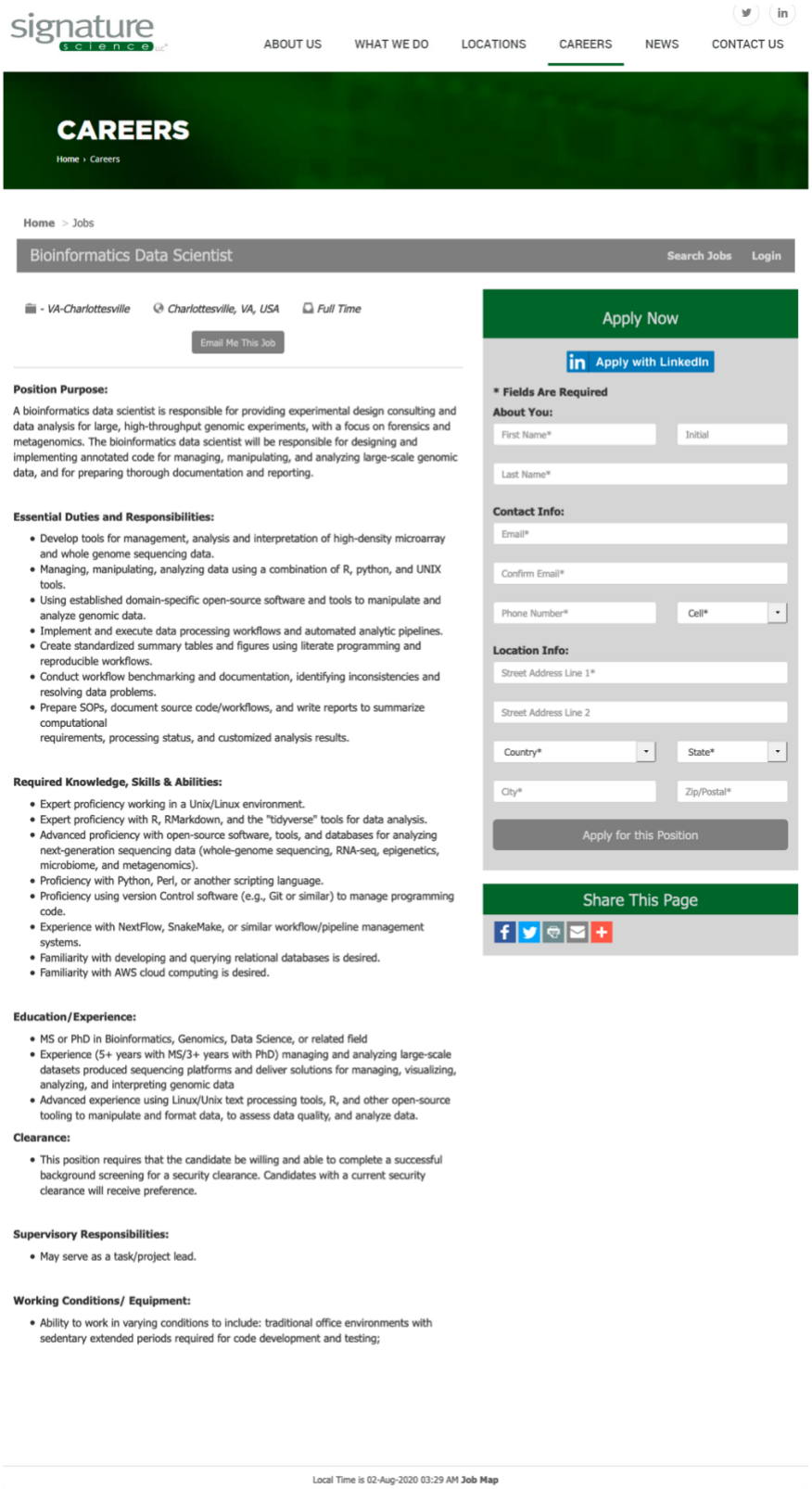
Case Studies Fortnightly Task 1

Stephanie Choo s3846487

Part 1: Job Role



signature science

ABOUT US | WHAT WE DO | LOCATIONS | CAREERS | NEWS | CONTACT US

CAREERS

Home › Careers

Home > Jobs

Bioinformatics Data Scientist — Search Jobs — Login

- VA-Charlottesville — Charlottesville, VA, USA — Full Time

Email Me This Job

Position Purpose:

