**Credit Card Fraud Detection in Python**

An average business loses not less than **5% of its annual profits to fraud**, according to a [survey](https://www.acfe.com/about-the-acfe/newsroom-for-media/press-releases/press-release-detail?s=ACFE-Estimates-Organizations-Lose-5-percent-to-Fraud) of Certified Fraud Examiners (CFEs), and this number is likely to keep growing if companies don’t take precautions.

Luckily, these days IT specialists can detect fraudulent transactions with the help of various techniques, such as fraud detection in Python, applying Machine Learning (ML) to analyze huge datasets, and other tools.

We will explain how we use Python to distinguish between fraud vs non-fraud.

## Fraud Scenarios for Detection

What is fraud detection? It is a collection of strategies, processes, methods, and techniques we use to identify unauthorized activity and prevent money or property from being taken by scammers.

According to this [research by Statista](https://www.statista.com/statistics/1296598/main-fraud-detection-tools-online-worldwide/), most companies use Card Verification Number (**54%**) and email (**43%**) for online fraud detection. Customer order history is another popular asset (**38%**), and this is where Machine Learning algorithms come in handy. ML helps process large sets of data with many variables finding unobvious correlations between normal user behavior and possible fraudulent activity.

E-commerce, MedTechand [FinTech](https://5ly.co/blog/4-fintech-industry-trends-to-keep-an-eye-on/) companies opt for maximum security, which can be enabled by Machine Learning algorithms that help with credit card fraud detection in Python.

How exactly does it work? How do you detect fraud in online transactions? There is a wide selection of methods in ML for that. For example, here is a brief list of cases we have dealt with:

1. [Solutions for the insurance industry](https://5ly.co/blog/11-smashing-tech-trends-in-the-insurance-industry/)

* Fake claims
* Duplicated claims
* Overstated repair cost

2. Healthcare insurance solutions

* Medical receipts and bills
* ID verification

3. E-commerce [web portals](https://5ly.co/web-portal-development-services/) and marketplaces

* Fraud in online orders
* Identity theft

4. Banking and credit cards

* Account theft and suspicious transactions
* Data credibility assessment
* Duplicate transactions

All in all, various businesses hire professionals that can provide fraud detection with Python. Data science specialists use ML algorithms to go through huge amounts of data as quickly as possible and discover suspicious actions on time. Let’s see how it works by using a credit card company as an example.

## Classification as a Fraud Detection Model (Python)

Some of the common cases that we have dealt with are potential credit card scams. This is important both for companies to not lose any money and for customers to not get charged for something they didn’t actually buy.

Let’s imagine we have a dataset with hundreds of thousands of transactions. Our task is to differentiate between the right ones and the suspicious ones, so we start a long process called credit card fraud detection in Python.  Cracking this case, we make classification models.

Why did we choose a classification model to discover fraud with Python? This method allows you to predict discrete variables such as true/false, yes/no, safe/fraud, etc.

Fraud detection with Python is considered to be a highly effective tool. **Why?** There is a bunch of reasons:

* Python is [among the most popular languages](https://en.wikipedia.org/wiki/Python_(programming_language)), loved by both developers and entrepreneurs
* It is relatively easy to learn, and a vast community is always here to help
* It supports lots of ML packages that enable a higher accuracy
* It’s effective for credit card fraud detection: Python uses a wide selection of tools to speed up complicated processes and make the right decision on time

Working on the credit card fraud detection project in Python, we will go through several steps:

1. Importing and preparing the data
2. Processing the data with Exploratory Data Analysis (EDA)
3. Splitting the data and fitting the model
4. Building 6 classification models
5. Checking our models with the help of 3 metrics

With that, let’s dive in, describing the process in detail and making code examples.

## Preparing the Data for Fraud Detection in Python

We start with reading the source data, studying the variables, and examining some samples. Our goal is to understand various columns of data, their features, and other necessary information.

Packages and libraries we normally use for a credit card fraud detection project in Python:

* Pandas
* NumPy
* Scikit-learn
* XGBoost

Let’s begin. We will apply Pandas to create a specific data frame for continuous use.



