**Data Mining Portfolio Entry for Final (Frank Liu)**

I will choose to have a **textual DMP** rather than making a presentation because I believe I am familiar with doing analysis (and probably do more analysis) with writing. I will structure the following DMP into Classification, Clustering, Association Analysis, and Anomaly Detection to have it better organized.

More focus will be on anomaly detection since I have done significant extra coding on other sections’ DMP, but there will be some new data and new analysis for them as well.

**Summary Table**

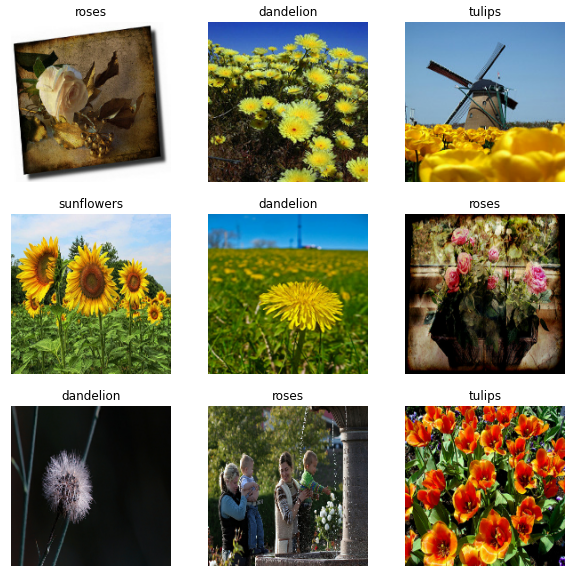
Summary table is attached below.

[Frank\_Liu\_Final\_DMP\_Data\_Coding\_Package\_Table](https://docs.google.com/spreadsheets/d/1ghbbPC0383bNX1dhI25z51ZOZqahBtPNgNf8eUjKjZI/edit#gid=0)

**Data Acquisition**

(This is a concise list, more description can be found in summary table)

New data for final DMP:

* **Flower Photo**
  + This is an image dataset that will be used to test the neural network image classifier in classification part
  + It has around 3670 photos of flowers that can be classified into 5 different categories, including daisy, dandelion, roses, sunflowers, and tulips
  + Some example picture
  + 
* **Marketing Campaign**
  + This is a dataset that contains customer information such as education, income, behavioral data, and purchasing preferences

| **Name** | **Type** | **Comments** |
| --- | --- | --- |
| Education | Nominal | Useful |
| Income | Numerical | Useful |
| Num\_Kid\_Home | Numerical | Useful |
| … | … | … |

* + Company can utilize this to segment customers (clustering) into similar groups to better design their product for different needs
* **Market Basket Data**
  + This is a dataset that contains transactions of products that similar to data we use in association analysis, but different one
* **Insurance Data**
  + This is a dataset that contains demographic information and charing cost of the insurance of a patient, and we will help insurance companies to identified some anomalies in the data to adjust the price of the charing

| **Name** | **Type** | **Comments** |
| --- | --- | --- |
| Sex | Nominal | Useful |
| BMI | Numerical | Useful |
| Charges | Numerical | Useful |
| … | … | … |

* **Whitewine Data**
  + This is a dataset that contains different chemical components of a wine to see how a wine is made and its characteristics. I will use it to do anomaly detection analysis.

| **Name** | **Type** | **Comments** |
| --- | --- | --- |
| Fixed acidity | Numerical | Useful |
| Density | Numerical | Useful |
| pH | Numerical | Useful |
| … | … | … |

Old data:

* Cancer data we use for classification (benign or malignant)

**Code Development / Program Development**

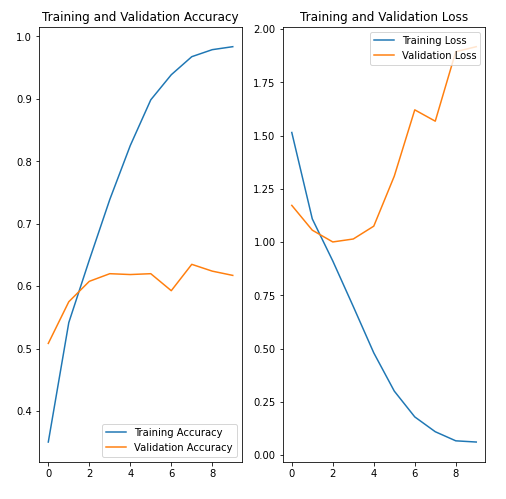
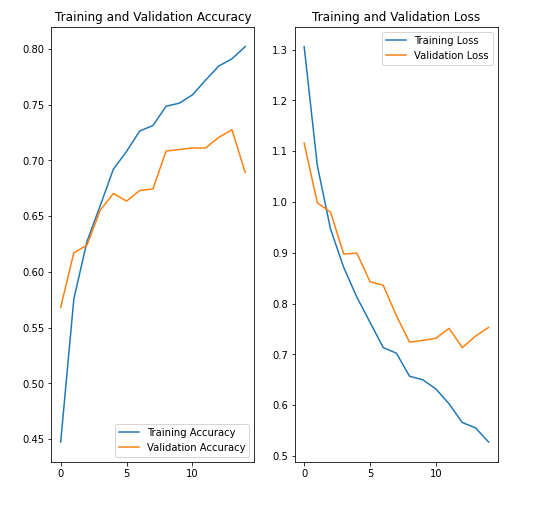
**Classification**