A bioinformatics data scientist is responsible for providing experimental design consulting and data analysis for large, high-throughput genomic experiments, with a focus on forensics and metagenomics. The bioinformatics data scientist will be responsible for designing and implementing annotated code for managing, manipulating, and analyzing large-scale genomic data, and for preparing thorough documentation and reporting.

Essential Duties and Responsibilities:

- Develop tools for management, analysis and interpretation of high-density microarray and whole genome sequencing data.
- Managing, manipulating, analyzing data using a combination of R, python, and UNIX tools.
- Using established domain-specific open-source software and tools to manipulate and analyze genomic data.
- Implement and execute data processing workflows and automated analytic pipelines.
- Create standardized summary tables and figures using literate programming and reproducible workflows.
- Conduct workflow benchmarking and documentation, identifying inconsistencies and resolving data problems.
- Prepare SOPs, document source code/workflows, and write reports to summarize computational
requirements, processing status, and customized analysis results.

Required Knowledge, Skills & Abilities:

- Expert proficiency working in a Unix/Linux environment.
- Expert proficiency with R, RMarkdown, and the "tidyverse" tools for data analysis.
- Advanced proficiency with open-source software, tools, and databases for analyzing next-generation sequencing data (whole-genome sequencing, RNA-seq, epigenetics, microbiome, and metagenomics).
- Proficiency with Python, Perl, or another scripting language.
- Proficiency using version Control software (e.g., Git or similar) to manage programming code.
- Experience with NextFlow, SnakeMake, or similar workflow/pipeline management systems.
- Familiarity with developing and querying relational databases is desired.
- Familiarity with AWS cloud computing is desired.

Education/Experience:

- MS or PhD in Bioinformatics, Genomics, Data Science, or related field
- Experience (5+ years with MS/3+ years with PhD) managing and analyzing large-scale datasets produced sequencing platforms and deliver solutions for managing, visualizing, analyzing, and interpreting genomic data
- Advanced experience using Linux/Unix text processing tools, R, and other open-source tooling to manipulate and format data, to assess data quality, and analyze data.

Clearance:

- This position requires that the candidate be willing and able to complete a successful background screening for a security clearance. Candidates with a current security clearance will receive preference.

Supervisory Responsibilities:

- May serve as a task/project lead.

Working Conditions/ Equipment:

- Ability to work in varying conditions to include: traditional office environments with sedentary extended periods required for code development and testing;

Apply Now

Apply with LinkedIn

* Fields Are Required

About You:

First Name* | Initial

Last Name*

Contact Info:

Email*

Confirm Email*

Phone Number* | Cell*

Location Info:

Street Address Line 1*

Street Address Line 2

Country* | State*

City* | Zip/Postal*

Apply for this Position

Share This Page

Local Time is 02-Aug-2020 03:29 AM **Job Map**

The role is a bioinformatics data scientist at Signature Science. This role relates into the health and bioinformatics fields.
https://signaturescience.plansource.com/jobs/153375.html?utm_source=Indeed&utm_medium=organic&utm_campaign=Indeed

This role is looking for both finding insights and to make predictions. This role Is primarily concerned with gaining insights into the data. Gaining insights into the genetic basis of genetic conditions is important in the health field in order to make further advancements in medicine.

Part 2: Data Set

The chosen data set is the Mice Protein Expression Data Set from the UCI Machine Learning Repository. (https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression) [1]. This data set is assessing the This dataset involves the data from 77 proteins that produced signals in the nuclear fraction of the cortex. The mice are in 8 classes which are defined by their genotype (c=control/t=trisomy), behaviour (CS=stimulated to learn/ SC= not stimulated to learn) and injection treatment (s=saline/ m=memantine).

