**Intro:**

Credit card fraud is a widespread issue in the modern day where it is often difficult to tell whether or not a transaction was fraudulent based on its data. One way to solve this problem is by sending the user a message every time a transaction was conducted, which the user either confirms or denies, but this can be an unnecessary hassle. Some of the largest challenges associated with credit card fraud identification are speed (the algorithm needs to be fast enough to process enormous amounts of data), imbalanced data (only a very small percentage of recorded transactions are fraudulent), and misclassification (some fraudulent transactions are allowed to go through). This project aims to use several machine learning algorithms in order to classify a new transaction as either fraudulent or legitimate based on the various parameters in the dataset.

**Dealing with Imbalanced Data:**

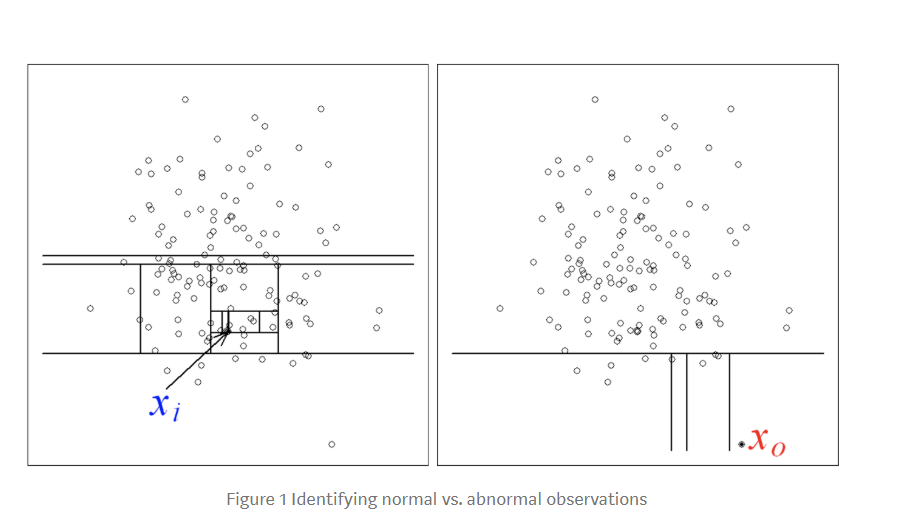
The dataset for credit card fraud transactions is highly imbalanced, with only about .17% of the cases being fraudulent. Machine learning algorithms are notoriously bad at handling imbalanced transactions because they are very likely to show a bias toward the majority data class (non-fraudulent transactions). In general, there are two easy methods to handle imbalanced data sets: oversampling and undersampling.

1. Undersampling: This involves simply removing instances from the majority class. One way to do this is by randomly selecting instances to remove (can be harmful because it can lead to changes in the makeup of the majority class which can affect overall trends). Another way is *informative sampling* which separates the instances of the majority class with multiple classifiers and decides which instances of the majority class to delete.
2. Oversampling: This method is essentially adding more instances to the minority class of data. One way to do this is again, by randomly selecting instances of the minority class, copying them, and adding them to the data set. The SMOTE (synthetic minority oversampling technique) algorithm is also a type of oversampling technique which creates artificial data to add to the dataset based on the feature space. It first takes the difference between the feature vector (sample) under consideration and its nearest neighbor, multiplies this difference by a random number between 0 and 1, and adds it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features.

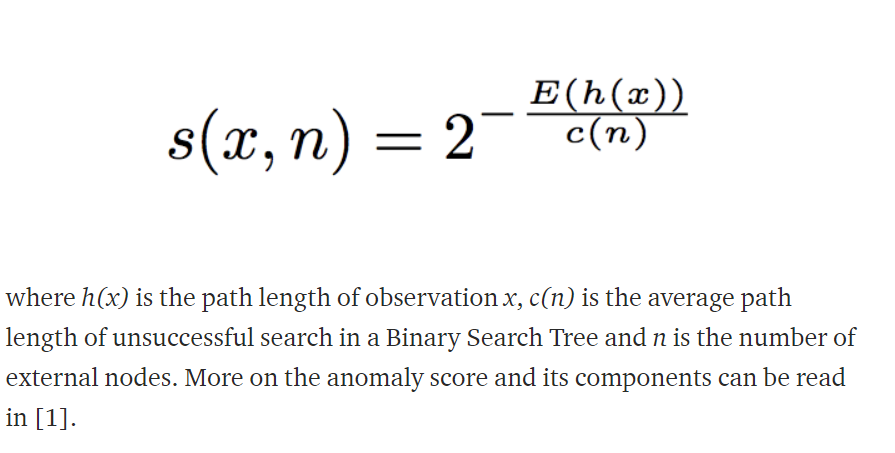
**Algorithms:**

A variety of traditional machine learning algorithms can be applied to this dataset. I’ve detailed a few of these possibilities below