Link to previous DMP: [DMP\_Classification\_Frank\_Liu](https://docs.google.com/document/d/1ILPeY_qFfMRo717A2R8VhuvDKC7DmctZJkZIunRIYgE/edit)

In my previous DMP, I built a decision tree and bayes classifier from scratch, and used decision tree packages to play around with the hyperparameters and visualize them, and also using KNN package to do more analysis.

**Package Use (Tensorflow)**

Code:<https://colab.research.google.com/drive/1b1TClFyuJX_PqWbErmH0YYr2R4zWjKsf#scrollTo=EHZHLdBUb7hq>

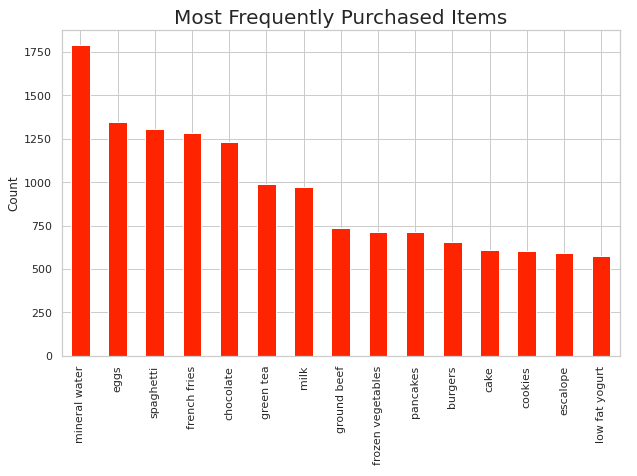
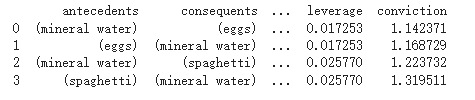
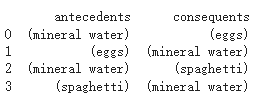
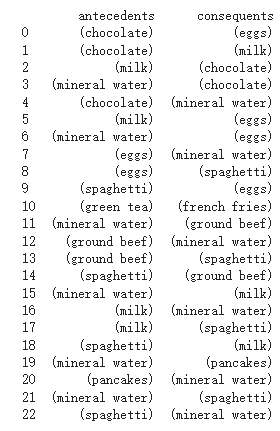
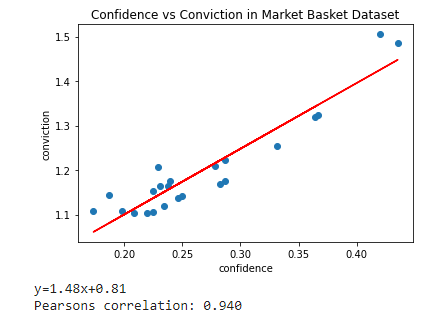
* In the final DMP, I will implement a **classification model using a deep neural network** with the help of a popular deep learning framework — TensorFlow. I reference this from: <https://www.tensorflow.org/tutorials/images/classification>
  + Specifically this will include the most state-of-the-art image classifier algorithm, CNN (convolutional neural network)
* The power of DNN is that it can **process high dimensional data**, and one of such is image. So this will be an image classification neural network model.
* One of the issue of dealing with image data is it is often with very high dimensional
  + Each picture data has (180, 180, 3) as the shape of input data, which is 180 x 180 pixels and 3 collar channels
  + To be able to feed this into the model, we need to **flatten the image to make it a vector** first.
* It is also important to normalize the data, and there are research papers published that prove mathematically that **deep neural networks will converge faster with normalized data, it will also prevent diminishing gradients during activation**.
* The structure of the model contains 3 layers of CNN with max pooling and 1 dense output layer with ReLu activation function
* After training, the result is plotted here:
* 
* The performance of the model on validation set is only around 60%, which means the structure of the model or other parameters still have much room for improvement, and some techniques include **regularization, batch training, hyperparameter tuning**. I plan to use those techniques to improve the performance but one single training is around 30 minutes due to the limited computing power of my laptop, so I will focus on evaluating the issue of model overfitting here.
* We can see the **training accuracy increase significantly over time but testing accuracy stops improving at around 60%**. The whole loss of the testing set is increasing. When testing accuracy is way lower than training accuracy, it is a clear sign of **model overfitting** (where the model gets overly complicated and fails to generalize).
* Generally, to reduce overfitting, we can use more data, build better architectures, but in deep learning, there is a method called **dropout**.
* Dropout randomly drops out some neurons from the layer during training so it only uses part of the model to train, which makes it not rely on too many details and improves the chance to generalize by decreasing the variance of prediction.
* After using randomly dropout 20% of the output neurons, I retrained the model
* 
* Now, we can see the training accuracy and testing accuracy pretty much follows the same pattern, and losses are decreasing simultaneously. Even though **training accuracy is 80% around (lower than before), the validation accuracy is improving to 75%. This is a good indication of a generalizable model.**
* CNN only applies to image data due it it’s powerful design, so I am not able to use it in old data (only numeric values). I can build all “fully-connected” layers of DNN to do that, but that is not worth the time.