I chose this dataset because often genetic experiments are conducted on genetically mutated mice or animals. In addition, protein expression and protein modifications are indicative of the what is happening both at the genetic level and the effect that the treatment is having on the mice. Therefore, the skills demonstrated within my analysis are transferrable to this job, especially when the analysis and modelling were all done in python.

Part 3: Experiment with Machine Learning Models.

This experiment aimed to classify the protein expressions of 77 genes into 8 classes. Also, the aim was to determine which proteins were the most significant to determined which mouse belonged to which class. Before running the models, there were a number of null values. Each were filled by the average protein expression of that protein for that class of mouse. This resulted in Figure 1.

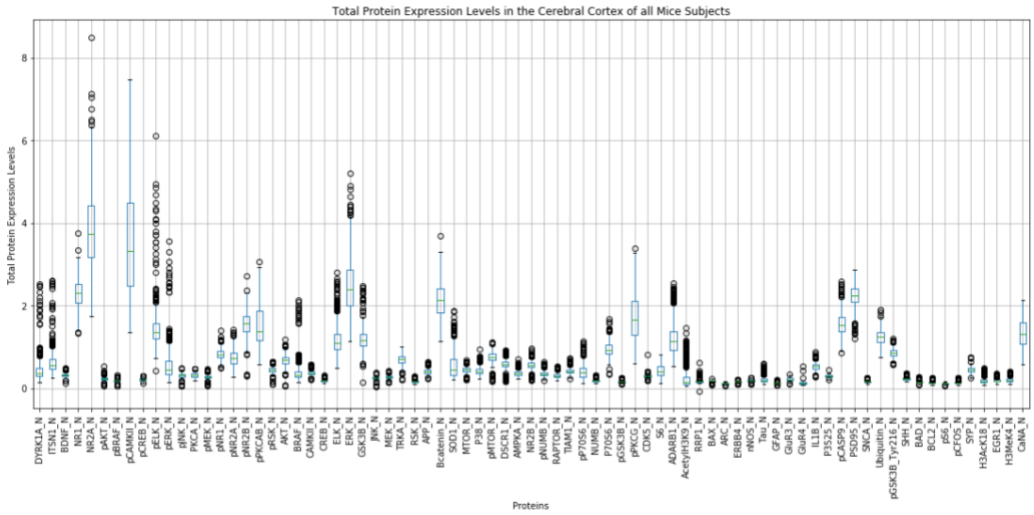


Fig 1.) Total Protein Expression Levels in Cerebral Cortex of all mice

For feature selection I had performed the Hill Climbing technique[2]. This was to extract only the most essential proteins to classify the mice. In total only 47 proteins were selected for the most optimal performance. Therefore, modelling was performed on a subset of the data consisting of the 47 proteins.

Both decision tree and random forest were used to classify the mice based on their protein expressions. Both decision tree and random forest use algorithms to classify using a tree representation of a series of decisions. In decision tree initially all observations are not yet classified at the beginning. However, the algorithm determines which decision makes the most homogenous split and splits the group of observations. The decision tree algorithm keeps on doing this for each group until it makes predications for each mouse [3]. A similar process occurs in random forest, except that multiple relatively uncorrelated trees are produced[4]. In random forest multiple decision trees are produced which are trained on different parts of data and used different features before averaging the result. [4]

For each model the process was to split the data into test and training (30% training data). This not only ensures that the models can learn from the data sufficiently before they are trained. Each model had a confusion matrix, classification report produced.

Then each model was validated by K-folds cross validation. This tests the model on different combinations of train and test data. Parameters were tuned using GridSeachCV and Stratified sampling [6]to ensure about an even number of observations are samples in each train-test split. Also, I've used GridSearchCV [5]to test many combinations of parameters for each model. The confusion matrix is an evaluation of classification accuracy. It also takes into consideration the following metrics:

- Precision: The fraction of how many of the predictions were correct, false positive rate.
- Recall: The fraction of how many observations had been correctly labelled, false negative rate.
- F1-score: the harmonic mean of precision and recall. (precision* recall)/precision + recall).