Our next steps will include further processing of transaction data, including the method called Exploratory Data Analysis or EDA.

We determine how many fraud and non-fraud actions there are. Running the Python code, we receive our first result.



**Result of fraud detection (Machine Learning, Python)**

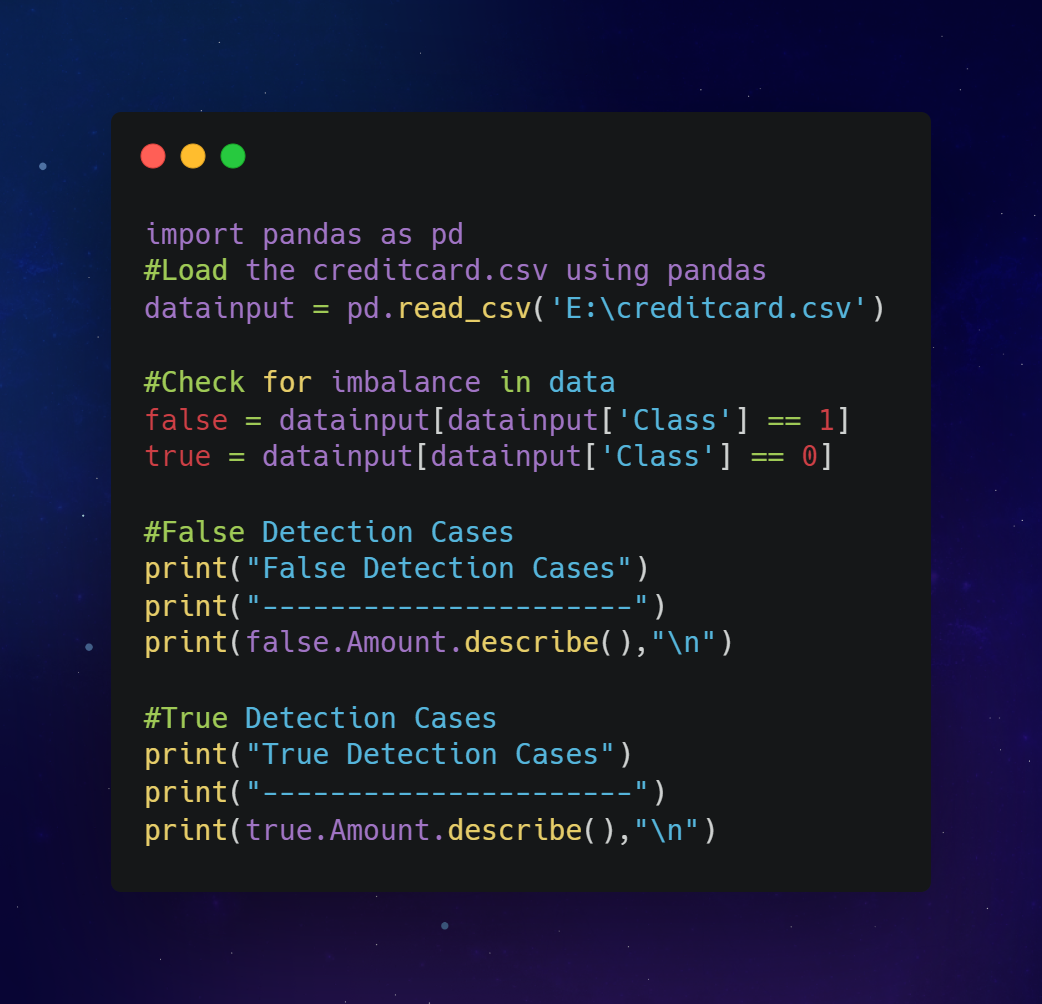


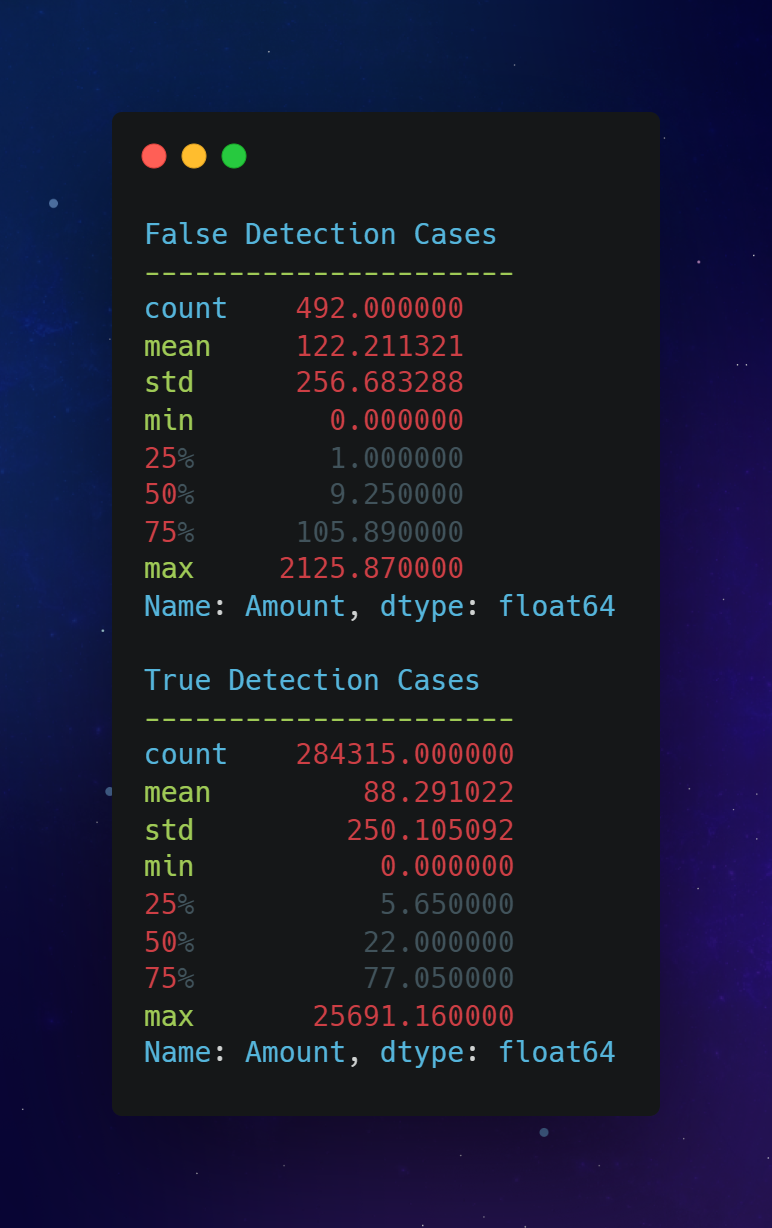
Next step – we investigate all the details about fraudulent transactions and non-fraudulent ones. What interest us here is a statistical picture that includes parameters like:

* Maximum value
* Minimum value
* Standard deviation of the mean
* Various percentiles

Using the method called ‘describe’, we find all of it with Python: fraud detection examples typically include all the statistics.

**Example in Python**



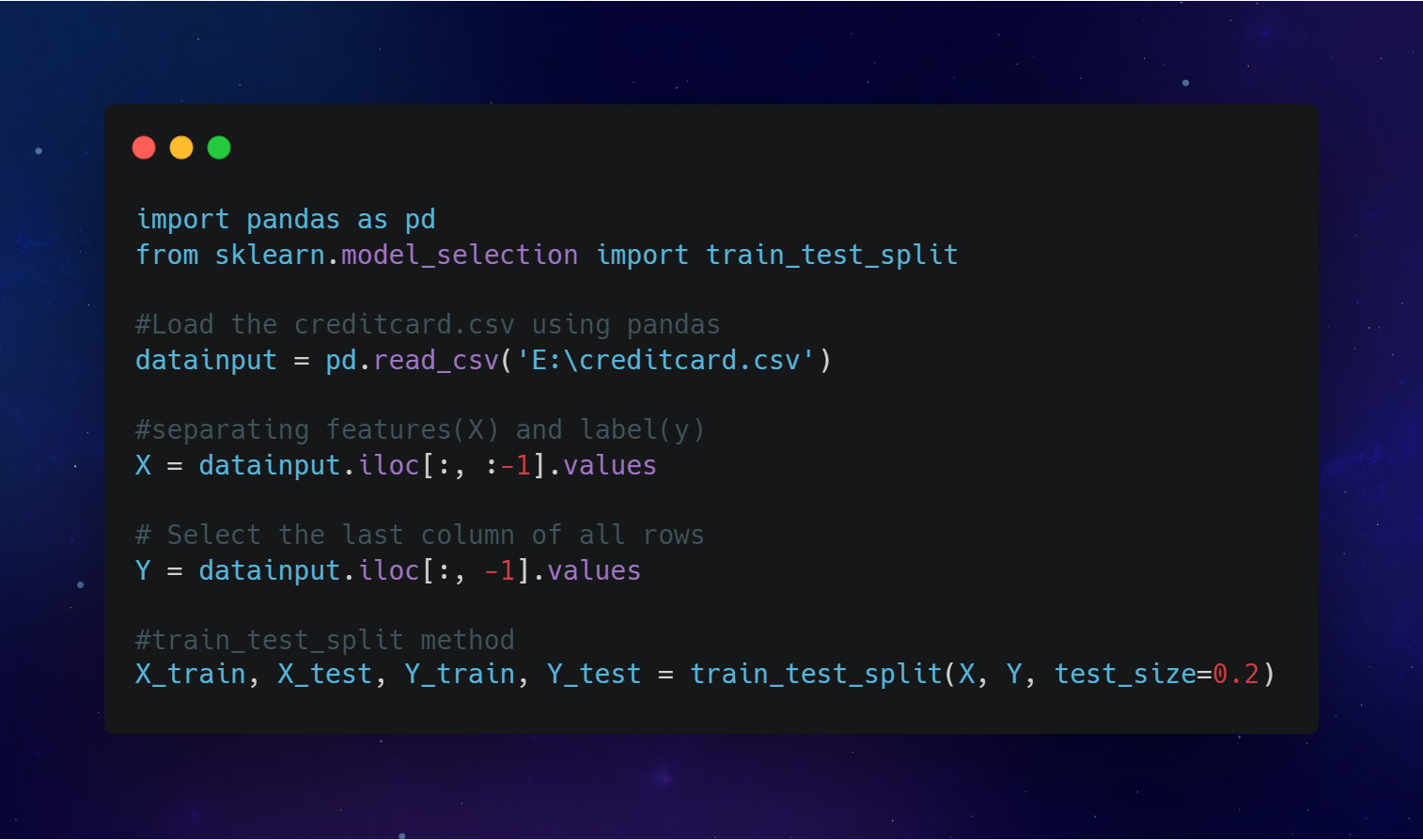
**Result of analysis for fraud detection (Machine Learning, Python)**

What we do next is called data split: defining two kinds of variables: dependent (Y) and independent (X). It is also called the ‘split x y test’ in data science.

Defined variables will help us split the data into two sets:

1. Test data
2. Training data

We will use the two sets to make a fraud detection model (Python) and evaluate the final results. In coding, we apply the ‘train\_test\_split’ algorithm to split the data efficiently.

**Example in Python**

## How we Build and Train Fraud Detection Model (Python)

In the previous step, we split the data set into two parts: test data and training data. As their names suggest, we use the first one to test the result while the second one is for training the model.

Fraud detection using Python enables us to apply the classification method and build different models. In the end, we will choose the ones that give the most accurate predictions.

6 classification models include:

* K-Nearest Neighbors (KNN)
* Logistic Regression
* Support Vector Machine (SVM)
* Random Forests
* XGBoost
* Decision Tree

Building the first five models, we apply an open-source library Scikit-learn. As for the XGBoost fraud detection model, Python supports the XGBoost package.

This is how we do the modeling.



## Evaluation for Python Fraud Detection: Example from our Experience

Having built the models, we start evaluating. But first, let’s define two important ML terms.

* True positives are results of predicting the positive class when a model does it correctly.
* False positives are results of predicting the positive class incorrectly.