**Association Analysis**

Link to previous DMP: [DMP\_Entry\_Association\_Frank](https://docs.google.com/document/d/1WEnbPZav4SYM9DqFtoRqWdIov8bRnlg1kywxi2pZGm8/edit)

In my previous association analysis, I implemented the Apriori Algorithm using set and dictionary, analyze the performance and limitation of the algorithms, then further apply mlxtend to different datasets.

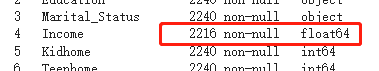
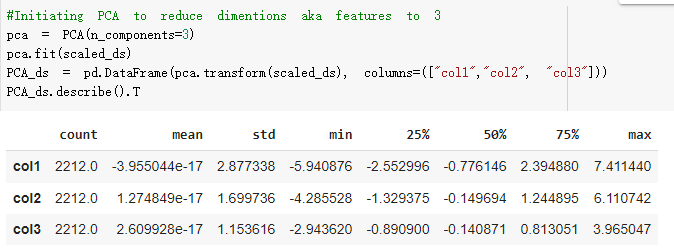
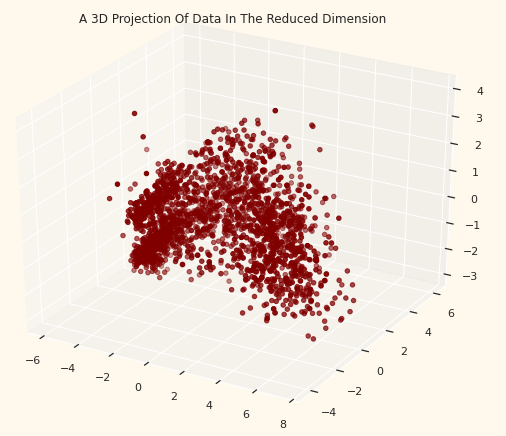
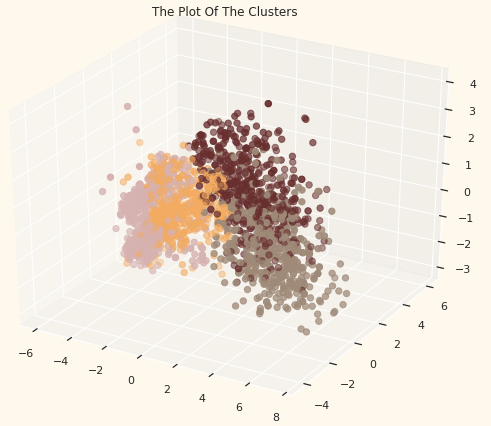
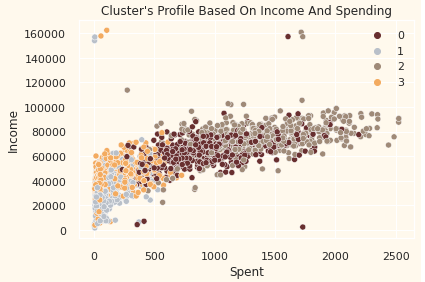
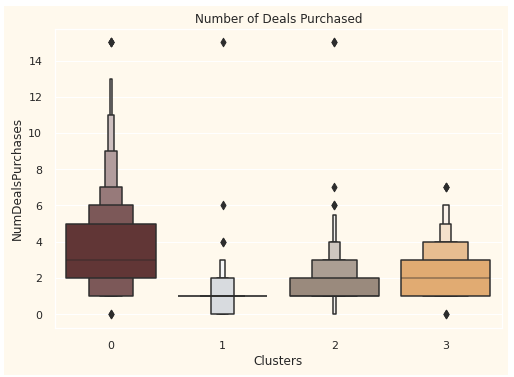
In the association analysis part, I will **use one more dataset on the mlxtend rule generation package** I implemented, and also improve on that to **add more visualization techniques** and graphs. Code: <https://colab.research.google.com/drive/1Dh8I7jaes5Cb3o--HnctFfjeGGk4hsEO>

