetics Dataset from FlyBase				
Name	Gene id	RNAi Data	ChIP-seq Data	RBP Data	Is_smORF
Columns	1-2	3-117	118-467	468-487	488
Data Content	Name	Fold Change	Order Statistics	Fold Change	0 or 1

Table 1: Dataset Overview

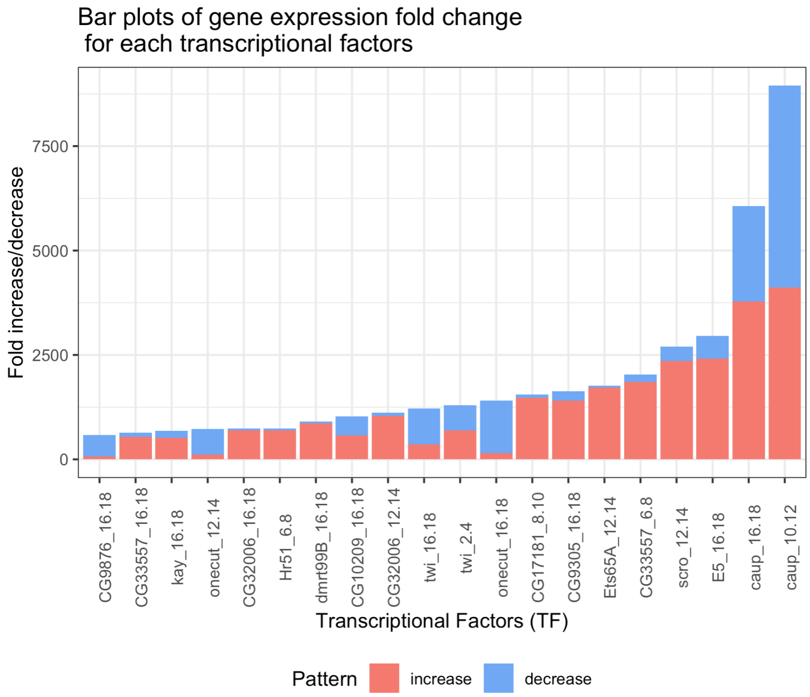


Figure 1: Bar plots of gene expression fold change for each transcriptional factor

in March 2020 (See Table 1). Now we introduce each part of data as follows:

RNAi Data:

In the RNA interference (RNAi) data of the full dataset, each row name represents each gene and each column name represent different stage of gene expression-controlling transcription factors. This data records the changes in gene expression measured by fold change after knocking down each transcription factor at different stages (a total of three). A value of 0 means no change; a positive value means an increase in gene expression; a negative value means a decrease in gene expression.

Figure 1 shows the increase and decrease of multiple changes of 36 transcription factors with the largest amount of gene expression multiple changes after RNA interference experiment under transcription control. Multiple changes are the ratio of the gene expression level before and after the deletion of transcription factors. It measures the change in the variable between the two. From Figure 1, transcriptional factors onecut and caup are noticeable. Most of the expression changes are decrease changes after knocking it down for onecut. The genes affected after knocking down caup are the most.

ChIP-seq Data:

ChIP-sequencing (ChIP-seq) is a method used to analyze protein interactions with DNA. Like many high-throughput sequencing methods, ChIP-seq generates a very large data set that requires appropriate computational and analytical methods to interpret useful information. To predict DNA binding sites from ChIP-seq data, peak detection is commonly used. Therefore, our ChIP-seq data selects the order statistics, that is, the peak value with the largest value.

RBP Data:

RNA-binding proteins (RBPs) are proteins that bind to the double or single stranded RNA in cells and participate in forming ribonucleoprotein complexes. Same as the RNAi data, the RBP data is a measurement of the gene expression level change before and after knocking down the corresponding RBP. (Wikipedia, RNA-binding proteins)

Whether smORF or Not:

Small Open Reading Frames (smORFs) are important sources of putative peptides previously dismissed as being non-functional or junk DNA, as determined by early gene prediction methods. (Guerra-Almeida, 2020) In our data, whether gene is or it not smORF is represented by 0 and 1, where 1 is a Small Open Reading Frame and 0 is not a Small Open Reading Frame.

