# **Credit Card Fraud Detection With Classification Algorithms In Python**

Fraud transactions or fraudulent activities are significant issues in many industries like banking, insurance, etc. Especially for the banking industry, credit card fraud detection is a pressing issue to resolve.

Credit card fraud detection using different Machine Learning Classification Algorithms.

**Classification Algorithms**

* Linear classifiers
  + Logistic regression
  + Naive Bayes classifier
  + Fisher’s linear discriminant
* Support vector machines
  + Least squares support vector machines
* Quadratic classifiers
* Kernel estimation
  + k-nearest neighbor
* Decision trees
  + Random forests
* Neural networks
* Learning vector quantization

## **Application of Classification Algorithms**

* Email spam classification
* Bank customers loan pay bank willingness prediction.
* Cancer tumour cells identification.
* Sentiment analysis.
* Drugs classification
* Facial key points detection
* Pedestrians detection in an automotive car driving.

# **Introduction to Decision Tree Algorithm**

The general motive of using Decision Tree is to create a training model which can use to predict class or value of target variables by **learning decision rules** inferred from prior data(training data).

**Introduction to Random Forest Algorithm**

The random forest algorithm is a supervised classification algorithm. As the name suggests, this algorithm creates the forest with a number of trees.

In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high the accuracy results.

## **Why Random forest algorithm**.

* The same **random forest algorithm** or the random forest classifier can use for both classification and the regression task.
* Random forest classifier will **handle the missing** values.
* When we have more trees in the forest, a random forest classifier won’t **overfit** the model.
* Can model the random forest classifier for **categorical values** also.

#### **Random Forest pseudocode:**

* Randomly select “k” features from total “m” features, where k<<m.
* Among the “k” features, calculate the node “d” using the best split point.
* Split the node into daughter nodes using the best split.
* Repeat 1 to 3 steps until “l” number of nodes has been reached.
* Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.

The beginning of random forest algorithm starts with randomly selecting “k” features out of total “m” features. In the image, you can observe that we are randomly taking features and observations.

In the next stage, we are using the randomly selected “k” features to find the root node by using the [best split](https://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/) approach.

In the next stage, We will be calculating the daughter nodes using the same best split approach. Will the first 3 stages until we form the tree with a root node and having the target as the leaf node.

Finally, we repeat 1 to 4 stages to create “n” randomly created trees. These randomly created trees form the random forest.

**Introduction**

Credit card fraud is a significant issue in the financial industry, leading to substantial financial losses each year. Detecting fraudulent transactions is crucial to mitigate these losses and protect consumers. This project aims to develop a machine learning model to detect fraudulent credit card transactions using various classification algorithms. The dataset used in this project contains transactions made by credit cards in September 2013 by European cardholders.

**Data Preprocessing**

Data preprocessing is a crucial step to ensure that the dataset is in an appropriate format for machine learning models. Here are the preprocessing steps undertaken:

1. Loading Data: The dataset is loaded into a pandas DataFrame.

2. Standardizing the 'Amount' Feature: The 'Amount' feature is standardized using `StandardScaler` from `sklearn.preprocessing` to ensure all features have a similar scale.

3.Dropping Irrelevant Features: The 'Time' feature, which represents the elapsed time between the transaction and the first transaction in the dataset, is dropped as it is not directly relevant to the prediction.

4.Handling Duplicates: Any duplicate rows in the dataset are removed.

5.Handling Class Imbalance: The dataset is highly imbalanced, with the majority of transactions being non-fraudulent. To address this, the `SMOTE` (Synthetic Minority Over-sampling Technique) is used to balance the classes.

**Machine Learning Models** - Several machine learning models were evaluated for their effectiveness in detecting fraud:

**1. Logistic Regression:**

Theory: Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. It is used to estimate the probability that a given input point belongs to a certain class.

Performance Metrics:

Without Resampling:

- Accuracy: 90.16%

- Precision: 93.75%

- Recall: 87.38%

- F1 Score: 90.45%

With Resampling (SMOTE):

- Accuracy: 97.22%

- Precision: 97.96%

- Recall: 96.47%

- F1 Score: 97.21%

**2. Decision Tree Classifier:**

Theory: A decision tree is a flowchart-like tree structure where an internal node represents a feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It is a supervised learning method used for classification and regression.

Performance Metrics:

Without Resampling:

- Accuracy: 84.97%

- Precision: 86.27%

- Recall: 85.44%

- F1 Score: 85.85%

With Resampling (SMOTE):

- Accuracy: 99.85%

- Precision: 99.76%

- Recall: 99.94%

- F1 Score: 99.85%

**3. Random Forest Classifier:**

Theory: Random Forest is an ensemble learning method that operates by constructing multiple decision trees during training and outputting the mode of the classes (classification) or mean prediction (regression) of the individual trees. It helps in improving the accuracy and controlling overfitting.

Performance Metrics:

Without Resampling:

- Accuracy: 96.11%

- Precision: 92.00%

- Recall: 82.52%

- F1 Score: 87.99%

With Resampling (SMOTE):

- Accuracy: 99.85%

- Precision: 99.76%

- Recall: 99.94%

- F1 Score: 99.85%

**Results Analysis**

Accuracy: This metric measures the overall correctness of the model. It is the ratio of correctly predicted instances to the total instances. All models showed high accuracy with resampling, with Decision Tree and Random Forest Classifiers achieving 99.85%.

Precision: This metric measures the accuracy of the positive predictions. Logistic Regression showed slightly lower precision compared to Decision Tree and Random Forest Classifiers.

Recall: This metric measures the ability of the model to find all the relevant cases within a dataset. Decision Tree and Random Forest Classifiers had higher recall, indicating they were better at identifying fraudulent transactions.

F1 Score: This is the harmonic mean of precision and recall, providing a single metric to evaluate the model. Decision Tree and Random Forest Classifiers outperformed Logistic Regression with an F1 Score of 99.85%.

**Conclusion**

The Decision Tree and Random Forest Classifiers demonstrated superior performance across all metrics compared to the Logistic Regression model. Their high accuracy, precision, recall, and F1 scores make them robust choices for detecting fraudulent transactions in this dataset.

The models were saved using `joblib` for future predictions, ensuring consistent preprocessing and feature engineering. This approach provides a reliable method to detect fraudulent transactions, potentially saving financial institutions and consumers from significant losses.

By using a combination of data preprocessing techniques, handling class imbalance, and employing powerful machine learning algorithms, this project successfully developed an effective fraud detection system.

.