* Since the dataset has 7500 transactions and 119 different kinds of products of commodities, I tune the min\_sup hyperparameters to be 0.05 so that at least 10 frequent itemset shows up for rule generation.
* The count for each item are shown as below, which confirm the later association rules
* 
* Then I use confidence as metric and set min\_conf 0.01, with 4 rules generated out
* 
* We can see from here that **mineral waters are often associated with eggs and spaghetti,** which is interesting to know because they are all from different categories.
* This transaction has 7500 transactions, so I have to use a min\_conf of 0.01 to generate 4 rules. This might be due to the fact that the **dataset has a really specific classification of products**. For example, rather than only having oil as a product, they have cooking oil, olive oil, etc. Therefore this will make frequent itemset less, meaning fewer association rules will be generated.
* I also use conviction as a metric with min\_conv = 1, the same 4 rules generated, which means the algorithm is stable with respect to this dataset regardless of metric.
* 
* I decide to lower the min support to include more frequent itemset because of the large dimension, so more rules can be generated
* 
* We can see we have more products being associated with each other, including chocolate, milks, pancakes, which are some pretty frequent-buying products in households.
* I also analyze the **association between different metric, particularly confidence and conviction**
* 
* We can see they are positively correlated, with **0.94 correlation (strongly positive),** which means they are almost interchangeable when generating association rules.

**Clustering**

Link to previous DMP: [DMP\_Cluster\_Analysis\_Entry\_Frank](https://docs.google.com/document/d/1Ou832bkY8kcxh-w5RyAC8GphiEJD417d5g5oW79D6XY/edit)

In the previous DMP, I implemented k-mean, k-mean ++, DBSCAN and other analysis metrics. I use packages of DBSCAN, Agglomerative, and k-mean to compare the result.

* I will discover more on clustering in Anomaly Detection section, but in this section, I **introduced a new dataset and perform Agglomerative Clustering** for a business related case study some more advanced clustering analysis referencing to: <https://www.kaggle.com/karnikakapoor/customer-segmentation-clustering>
* Code is : <https://colab.research.google.com/drive/12TpA-bn0dHhp66W6h192IxhvhUMd8diP#scrollTo=9WOe_Dz94ZmG>
* This task, given customer data, **helps a company to analyze its potential segmentation of customers** which they can target. This unsupervised clustering helps companies to modify the product to satisfy the need and behavior of each segment. This is an important analysis in the field of business marketing to best utilize marketing budget.
* When looking at the data using pandas, there are missing data on income
* 
* Also we need to encode the data type ‘object’ since they are categorical
* They require some **data preprocessing**, and I just remove all missing data from this dataset, engineer some data (create extra features that combine several features, simplify some categorical data into fewer ranges, etc.), drop some outliers of extreme values, and scale the data into [0,1]
* We learned **PCA (principal component analysis),** and I applied this dimensional reduction technique to this dataset to increase interpretability and decrease information loss. (There are packages in python that helps you to do so)
* 
* 
* Then, I will use **Agglomerative Clustering** — a hierarchical clustering method — to cluster the dataset. It involves merging examples until the desired number of clusters is achieved.
* I choose a k = 4 by using the elbow method (**choose the turning point** as we learned in class) but in a more formal mathematical way.
* This is the clusters:
* 
* It seems like the clusters are successful in segmenting the most important factor — income (representing the purchasing power) — even using reduced dimension data (which means the most important information has remained).
* 
* There are lots of interest analysis you can do this the package seaborn and pandas, and I have one more that analyze number of deals each cluster participated so company can design future product with the popularity among different income group in mind
* 
* In this section, in addition to clustering, I gain more insight on how to utilize existing clusters and powerful graphing tools to analyze information from it. This is the essence of “data MINING”, isn’t it?

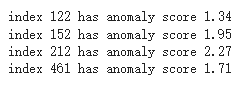
**Anomaly Detection**

**Code Development**

1. Proximity-Based Anomaly Detection

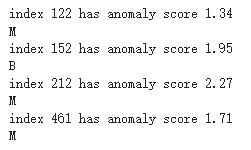
Here I implemented a **proximity-based approach detection**, specifically, using the distance to its kth nearest neighbor dist(x,k), and ran against several datasets to analyze the result. I also do hyperparameter tuning to make sure it is good.