3 Methods

Before introducing the detailed methods for each question, we first need to introduce the concepts of some sampling and machine learning methods we are using:

Wrapper for rapidly converging Gibbs algorithm (wRACOG): This algorithm is used for generating more samples of the minority class for a imbalanced dataset. It generates synthetic minority examples by approximating their probability distribution until sensitivity of wrapper over validation cannot be further improved.

Random Forest and iterative Random Forest: Random Forest (RF) is an ensemble of decision trees and its classification results are based on the majority voting results of these decision trees. Decision tree is a classifier and it has a tree-like structure in which each internal node represents a ”test” on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from root to leaf represent classification rules.

Iterative Random Forest (iRF) was first developed to search for high-order feature interactions, however, it is actually an optimized version of Random Forest and therefore can function well as a classifier. The iRF algorithm did iterative feature reweighting adaptively. It starts by equal weight and update the weight in each iteration according to the feature importance of the new model. Subsets of features with high importance are chosen in the iterations for efficiency.

XGBOOST: XGBoost is an implementation of gradient boosted decision trees (Decision tree has been introduced in the previous section) introduced by Tianqi Chen. Boosting is a method that can convert a set of weak learners into strong learners by using a bunch of simple trees to develop a complex model which performs better. The result of gradient boosting is a weighted sum of different trees. In each iteration, we greedily minimize the misclassification error by continuously choosing the split points. Observation weights are chosen according to its importance in achieving a correct classification. XGboost is one type of such boosting algorithm that takes a highly efficient use of compute time and memory resources. (Morde, 2019)

3.1 Is the data appropriate to do classification performance comparison?

To answer this question, we calculated the imbalance ratio of the response variable. Besides, after spliting the original dataset into 80% training set and 20% test set by random sampling, we trained Random Forest, iterative Random Forest training set. We also trained XGBoost model with training data. For XGBoost, we first specified a hyper parameter accounted for imbalance ratio and then did XGBoost cross validation to select the best iteration number. Then using the best iteration number and the hyper parameter accounted for imbalanced ratio, we ran the XGBoost model again with the same training data to get the final XGBoost model.

After training our models, we predicted the response by testing set. We made ROC Curve Plots and calculated the AUC and Precision of three models to evaluate whether this dataset is appropriate for performance comparison.

3.2 How do we get the classification results to compare?

After getting the iterative Random Forest result from Question 1, we were able to extract importance of the features from it. To reduce computation cost, we selected 70 most important predictors as our new features. To solve the imbalance problem for training set, we did oversampling to the minority class for the training set using oversampling technique wRACOG to generate an oversampled data. By oversampling, we increased the ratio of the minority class of the response variables for the training set to 13%. We couldn't generate more samples due to the limitation of the algorithm.

We repeated the model fitting and predicting process in Question 1 for the oversampled data and also made ROC Curve Plots and calculated the AUC and Precision of three models for it. This time, together with the results from Question 1's model, we compared the performance of classification modelling of all four datasets together.

3.3 Except for classification performance, is there something else that we can compare these models for?

As iterative Random Forest is an optimized version of Random Forest and we can extract importance of features from both Random Forest and XGBoost models, we are able to compare the feature of importance they generated. For extracting importance, the imbalance of the response variables doesn't affect much and the feature importance of iRF is an optimized and more converged version of feature importance of Random Forest as mentioned before. Therefore we used the feature importance extracted from the iterative Random Forest and XGBoost models using the original dataset with all features to compare.