We use both true and false positives to analyze datasets and detect fraudulent actions with Python.

After building 6 models that can recognize fraud using Python, it is vital to evaluate their quality.

We usually apply these three evaluation methods:

* Accuracy score
* F1\_score y test
* Decision tree
* Confusion matrix

Given the fact that our case is credit card fraud detection, machine learning (Python) can give us exhausting answers about any suspicious transaction.

**Accuracy score**is a simple and basic evaluation metric for classification models in Machine Learning.

Accuracy score = Number of correct predictions / Total number of predictions

How do you express it in percentages? Just multiply by 100.

**F1 Score** is a popular metric for evaluation in ML. It is deeply connected to metrics like recall and precision, representing the balance.

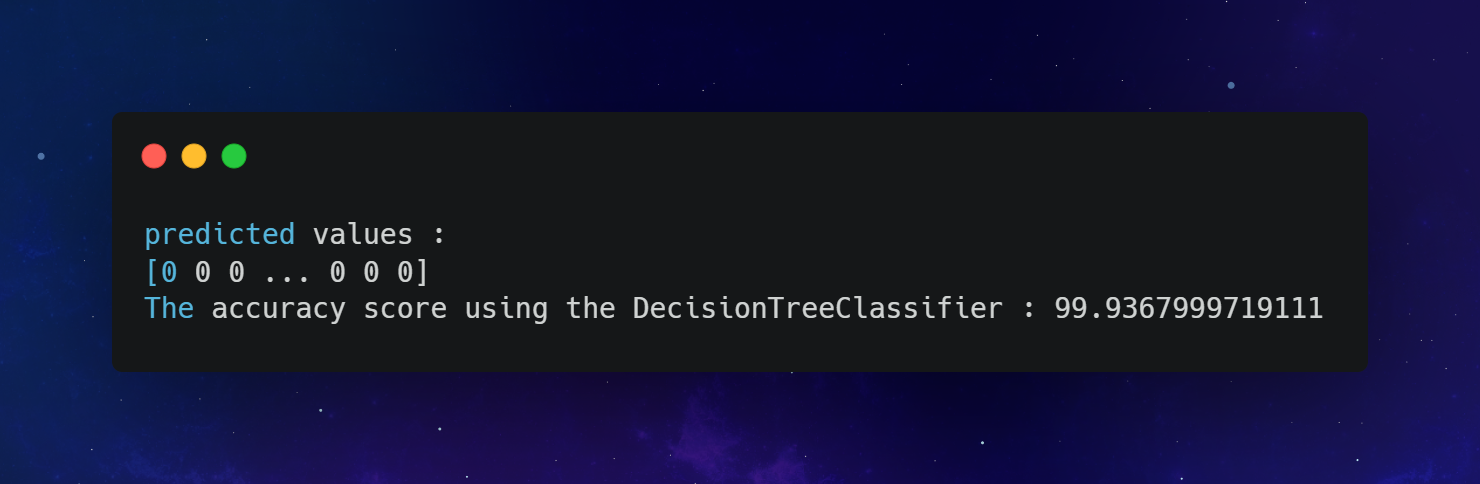
F1 score = 2( (precision \* recall) / (precision + recall) )

We can calculate the F1 score with Python using the f1\_score y test provided by the Scikit-learn package.

**Decision tree (DT)** is a powerful method to use in machine learning. One of the best things about it is that you can represent decision-making visually, like a tree with lots of branches.

Here is how we use the DT as an evaluation method in Python fraud detection: an example of the whole process.

**Example in Python**

**Result of evaluation for fraud detection (Machine Learning, Python)**

# CONCLUSION

In this project, we applied machine learning algorithms to predict genuine and fraudulent transactions. For doing so, we collected data from the Kaggle website which had around 284,807 transactions out of which only 492 were fraudulent. We learned that the data is highly imbalanced and that must be balanced by using sampling methods. Our main aim was to compare the best model i.e., Random Forest Classifier which comes under ensemble learning methods and is a bagging method with other three boosting methods that included AdaBoost, CatBoost, and XGBoost Classifiers. The result was clear that Catboost gave us high accuracy in very less time.