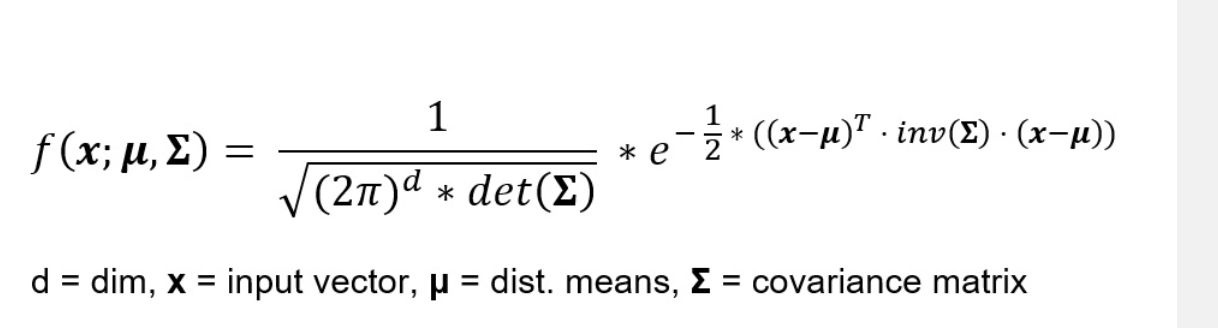
1. **Isolation Forest Algorithm (unsupervised):** This algorithm is a great choice for anomaly detection. It is based on multiple decision trees (binary trees) where partitions in the trees are created by randomly selecting a dimension and determining a random split value between the minimum and the maximum of the selected feature. In the case of this application, this would mean randomly selecting a feature from V1 through V28 and selecting a random value between the min and max of the feature. More partitions are recursively generated, until each data point is its own separate leaf. Anomalous data point leafs will have a much smaller path from the root because they will require less partitions to be created. This process can be repeated in multiple dimensions.



The image shows a dataset with two dimensions where anomalous points like X0 require less splits than normal data like X1. In addition, each point is given an anomaly score.



1. **Local Outlier Factor (Unsupervised):** This is a calculation that looks at the neighbors surrounding a given point and compares this to other points to estimate the likelihood that the given point is an outlier. The parameter *k* is the number of neighbors that will be considered with any point. The *k-distance* for any point can then be calculated as the distance from the point to the kth closest point. The *reachability distance* of two points A and B is the maximum value between the distance between the points and the k-distance of the second point. Basically, if B is within the K-distance of A, then instead of the real distance between the two point, that k-distance is used. The *local reachability density* is found by taking the reachability distance of a to all its k nearest neighbors, taking the average, and taking the inverse of that average. This measure basically shows how far away the next nearest cluster of points is, so the lower it is, the more of an outlier the point is. The last step is comparing the reachability density of every point with that of its k neighbors. There will be k ratios calculated (each one compares the lrd of the current point to that of its neighbors) and they are averaged. If the LOF is much greater than 1, it is an outlier because that means its much farther away from its neighbors than average.
2. **Random Forest Algorithm:** This is a supervised machine learning algorithm which works with multiple decision trees. First, random subsets with various random combinations of columns(features) and rows of data are created. A decision tree algorithm such as Gini index or ID3 is used to create a separate decision tree for each random subset. ID3 would be a good choice to generate a decision tree for this type of classification problem. First, the information gain is calculated for each feature in the subset, and the feature with the highest gain becomes the root. Going down from the root, the information gain is recursively calculated and the feature with the highest information gain becomes the next decision node. This goes on until either all of the data is classified perfectly, or we run out of attributes. When a new data point (a credit card transaction in this case) is fed through the algorithm, each decision tree in the forest has an output of either 1 (fraudulent) or 0 (non-fraudulent). Whichever output occurs more often across all the decision trees is the algorithm’s overall output.
3. **Multivariate Gaussian Distribution:** This algorithm essentially fits the data to a multi variable gaussian (normal) distribution, the pdf for which is given below.



Based on some threshold, a particular data point can be classified as normal or abnormal.

**Bibliography:**

Articles:

1. <https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/>
2. <https://builtin.com/data-science/random-forest-algorithm>
3. <https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1>

Selected Papers:

1. <https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf?q=isolation-forest>
2. <https://www.irjet.net/archives/V6/i3/IRJET-V6I3710.pdf>(IEEE Research Paper on Credit card fraud)
3. <https://arxiv.org/pdf/1905.07107.pdf>
4. <https://ieeexplore.ieee.org/document/7100189>
5. <https://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm>