CREDIT CARD FRAUD DEDUCTION USING APPLIED DATA SCIENCE

PHASE-02(INNOVATION)

Data science is an interdisciplinary field that encompasses a set of techniques, processes, and methods for extracting valuable insights, knowledge, and patterns from data. It combines elements of statistics, computer science, domain expertise, and data engineering to collect, analyze, and interpret large and complex datasets. The ultimate goal of data science is to use data to inform decision-making, solve problems, and drive improvements in various domains, including business, healthcare, finance, and more. Data scientists use a combination of data analysis, machine learning, data visualization, and domain-specific expertise to uncover meaningful information from data and provide valuable insights for organizations and individuals.

1.CREDIT CARD FRAUD DETECTION USING DATA SCIENCE

Data science is used for credit card fraud detection because it offers powerful tools and Techniques to identify suspicious patterns and anomalies in large volumes of transaction Data. Here's why data science is particularly well-suited for this task:

1. Handling Big Data:

Credit card transactions generate massive amounts of data. Data science Techniques, including big data processing frameworks like Hadoop and Spark, Enable the efficient handling and analysis of this data.

2.Pattern Recognition:

Data science algorithms excel at recognizing patterns within data. They can Identify unusual behaviors or transactions that deviate from established patterns, Which is crucial for fraud detection.

3. Real-Time Analysis:

Many data science models can analyze transactions in real-time, allowing for Immediate detection and response to suspicious activities as they occur.

4.Adaptability:

Fraudsters constantly evolve their tactics, so fraud detection systems need to Adapt. Data science models can be regularly retrained to stay up-to-date with Emerging fraud patterns.

5.Anomaly Detection:

Anomaly detection techniques in data science can flag transactions that are Statistical outliers, helping to pinpoint potentially fraudulent activity.

6. False Positive Reduction:

Data science can help reduce false positives (legitimate transactions mistakenly identified as fraud), which is crucial to avoid inconveniencing genuine cardholders.

7. Ensemble Techniques:

Combining multiple machine learning models using ensemble methods can improve fraud detection accuracy and reduce false alarms.

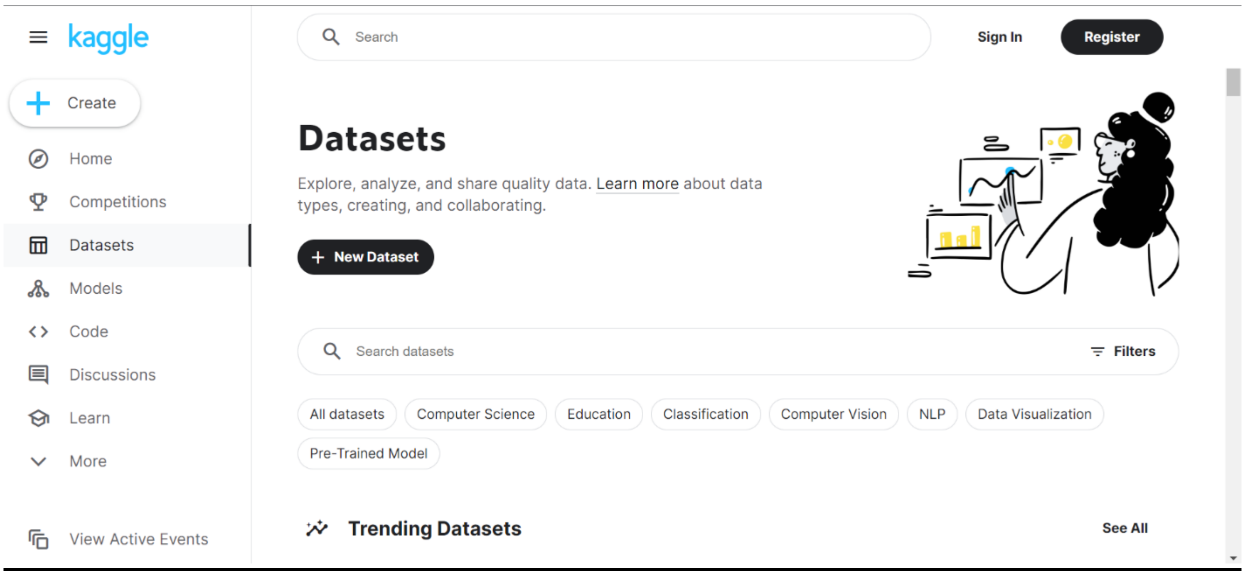
8. Predictive Analytics :

Data science can be used not only to detect ongoing fraud but also to predict future fraudulent activities based on historical data and trends.

2.SOURCE OF THE DATASET

This dataset is taken from the website name ”**kaggle**” and the link of the website is given below

Link: www.kaggle.com/data



Datasets

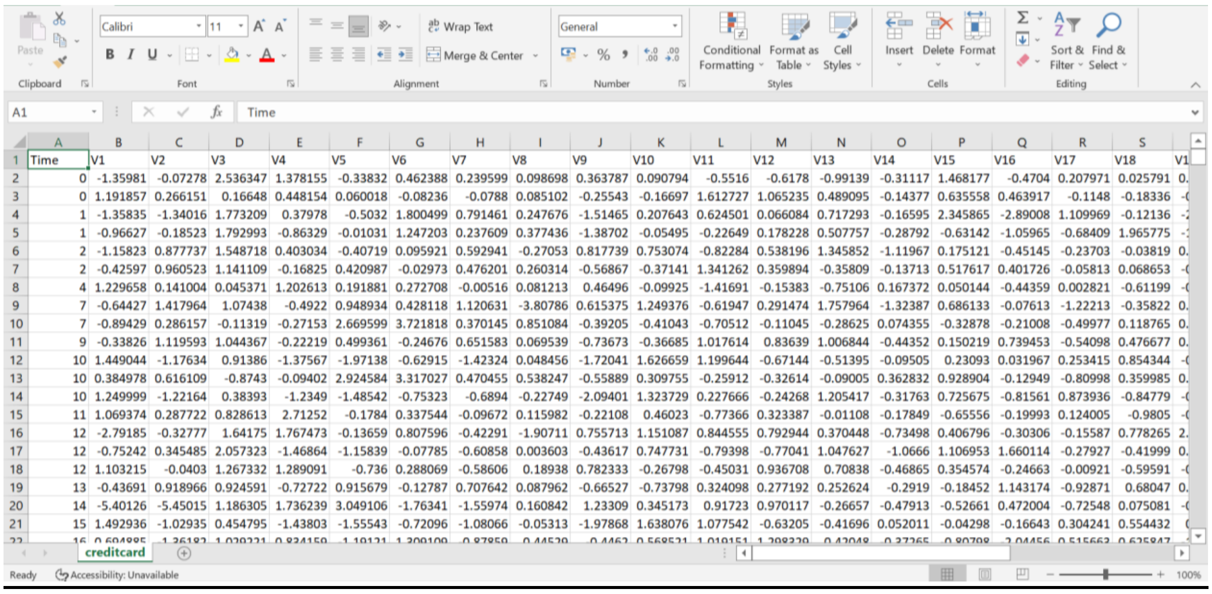
Explore, analyze, and share quality data. Learn more about data types, creating, and collaborating.

Trending Datasets

In this page search for your desired dataset about the credit card fraud deduction

The link for this project:

https://www.kaggle.com/datasets/mig-ulb/creditcardfraud



Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18
0	-1.35981	-0.07278	2.536347	1.378155	-0.33832	0.462388	0.239599	0.098698	0.363787	0.090794	-0.5516	-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791
0	1.191857	0.266151	0.16648	0.448154	0.060018	-0.08236	-0.0788	0.085102	-0.25543	-0.16697	1.612727	1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336
1	-1.35835	-1.34016	1.773209	0.37978	-0.5032	1.800499	0.791461	0.247676	-1.51465	0.207643	0.624501	0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136
1	-0.96627	-0.18523	1.792993	-0.86329	-0.01031	1.247203	0.237609	0.377436	-1.38702	-0.05495	-0.22649	0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775
2	-1.15823	0.877737	1.548718	0.403034	-0.40719	0.095921	0.592941	-0.27053	0.817739	0.753074	-0.82284	0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819
2	-0.42597	0.960523	1.141109	-0.16825	0.420987	-0.02973	0.476201	0.260314	-0.56867	-0.37141	1.341262	0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653
4	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.00516	0.081213	0.46496	-0.09925	-1.41691	-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199
7	-0.64427	1.417964	1.07438	-0.4922	0.948934	0.428118	1.120631	-3.80786	0.615375	1.249376	-0.61947	0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822
7	-0.89429	0.286157	-0.11319	-0.27153	2.669599	3.721818	0.370145	0.851084	-0.39205	-0.41043	-0.70512	-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765
9	-0.33826	1.119593	1.044367	-0.22219	0.499361	-0.24676	0.651583	0.069539	-0.73673	-0.36685	1.017614	0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677
10	1.449044	-1.17634	0.91386	-1.37567	-1.97138	-0.62915	-1.42324	0.048456	-1.72041	1.626659	1.199644	-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344
10	0.384978	0.616109	-0.8743	-0.09402	2.924584	3.317027	0.470455	0.538247	-0.55889	0.309755	-0.25912	-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985
10	1.249999	-1.22164	0.38393	-1.2349	-1.48542	-0.75323	-0.6894	-0.22749	-2.09401	1.323729	0.227666	-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779
11	1.069374	0.287722	0.828613	2.71252	-0.1784	0.337544	-0.09672	0.115982	-0.22108	0.46023	-0.77366	0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805
12	-2.79185	-0.32777	1.64175	1.767473	-0.13659	0.807596	-0.42291	-1.90711	0.755713	1.151087	0.844555	0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265
12	-0.75242	0.345485	2.057323	-1.46864	-1.15839	-0.07785	-0.60858	0.003603	-0.43617	0.747731	-0.79398	-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999
12	1.103215	-0.0403	1.267332	1.289091	-0.736	0.288069	-0.58606	0.18938	0.782333	-0.26798	-0.45031	0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591
13	-0.43691	0.918966	0.924591	-0.72722	0.915679	-0.12787	0.707642	0.087962	-0.66527	-0.73798	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047
14	-5.40126	-5.45015	1.186305	1.736239	3.049106	-1.76341	-1.55974	0.160842	1.23309	0.345173	0.91723	0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081
15	1.492936	-1.02935	0.454795	-1.43803	-1.55543	-0.72096	-1.08066	-0.05313	-1.97868	1.638076	1.077542	-0.63205	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432