4 Results

Before interpreting the results, we first need to introduce the concepts of criteria used for model performance evaluation:

$$Sensitivity = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

We can use our response variable to interpret these criteria:

In our experiment, Sensitivity evaluates how many real smORFs are predicted as smORFs; Specificity evaluates how many real non-smORFs are predicted as non-smORFs; Precision evaluates how many genes that are predicted as smORFs are real smORFs.

ROC is a curve with Specificity (False positive rate) on the x-axis and Sensitivity (True positive rate) on the y-axis with positive class as the one with less cases. AUC is the area under the ROC curve. The range of AUC value is 0 to 1. Here are the possible inferences we can get from the AUC values and ROC curves:

1. $AUC = 0$: The prediction is 100% wrong, which also means you can get a 100% correct prediction if you reverse every prediction.

2. $0 < AUC < 0.5$: It ranks a random positive example higher than a random negative example less than 50% of the time. The model performs worse than random guessing.

3. $AUC = 0.5$: The model is worthless. Its ability of prediction is no better than random guessing.

4. $0.5 < AUC < 1$: It ranks a random positive example higher than a random negative example more than 50% of the time. Most classification models fall in this range. The prediction power of a model can be predicted by how much its AUC value exceeds 0.5.

5. $AUC = 1$: The prediction is 100% correct.

(Narkhede, 2019)

4.1 Is the data appropriate to do classification performance comparison?

The imbalanced ratio for the original dataset is 0.056 and the imbalance ratio for the training set in Question 1 is 0.054, which means only around 5% of the response variables is of class 1 (See Figure 2). Such imbalanced data will pose a challenge for predictive classification modeling as most models used for classification were designed under the assumption of an equal number of examples for each class. The evaluation results verified our assumption: The XGBoost prediciton results are just as bad as random guess based on AUCs.

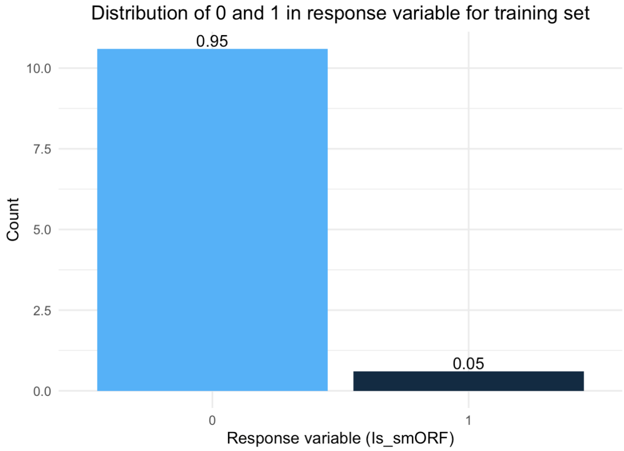


Figure 2: Bar plots of the distribution of 0 and 1 in response variable for training set

4.2 How do we get the classification results to compare?

As mentioned before, to reduce computation cost, we only selected 70 most important features from the iterative Random Forest fitted on the full dataset.

	iRF	RF	XGBoost	Imbalance Ratio
Original Data	0.65	0.63	0.50	0.06
Reduced Data	0.64	0.62	0.50	0.06
Reduced Oversampled Data	0.63	0.62	0.50	0.13

Table 2: AUC for different models and data

	iRF	RF	XGBoost	Imbalance Ratio
Original Data	0.37	0.50	N/A	0.06
Reduced Data	0.14	0.22	N/A	0.06
Reduced Oversampled Data	0.42	0.43	N/A	0.13

Table 3: Precision for different models and data

The AUC values for different models and datasets are shown in Table 2 and The ROC curves used to calculate AUC are for the models are shown in Figure 3. The area under curves corresponds to the results in Table 2.

