**STEP 1:**

**DATA PREPARATION AND FEATURE ENGINNEERING**

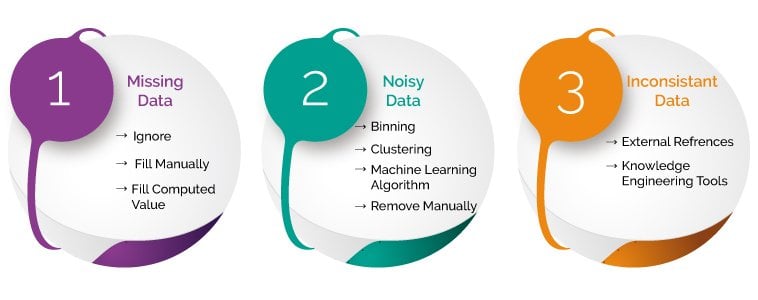
* The first step is to prepare the data for anomaly detection. We will start by importing the necessary libraries and loading the dataset into a Pandas.
* Convert categorical variables into numerical representations for model compatibility.

**STEP 2 :**

**DATA PREPROCESSING**

Before applying any anomaly detection algorithm, it is essential to preprocess the data to ensure that it is in a suitable format for the algorithm. Here are some steps that we can follow to preprocess the dataset.

* **Handling Missing Values**
* Missing values can affect the performance of the anomaly detection algorithm. Therefore, it is essential to check whether there are any missing values in the dataset and take appropriate action.
* **Scaling the Data**
* Anomaly detection algorithms can be sensitive to the scale of the data. Therefore, it is important to scale the data before applying the algorithm. We can use the StandardScaler class from the sklearn.preprocessing module to scale the data.



**Step 3:**

**MODEL SELECTION AND TRAINING**

**Algorithm Selection:**

* Choose the advanced techniques such anomaly detection algorithms(e.g., Isolation Forest, One-Class SVM) considering the dataset's characteristics and complexity.
* Experiment with ensemble methods for improved fraud detection accuracy.

**(i)Anomaly Detection Algorithms** :

There are various anomaly detection algorithms available. In this section, we will discuss some popular algorithms along with their implementation in Python.

* Isolation Forest:
* Isolation Forest is a popular algorithm for anomaly detection that is based on the concept of decision trees". It works by creating random decision trees for the given data and isolating the anomalies by creating shorter paths for them.
* One-class SVM:
* One-class SVM is another popular algorithm for anomaly detection that is based on the concept of maximum margin hyperplanes. It works by creating a hyperplane that separates the normal data points from the anomalies and identifying points that lie on the wrong side of the hyperplane as anomalies.

**(ii)Ensemble techniques:**

* We can make an ensemble model of a few of the models like **Logistic Classifier, Random Forest, K nearest neighbor, and Decision Tree**(eg.,bagging,boosting,stacking etc.,)see if the ensemble model performs better than the individual model. Generally, in practice also if the majority of the models predict a particular transaction as fraud, it is highly likely that it ultimately turns out to be a fraud. One downside is that the processing time of the model takes a hit even though the accuracy improves.



**STEP 4:**

**EVALUATION AND OPTIMISATION**

* The final step is to evaluate the selected model performance to make predictions on new data. In this step, we will use the selected model to make predictions on the test dataset and evaluate its performance using classification metrics.
* We will use the predict method of the trained model to make predictions on the test data, and then evaluate the model’s performance using accuracy\_score, precision\_score, recall\_score,  F1\_score and ROU-AUC metrics from the sklearn.metrics module.

**STEP 6 :**

**DOCUMENTATION**

* Document the entire analysis process, including data preprocessing, model selection, training, evaluation optimization.
* Provide clear explanations of chosen algorithms, parameters, and evaluation metrics.

**CONCLUSION:**

* OUTPUT :
* In this project we will able to develop and learn from a training set of historical fraudulent and legitimate transactions.
* Statistical deviations from the customer's credit card profile will act as variables that show inconsistencies in comparison with the credit card's historical trends.