**Credit Card Fraud Detection**

Before doing dimensionality reduction (PCA), the values should be scaled

**RobustScaler** – it is a scaling technique which can be used when we have outliers in the data. It is less prone to outliers

**TYPES OF CROSS VALIDATION**

Whenever we change the random state, the accuracy will also change in train test split. So, instead we can use cross validation techniques

* **Leave one out CV(LOOCV):** Only one data will be used for testing purpose, remaining will be training data. Need many iterations and computational power for this, low bias prblm
* **kfold cross validation: K- no.of experiments**
* **StratifiedKFold:** same like kfold cross validation, in this it makes sure that number of instances are taken in a proper way (eg: if yes/no. in both training and test we will have a good propotion of both)

**Stratified kfold cross validation** is typically useful when we have imbalanced data and where the data size is on the small side

When the data is large enough, we can still use **regular kfold cross validation**

**StratifiedKFold** is the improved version of **KFold**

**we should prefer StratifiedKFold over KFold when dealing with classification tasks with imbalanced class distributions**

* **Time series cross validation:** used for time series related problems

The main issue with "Random Under-Sampling" is that we run the risk that our classification models will not perform as accurate as we would like to since there is a great deal of **information loss** (bringing 492 non-fraud transaction from 284,315 non-fraud transaction)

**Gaussian distribution  = normal distribution = Z distribution**

Dimensionality reduction technique:

# **t-SNE: T-distributed Stochastic Neighborhood Embedding**

It takes a high dimensionality dataset and reduce it to a low dimensional graph.

Chart

Description automatically generated with medium confidence

Chart, bubble chart

Description automatically generated

Chart

Description automatically generated

Method to Visualizing high dimensional data

t-SNE tries to figure out how similar the points are and groups them based on their similarity and is therefore able to preserve the inner structure of the data.

Consider the figure below, t-SNE figures out the distances between the points and groups the closer points to form clusters in lower dimensions as well

Chart

Description automatically generated with medium confidence

With the query point at the center, the distances relative to the other points are measured for each point. The further a point is away from a query point, its distance would lie away from the peak of the curve. The points which lie near the peak would be considered the neighbors of the query point.

Chart

Description automatically generated

This is repeated for all the points to figure out multiple clusters.

**Student’s t distribution (t distribution):**

Solving the problem of “small sample statistics”

**Central Limit Theorem (C.L.T):** If your sample size gets large enough, we can just use a normal distribution for our sample statistics. But if it is too small we can’t apply C.L.T

A picture containing text, green, sign, outdoor

Description automatically generated

MEAN, SIGMA

N – Normal distribution

A picture containing text, green, sign, street

Description automatically generated

A picture containing text, hitting, sign

Description automatically generated

Diagram

Description automatically generated with medium confidence

Diagram, schematic

Description automatically generated with medium confidence

**yan yun liu**

**TruncatedSVD (**singular-value decomposition)**: it is a dimensionality reduction technique**

* The **wider the gap** between the training score and the cross validation score, the more likely your model is **overfitting (high variance)**.
* If the score is low in both training and cross-validation sets this is an indication that our model is **underfitting (high bias)**
* **Logistic Regression Classifier** shows the best score in both training and cross-validating sets.

**Near miss algorithm – for imbalanced dataset**

The Near-miss Algorithm is used to balance an imbalanced dataset and is considered as an algorithm for **undersampling** and is one of the most powerful ways to balance data

The Near-Miss algorithm works by observing the class distribution, removing samples located in the higher class. Simply put, if the algorithm witnesses a case in which two near points that pertain to different classes occur, it simply excludes the one from the higher class and ensures that the balance is preserved

# **Learning curve: Diagnose Machine Learning Model Performance**

## **Learning Curves:**

* The **wider the gap** between the training score and the cross validation score, the more likely your model is **overfitting (high variance)**.
* If the score is low in both training and cross-validation sets this is an indication that our model is **underfitting (high bias)**
* **Logistic Regression Classifier** shows the best score in both training and cross-validating sets.

**Confusion matrix:**

* **True Positives:** Correctly Classified Fraud Transactions
* **False Positives:** Incorrectly Classified Fraud Transactions
* **True Negative:** Correctly Classified Non-Fraud Transactions
* **False Negative:** Incorrectly Classified Non-Fraud Transactions
* **Precision:**True Positives/(True Positives + False Positives)
* **Recall:**True Positives/(True Positives + False Negatives)
* Precision as the name says, says how precise (how sure) is our model in detecting fraud transactions while recall is the amount of fraud cases our model is able to detect.
* **Precision/Recall Tradeoff:**The more precise (selective) our model is, the less cases it will detect. Example: Assuming that our model has a precision of 95%, Let's say there are only 5 fraud cases in which the model is 95% precise or more that these are fraud cases. Then let's say there are 5 more cases that our model considers 90% to be a fraud case, if we lower the precision there are more cases that our model will be able to detect.

**SMOTE** stands for Synthetic Minority Over-sampling Technique -**Oversampling technique**

## **Overfitting during Cross Validation:**

In our undersample analysis I want to show you a common mistake I made that I want to share with all of you. It is simple, if you want to undersample or oversample your data you should not do it before cross validating. Why because you will be directly influencing the validation set before implementing cross-validation causing a "data leakage" problem. **In the following section you will see amazing precision and recall scores but in reality our data is overfitting!**

Chart, bar chart

Description automatically generated

Chart

Description automatically generated

Feature scaling is mandatory for all deep learning models