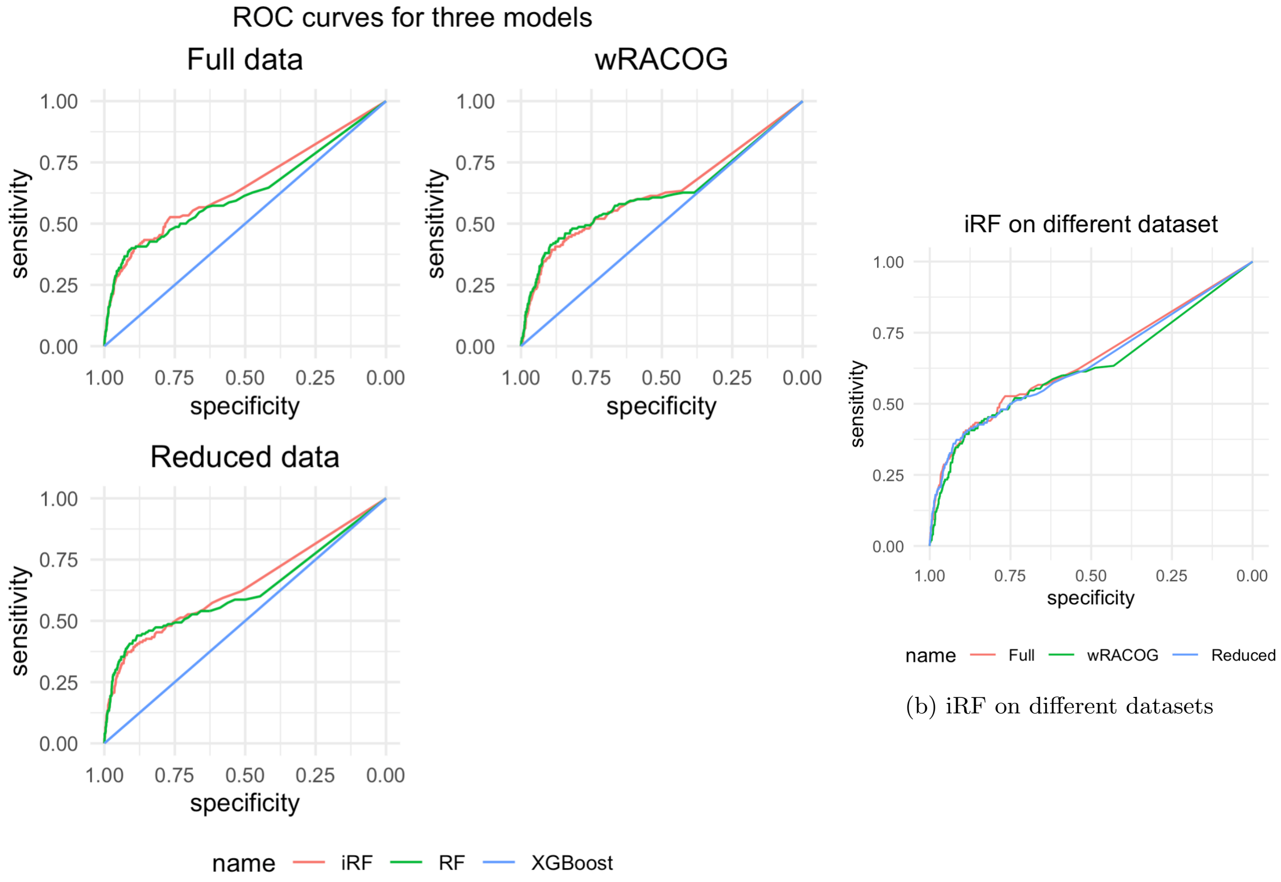
1. XGBoost models failed all three dataset no matter what imbalance ratio they have as AUCs for XGBoost models are always 0.5, which is no better than random guessing.

2. The AUCs of the original data is slightly higher than those of reduced dataset for both iRF and RF and the AUCs of the reduced data is slightly higher than those of the reduced oversampled data for both iRF and RF.
3. Overall, the AUCs for iRF are the highest out of all three models and are always higher for predictive classification than RF.

The Precisions for different models and data are shown in Table 3. Results in Table 3 contradict the results in Table 2 a lot.

1. Again, XGBoost failed all datasets and we couldn't calculate its Precisions as it classified every gene as non-smORF.
2. The original data has the highest Precision for RF but doesn't have the highest Precision for iRF.
3. The reduced oversampled data has much higher Precisions for both iRF and RF. Its Precision of iRF is the highest among all three datasets.

The numbers of true positive and the false positive used to calculate Precisions are shown in Table 4.



(a) Three models on the four datasets

(b) iRF on different datasets

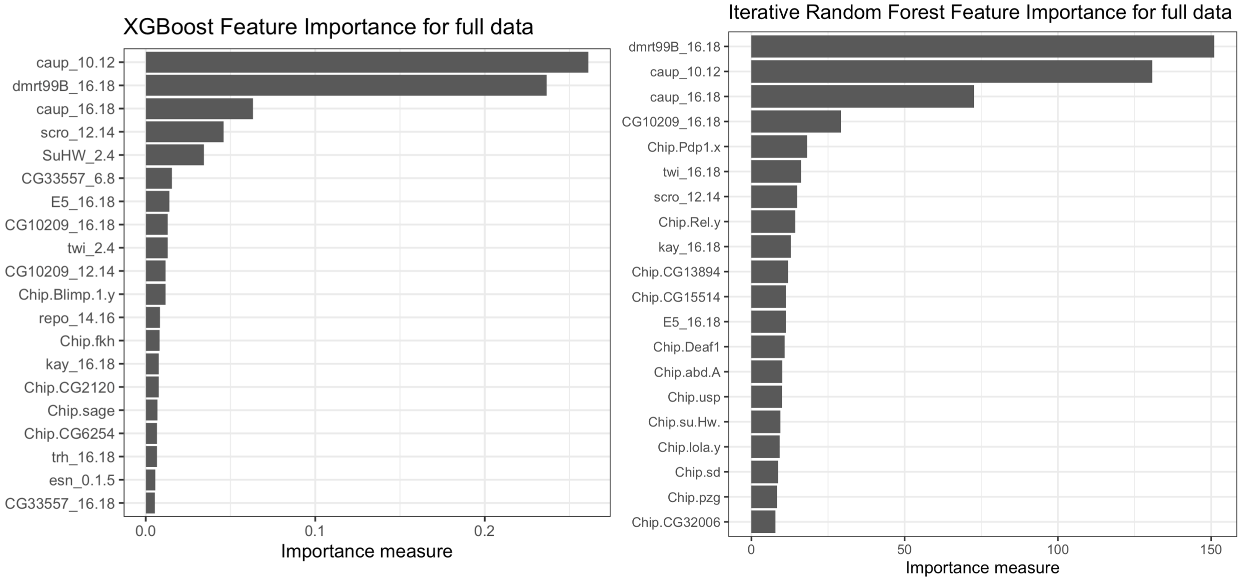
Figure 3: ROC Curves

	True Positive	False Positive
iRF	19	33
RF	4	4

Table 4: Part of the Confusion Matrix for iRF and RF on the original data

4.3 Except for classification performance, is there something else that we can compare these models for?

We compared the feature importance extracted from the model output of iterative Random Forest and XGBoost Models that were fitted on the full dataset. Here we didn't include the feature importance extracted from Random Forest because the feature importance of iRF is usually a optimized and probably optimized version of the importance from Random Forest. Comparison of feature importance is shown in Figure 4. From Figure 4, we find that there is considerable overlap between the 20 most important features selected by XGBoost and iterative Random Forest and the important orders for two models are also quite similar for the top 5.



(a) XGBoost

(b) iterative Random Forest

Figure 4: Feature Importance

5 Discussion

After evaluating the predictive classification performance of all the models and datasets above, we have some really interesting findings to discuss about.