Code: <https://colab.research.google.com/drive/1uKELN90AomE__CcK39dm-JH0gVoZoxqQ#scrollTo=XYxY_oBKF2x->

* I use the **cancer data we have in classification**, using the proximity-based anomaly detection approach I used to detect some anomalies.
* I use **euclidean distance as L2 norm, specifically choose k = 3** for cancer data with anomaly score disk(x,k) threshold set to be 1.3
* I identified 4 anomalies (potential) out of the data who have anomaly scores of the following:
* 
* I **delete the anomalies and reclassify them using the same decision tree (sklearn) to compare difference performance measure** so see if anomalies impact the accuracy of the model
* Here is the result with same hyperparameters (test\_train\_split = 0.1, entropy as criterion, max\_depth = 30, min\_split = 5)

|  | Before Delete Anomalies | After Delete Anomalies | Improvements |
| --- | --- | --- | --- |
| Accuracy | 91.23% | 96.49% | +5.26% |
| Recall | 85.19% | 95.83% | +10.64% |
| Precision | 95.83% | 95.83% | +0% |

We can see that **Accuracy, and Recall are all improved to a significant amount**. Recall is the true positive rate, and in this case it means the percentage of M cancer we identified as M. This is an important metric to see if our model can correctly identify more M cancer since M is definitely weighted more than B. To further discover why the performance improved. **I identified the class label of the four data I dropped**



We can see that 3 of them are malicious cancer.

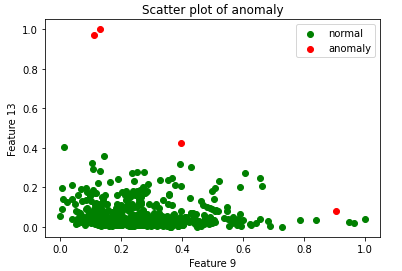
* Before dropping the anomalies, there are 27 M, we successfully classified 23 (85%)
* After dropping the anomalies, there are 24 M, we successfully classified 23 (96%)

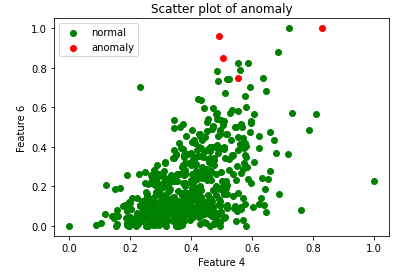
Surprisingly, **all the malign cancer data we dropped are incorrectly classified before**. This is pretty amazing because it means our anomaly detection method is powerful in identifying anomalies and surprisingly all the anomalies are being incorrectly classified.

This also is a bad sign because it means **malignant cancer is not following a similar / predictable pattern** (clustered together). Most of them can be identified as anomalies in a dataset. The fact that it is being identified as anomalies in the dataset doesn’t mean contextually the data point should not be taken care of. The model should be more reliable in classifying them (outliers) to be able to save patient’s lives because this is cancer, not other things.

I also randomly choose 2 features to plot the graph out

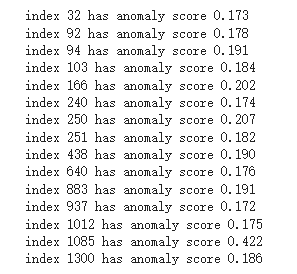
* Feature 9 versus Feature 13



* Feature 4 versus Feature 6
* 

We can see our anomalous does not fit into the large cluster of our data.

This dataset is not complicated enough, so I will test on new data. I use **insurance data** that has demographic information about the patients and the cost of their insurance. Insurance wants to identify anomalies about these insurance buyers to make sure they have enough profits. Here is my result after hyperparameter tuning to make sure anomalies are further enough from other points (k = 3, threshold = 0.17)

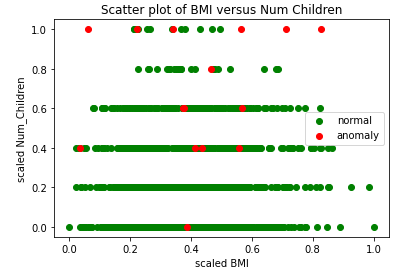


Index 1085 who has an anomaly score of 0.422 is definitely an anomaly because it is so different from the other points.

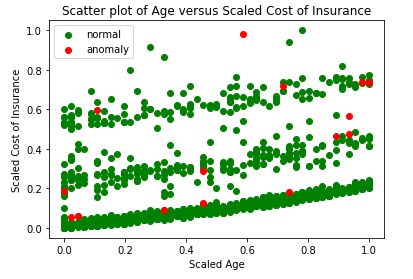


Obviously, a 39-year-old female that has a BMI of 18.3 (underweight) that has 5 children and smoke is some profile that is super rare, even rare in population.

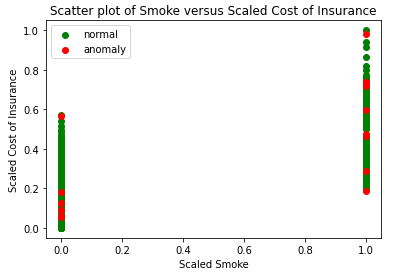
I plot out the graph and color those anomalies in a different color so that it is easier to visualize.



* More children an individual has, more likely it is anomaly



* Older the individual is, more likely it is anomaly



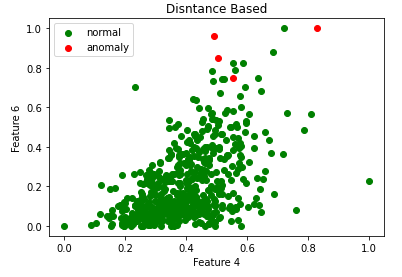
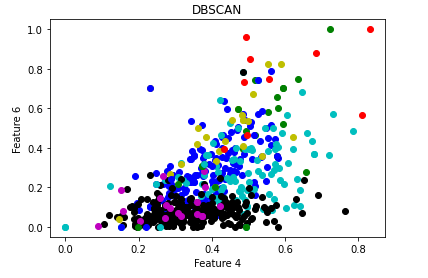
* We can also see person who smoke will has a higher charge of insurance (don’t smoke)
* And also they are more likely to be anomalous

The strength of proximity based anomaly detection is that it obviously is **not restricted to the distribution of the data and very generalizable to different types of datasets**. However, the weakness is that the result really depends on the **choice hyperparameters** (in this case is k and threshold). It often takes a very long time to choose a good combination of k and threshold to make the anomalies really anomalous. This is especially more challenging in high-dimensional datasets.

1. Clustering-Based Anomaly Detection

I will reuse the **DBSCAN algorithm** I implemented during clustering to do it with anomaly detection. Specifically, the noise point I identified will be classified as anomaly. In my clustering DMP, I didn’t try it with cancer data. This time I will **try it with cancer data to see if it follows the pattern I have in proximity-based anomaly detection**.

* Hyperparameters for DBSCAN
  + MinPts = 5
  + Eps = 0.8



* First graph is DBSCAN, with the RED color indicating the anomalies and other colors for different clusters, and the second graph is the distance (proximity) based, with the RED color indicating the anomalies.
* Generally speaking, they identified some points the same but not all anomalies the same. However, **both methods identified points that are around the border of the clustering graph**, which is what we expected.

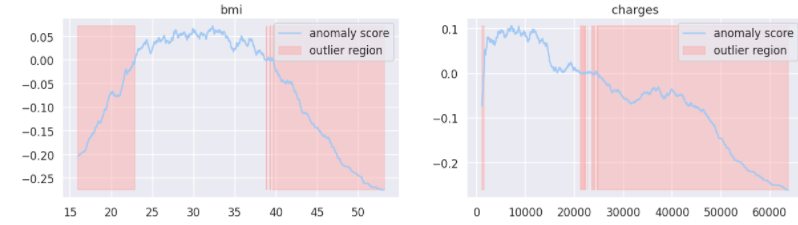
The strength of clustering-based approach is that it can help us understand the nature of normal data, and be more time efficient. However, the performance also heavily relied on the choice of hyperparameters and it is operated under an unsupervised setting.

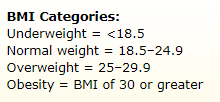
**Package Usage (scikit-learn)**

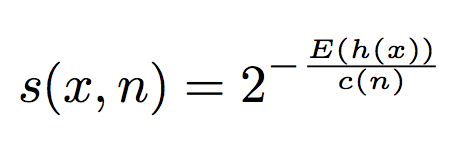
I will applied **Isolation Forest, DBSCAN, and Z-score** (normal distribution) to analyze the anomaly for a new data

1. **Isolation Forest**

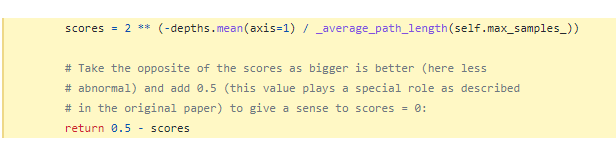
* Isolation Forest detect anomalies by **partitions the data and provide an anomaly score** (number of split need to isolate a point) that representing how isolation the data point is from other data
* I will use the insurance dataset and identify anomalous in terms of 2 numeric data — BMI, costs of insurances.



* We computed the anomaly score using the package of IsolationForest and then classified each observation as an outlier or non-outlier. I shaded the outlier region with red color, and as what we are expected, the anomalies are corresponding to low probability areas (which typically are some extreme values)
* 
* We can see our BMI anomalies generally follow the standard, but notice that BMI higher than 30 are identified as obesity. However, in the graph, BMI > 30 is still normal, meaning that there is a greater percentage of the population suffering from obesity than it should, which urges us to adopt a more healthy lifestyle.
* I am also interested **why anomaly score can be negative**, and I searched out that the anomaly score of isolation forest are calculated as