For each of the above they range from 0 (highly inaccurate) to 1 (most accurate). For precision a high value means a low false positive rate, and for recall a high value means a low false negative rate [7]. The correctly predicted observations lie along the diagonal, whilst the false positive and false negatives lie on either side of it. Also, each model was assessed with the precision-recall curve which gives the ratio between precision and recall, giving an indication of the true positive classification rate [7].

After I had finalised the parameters of each model, I had plotted bar graphs of how each model, decision tree and random forest, ranked each of the proteins.

Results and Discussion:

Presented below are the decision tree classifier's classification report, confusion matrix and precision recall curve. This model had a high f1-score for all classes, although for most classes indicating a high ability to correctly classify mice based on protein expression. This model also had an average k-folds cross validation score of 0.89.

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
[[30  3  0  0  4  0  0  0]
 [ 2 32  0  0  2  3  0  0]
 [ 2  0 36  2  1  2  0  0]
 [ 0  0  0 41  0  0  3  0]
 [ 1  9  0  0 38  1  0  1]
 [ 2  0  0  0  0 32  0  0]
 [ 0  0  1  1  0  0 33  0]
 [ 0  0  1  0  0  0  1 40]]
              precision    recall  f1-score   support

      c-CS-m       0.81      0.81      0.81        37
      c-CS-s       0.73      0.82      0.77        39
      c-SC-m       0.95      0.84      0.89        43
      c-SC-s       0.93      0.93      0.93        44
      t-CS-m       0.84      0.76      0.80        50
      t-CS-s       0.84      0.94      0.89        34
      t-SC-m       0.89      0.94      0.92        35
      t-SC-s       0.98      0.95      0.96        42

    accuracy                           0.87       324
   macro avg       0.87      0.87      0.87       324
weighted avg       0.87      0.87      0.87       324
```

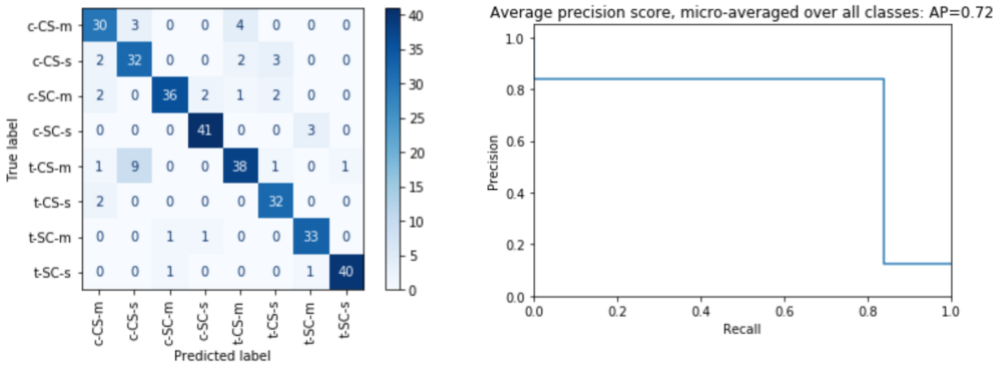


Figure 2. Decision Tree Classifier

Compared to decision tree, the random forest classifier had an increased precision, recall, f1-score for a number for all classes. In this model I had set the number of trees to 1000 in this case. There was almost 0.92 to 1.00 precision, recall and f1-scores. It perfectly predicted the c-SC-m (control mouse stimulated to learn with memantine treatment), t-SC-m and t-CS-s (2 classes of trisomic mice with different behaviours and treatments). In addition, the precision recall curve only started to fall around the 0.90 precision-recall ratio. That along with the confusion matrix, indicate that most of the mice were correctly classified based on their protein expression. At first there was a risk of the model learning the training data tool well, and that it may not generalise well to future genetic data from control and trisomy mice. However, attempts to prevent this had resulted in significant decreases in precision, recallf1-score and k-folds cross validation. This means that alterations to the random forest model would mean that it won't generalise to future genetic data for control and trisomy mice. Therefore, I chose the below random forest model. The high performance compared to decision tree can be attributed to how the random forest model functions. Since random forest averages multiple trees which are trained on parts of the training set to reduce variance and overfitting [4], it is more likely to produce a more accurate model than one decision tree.

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=15, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=1000,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
[[37  0  0  0  0  0  0  0]
 [ 2 37  0  0  0  0  0  0]
 [ 0  0 43  0  0  0  0  0]
 [ 0  0  0 44  0  0  0  0]
 [ 1  3  0  0 46  0  0  0]
 [ 0  0  0  0  0 34  0  0]
 [ 0  0  2  0  0  0 33  0]
 [ 0  0  0  0  0  0  0 42]]
              precision    recall  f1-score   support

      c-CS-m       0.93      1.00      0.96        37
      c-CS-s       0.93      0.95      0.94        39
      c-SC-m       0.96      1.00      0.98        43
      c-SC-s       1.00      1.00      1.00        44
      t-CS-m       1.00      0.92      0.96        50
      t-CS-s       1.00      1.00      1.00        34
      t-SC-m       1.00      0.94      0.97        35
      t-SC-s       1.00      1.00      1.00        42

    accuracy                           0.98       324
   macro avg       0.98      0.98      0.98       324
weighted avg       0.98      0.98      0.98       324
```

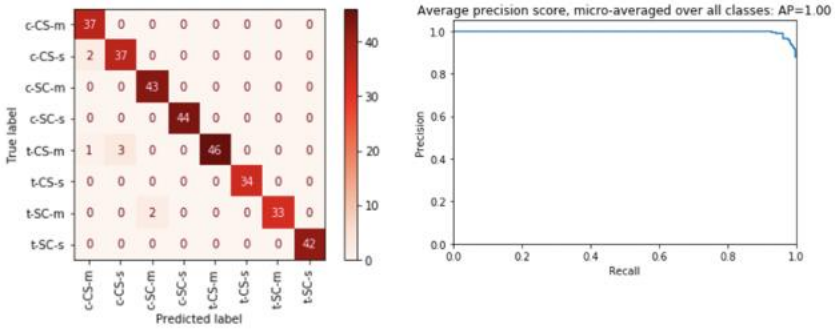


Figure 3. Random Forest Classifier

In addition to classification, each model has ranked each of the proteins by how importance they were in classifying the mice into their classes, which can be viewed in the below figure. The below figure shows that both decision tree and random forest had ranked SOD1_N, pPKCG_N and APP_N as the most important proteins to consider which class the mice belonged in. This indicates that SOD1_N, pPKCG_N and APP_N are the most important proteins to consider when studying the combined interactions of Down Syndrome, learning and memantine treatment. However, the ranking of the other proteins does vary. This indicates that certain proteins may be only linked with changes in genotype, behaviour or treatment, but not necessarily in combination. Therefore, I suggest further modelling studies in random forest classifiers to be built for each of the following: genotype, behaviour and treatment. This is to see if certain proteins are working alone in determining genotype, behaviour or the effect of treatment.