5.1 Is the data appropriate to do classification performance comparison?

Based on the fact that the XGBoost always predicts all the response class as 0 and has a 0.5 AUC for the test sets, we can conclude that XGBoost method failed an imbalanced dataset like this. However, we can still compare the predictive classification performance of iterative Random Forest and Random Forest.

5.2 How do we get the classification results to compare?

We divide our discussion for this question into two parts:

5.2.1 Out of all three models, which model performs the best?

If we use AUC as the criterion, the model that performs best is iterative Random Forest. However, if we use Precision as the criterion, the model that performs the best is Random Forest, which violates our assumption that iterative Random Forest is a optimized version of Random Forest.

5.2.2 Does oversampling the minority class observations improve the model performance?

From both AUC and Precision results in Table 2 and Table 3, oversampling is not helpful for XGBoost models and it failed all the situations. If we use AUC as the criterion and compare the performance of the reduced data and the reduced oversampled data, it doesn't improve the performance of both iRF and RF. If we use Precision as the criterion and compare the performance of the reduced data and the reduced oversampled data, it improves the performance of both models a lot.

In summary, two criteria give the opposite answers for two questions above. However, from Table 4, the reason that Precision of RF is a lot higher than iRF is that the total number of genes that are predicted as smORFs in RF is a lot lower than iRF. Therefore, although Precision is usually considered as a better way to evaluate classification on imbalanced dataset than AUC, it doesn't work well here. There should be future work in finding a better criterion.

5.3 Except for classification performance, is there something else that we can compare these models for?

As mentioned before, there is a considerable overlap of the most important features selected by both iRF and XGBoost models, which means both models reach a consensus on what features are important for predicting whether gene is a smORF. These important features here may be very useful in future smORF detecting.

5.4 Weaknesses and Future Work

1. Our data is extremely imbalanced, even after oversampling. So it is hard to draw a conclusion on how well the models actually do on high dimensional genomics data.

2. Except the three models we used in the analysis, there may be other models that can do better in prediction. For the next step, we can try models such as generalized additive model, neural network, and support vector machine.

3. We have not yet found any methods that can improve the performance of XGBoost. Given that XGBoost is a highly efficient model, we may try other methods in the future.

4. There should be future work in finding a better criterion to compare model performance on imbalanced datasets.

5. Limited by the lack of genomics knowledge, we cannot draw any useful genomics inference from our model. In the future, we can collaborate with biologist and give more useful conclusions.

References

[1] Thurmond, Jim; Goodman, Joshua L, (8 January 2019). *"FlyBase 2.0: the next generation"*. Nucleic Acids Research 47 (D1): D759–D765. doi:10.1093/nar/gky1003.

[2] Park P. (2009) *Chip–seq: advantages and challenges of a maturing technology.* Nat Rev Genet, 10:669–680. https://doi.org/10.1038/nrg2641.

[3] Guerra-Almeida, D., amp; Nunes-da-Fonseca, R. (2020). *Small open reading frames: How important are they for molecular evolution?* Frontiers in Genetics,11. https://doi.org/10.3389/fgene.2020.574737

[4] B. Das, N. C. Krishnan and D. J. Cook, "wRACOG: A Gibbs Sampling-Based Oversampling Technique 2013 IEEE 13th International Conference on Data Mining, 2013, pp. 111-120, doi: 10.1109/ICDM.2013.18.

[5] Basu, S., Kumbier, K., Brown, J. B., amp; Yu, B. (2018). *Iterative random forests to discover predictive and stable high-order interactions.* Proceedings of the National Academy of Sciences, 115(8), 1943–1948. https://doi.org/10.1073/pnas.1711236115

[6] Morde, V. (April 8, 2019). *XGBoost algorithm: Long may she reign!* Medium. Retrieved December 11, 2021, from https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d.

[7] Narkhede, S. (May 26, 2019). *Understanding AUC - ROC Curve.* Retrieved December 11, 2021, from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5