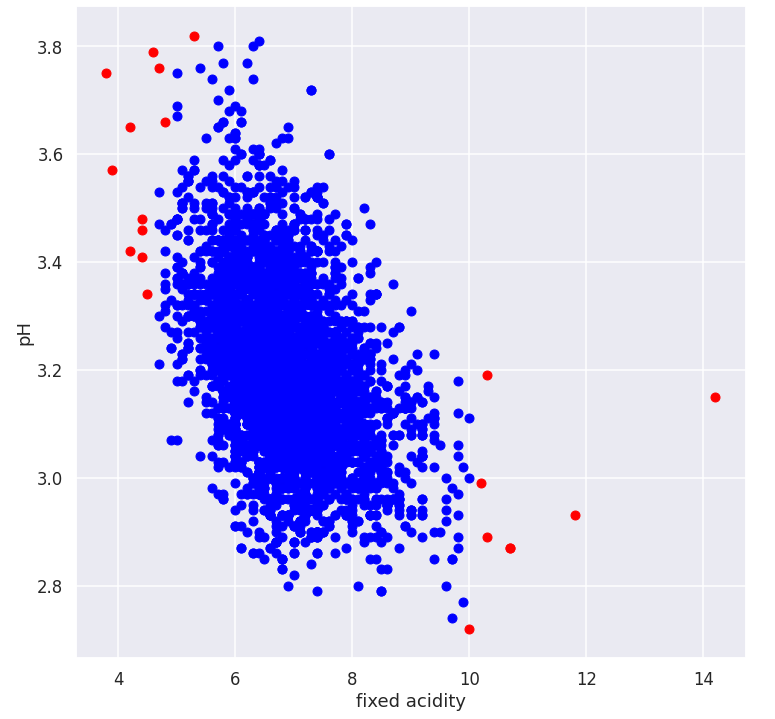
* h(x) is the path length, c(n) is the average path length. This would range in [0,1], but on Google, it is said that the anomaly score for the isolation forest package is range in [-0.5, 0.5], with -0.5 indicating most anomalous. Out of curiosity, **I looked at the source code from the GitHub of scikit-learn isolation forest:**



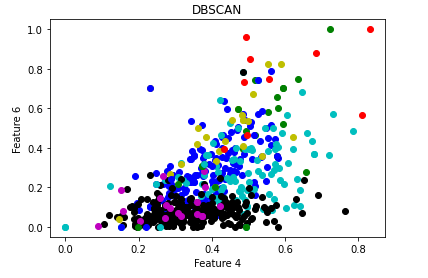
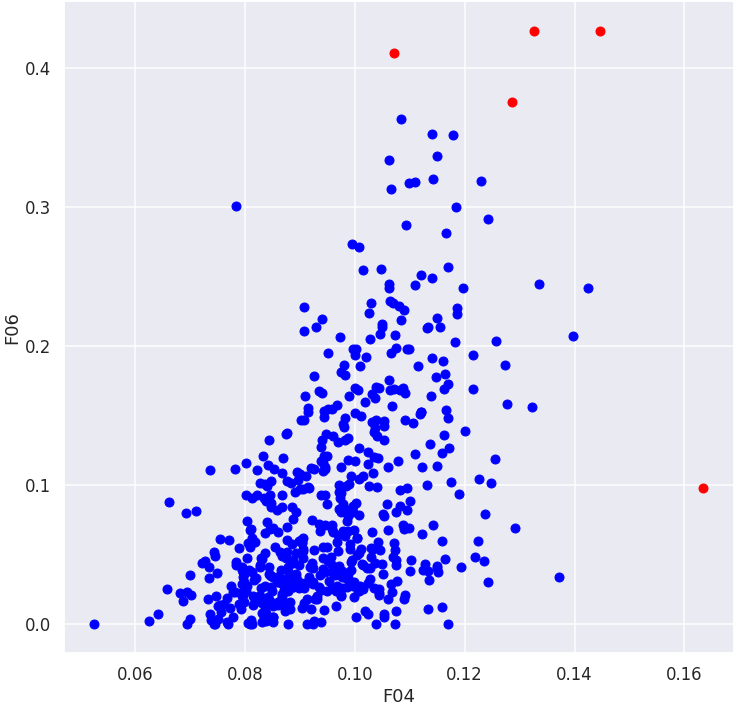
* They use 0.5 subtract the score so that it is more align with our common sense (negative is worse)
* This is interesting! Looking back to how people implement those packages can give you some insight you never thought about.

1. **DBSCAN**

* Here I use the DBSCAN package to apply anomaly detection to the **wine dataset.** Specifically, I want to know of the acidity of a whine can create anomalous pH value of the while
* I use hyperparameters of Eps = 0.2 and MinPts = 10

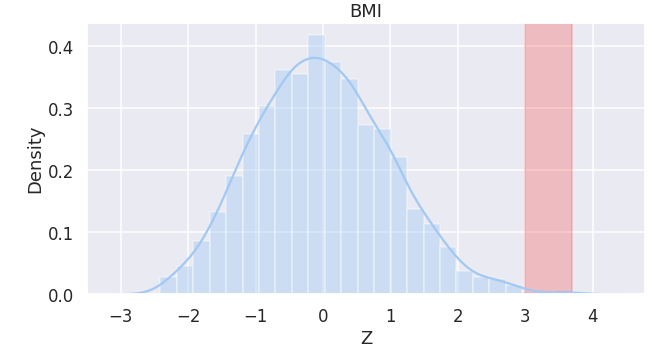


* Regardless of the content, we can see that anomalies are often on the border of the clusters, and are what we expected to see.
* I will also compare the package DBSCAN with my own DBSCAN using cancer data (Feature 4 and Feature 6)



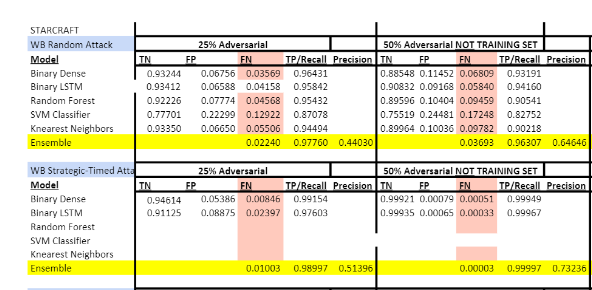
* Note the left one is package and right one is code
* The scale may be different because package DBSCAN does not sale the data, but the outliers are in similar areas (border)

1. **Z-Score**

* We learn that **normal distribution follows 68, 95, 99.7 patterns**, which means 68%, 95%, 99.7% of data are within 1, 2, 3 standard deviations from the mean respectively.
* We can use this property to conduct anomaly detection.
* I will continue to use the insurance data, and identified points that are 3 standard deviation away from the data to be anomalous
* 
* We have total **4 anomaly data points in terms of BMI, and they take up less than 0.3% of the data (ideally if it follows normal distribution)**

1. **Machine Learning Approach from my research**

In the textbook, anomalies are often defined as a data point that is distance away from other normal points, and either in proximity-based approaches or clustering-based approaches, data points are being identified as anomalies using proximity or distance measures. However, not all situations are similar to this. In my Reinforcement Learning research, **we use anomaly detection to detect adversarially attacked agent’s observations**. In normal situations, an AI agent typically will observe its environment and use its policy network to choose the action that can give the agent the most reward. If the environments are attacked or polluted, agents may observe anomaly observations. **This application cannot use distance measures because all observations are very high-dimensional (image pixels)**, and for example, a self-driving car observing a red light or stop sign are all normal observations, but two inputs are very different from each other. Reinforcement Learning also has a time-series in nature. This would require some AI algorithms that can learn the pattern of normal observations, so **KNN, SVM, Random Forest, Neural Networks, and Ensemble method** can play an important role here. This work is done by our group, I will attached part of the result table here, and code can be found on my GitHub:<https://github.com/frank47ltt/Multi-Agent-Security/tree/main/MARL_Detection>



This only serves as an application of the concept I learned this semester that I have applied in another situation, but I have not done any extra coding here in this DMP.

**Final Exercise Theory**

[Anomaly&Final Exercise](https://docs.google.com/document/d/1Aol3Amsc3yG84WBFVD-fXtZFdaSoroNfv0bZrq4-Mpo/edit)

**Semester Summary and Major Learning Point**

* Different algorithms and their strengths and weaknesses in following big categories
  + Classification
    - Decision Tree
    - KNN
    - Bayes
    - Neural Networks
    - Random Forest
  + Clustering
    - K-mean
    - Derivatives of K-mean(bisecting, etc.)
    - DBSCAN
    - Agglomerative
  + Association Analysis
    - Apriori
  + Anomaly Detection
    - Isolation Forest
    - Proximity-Based
    - Density-Based
    - Clustering-Based
    - ANN
* Fundamental tasks of Data Mining
* How to clean and pre-process data, and understand their importance quantitatively and qualitatively
* Different techniques to improve performance
* Know how to deal with different types of model problems (overfitting, underfitting, etc.) and when to use what in different contexts
* Use different packages of Python and ability to code in python
  + Matplotlib
  + Pandas
  + Seaborn
  + Numpy
  + Tensorflow
  + Scikit-learn
  + Etc.
* Ability to implement algorithms from scratch
* Analysis skills
* Ability to apply concept to different situations outside of CS
* Self-learning and improving ability

**Learning Summary and Self-Assessment:**

Overall, I think I had a decent understanding of the material and gained lots of insights in doing the coding and DMP. I spent significantly more time on this class than the other class, part of it might be because the workload is not easy, also because this is the thing I enjoy doing and want to continue in the future. In the era of big data, the analytical ability and problem solving skills I learned from this class can better equip me to be successful in any career I choose to do in the future. Overall, I think I have an A level mastery in CSC373.

By the end of this DMP, I also want to share the good news with you that I just got an offer from IBM as an IBM Research Summer Intern in machine learning. Machine learning and data mining techniques and the portfolios I did became a big part of the discussion during the interview, and I applied lots of concepts to their questions. I really appreciate the design of you having us to learn and discover by ourselves. I learned a lot from this, and this is typically how you approach a project in the workplace. I really enjoyed this class, and hope you have a wonderful winter break.