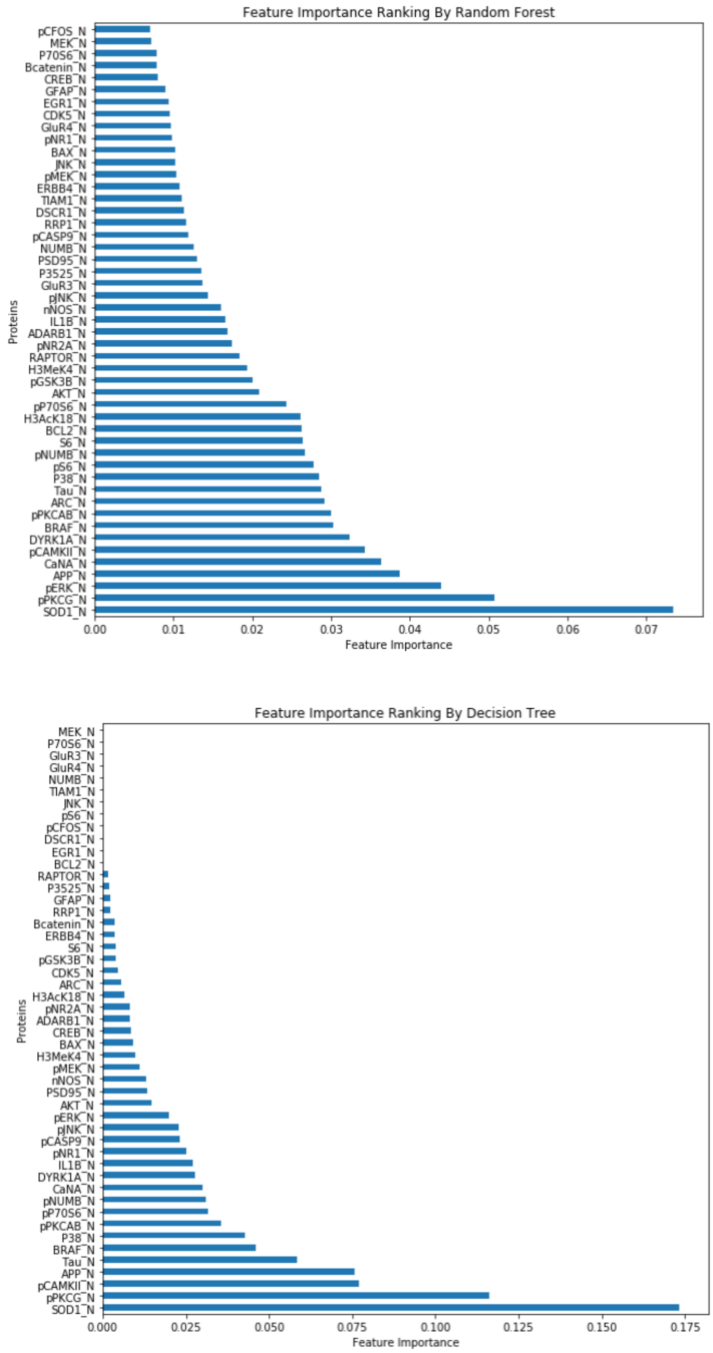


Figure 4. Protein's feature importance ranked by decision tree and random forest alogrithms.

References:

[1]D. Dua and C. Graff, "UCI Machine Learning Repository: Mice Protein Expression Data Set", Archive.ics.uci.edu, 2019. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression. [Accessed: 21- Jul- 2020].

[2]"Introduction to Hill Climbing | Artificial Intelligence - GeeksforGeeks", GeeksforGeeks, 2017. [Online]. Available: https://www.geeksforgeeks.org/introduction-hill-climbing-artificial-intelligence/. [Accessed: 02- Aug- 2020].

[3]J. Mount, "Why do Decision Trees Work?", R-bloggers, 2017. [Online]. Available: https://www.r-bloggers.com/why-do-decision-trees-work/. [Accessed: 02- Aug- 2020].

[4]T. Yiu, "Understanding Random Forest", Medium, 2019. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2. [Accessed: 02- Aug- 2020].

[5]"sklearn.model_selection.GridSearchCV — scikit-learn 0.23.1 documentation", Scikit-learn.org, 2007. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. [Accessed: 03- Aug- 2020].

[6]"sklearn.model_selection.StratifiedKFold — scikit-learn 0.23.1 documentation", Scikit-learn.org, 2007. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html. [Accessed: 03- Aug- 2020].

[7]"Precision-Recall — scikit-learn 0.23.1 documentation", Scikit-learn.org, 2007. [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html. [Accessed: 03- Aug- 2020].