3.DETAILS ABOUT DATASET

The dataset consists about the data's of the credit cards and the time in which the fraud has been taken place with the attributes

- In the given dataset for Credit card fraud detection,the rows are aligned from V1 to V28 and with amount and class.
- The columns are aligned with integers from 0 to 114.
- The values in dataset are both positive and negative.
- In the amount column, the highest value is 3828 and the least value is 0.75.
- In the class column,the values are almost equal to zero's and one's(Boolean values)
- There are almost 284808 tuples present

4.Libraries used in credit card fraud detection

We use the following libraries and frameworks in credit card fraud detection project.

1. Python – 3.x
2. Numpy – 1.19.2
3. Scikit-learn – 0.24.1
4. Matplotlib – 3.3.4
5. Imblearn – 0.8.0

How to download the libraries:

Download python 3.x

This document describes how to install Python 3.6 or 3.8 on Ubuntu Linux machines.To see which version of Python 3 you have installed, open a command prompt and run

```
$ python3 –version
```

If you are using Ubuntu 16.10 or newer, then you can easily install Python 3.6 with the following

commands:

```
$ sudo apt-get update

$ sudo apt-get install python3.6
```

If you're using another version of Ubuntu (e.g. the latest LTS release) or you want to use a more current

Python, we recommend using the deadsnakes PPA to install Python 3.8:

```
$ sudo apt-get install software-properties-common

$ sudo add-apt-repository ppa:deadsnakes/ppa

$ sudo apt-get update

$ sudo apt-get install python3.8
```

If you are using other Linux distribution, chances are you already have Python 3 pre-installed as well. If not, use your distribution's package manager. For example on Fedora, you would use dnf:

```
$ sudo dnf install python3
```

Download numpy 1.19.2

Installed Pythons found by C:\WINDOWS\py.exe Launcher for Windows

I use Python 3.6 for ESA SNAP snappy, so

```
PS C:\> cd $env:USERPROFILE\.snap\snap-python\

PS C:\Users\XXXXX\.snap\snap-python> py -3.6

Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import snappy
```

Numpy installation:

To install numpy on a Python.org 2 version for use with snappy:

```
PS C:\> py -3.6 -m pip install numpy
```

Collecting numpy

```
 Downloading
51986f6d0c5c2/numpy-1.19.5-cp36-cp36m-win_amd64.whl (13.2MB)
 100% |████████████████████████████████| 13.2MB 3.9MB/s
Installing collected packages: numpy
 The script f2py.exe is installed in 'D:\Python36\Scripts' which is not on PATH.
 Consider adding this directory to PATH or, if you prefer to suppress this warning, use –no-warn-script  location.
Successfully installed numpy-1.19.5
You are using pip version 18.1, however version 21.0.1 is available.
You should consider upgrading via the 'python -m pip install –upgrade pip' command.
```

Note that using py -M.N doesn't adjust the PATH

Download matplotlib

Matplotlib releases are available as wheel packages for macOS, Windows and Linux on PyPI. Install it

using pip:

```
Python -m pip install -U pip
Python -m pip install -U matplotlib
```

If this command results in Matplotlib being compiled from source and there's trouble with the compilation, you can add –prefer-binary to select the newest version of Matplotlib for which there is a precompiled wheel for that.

Install an official release

Matplotlib releases are available as wheel packages for macOS, Windows and Linux on PyPI. Install it

using pip:

```
Python -m pip install -U pip
Python -m pip install -U matplotlib
```

If this command results in Matplotlib being compiled from source and there's trouble with the

compilation, you can add –prefer-binary to select the newest version of Matplotlib for which there is a

precompiled wheel for your OS and Python.

Third-party distributions

Various third-parties provide Matplotlib for their environments.

Conda packages

Matplotlib is available both via the anaconda main channel

Conda install matplotlib

As well as via the conda-forge community channel

```
Conda install -c conda-forge matplotlib
```

Downloading Imblearn 0.8.0

Imbalanced-learn

Imbalanced-learn is a python package offering a number of re-sampling techniques commonly used in datasets showing strong between-class imbalance. It is compatible with scikit-learn and is part of scikit learn-contrib projects.

Documentation

Installation documentation, API documentation, and examples can be found on the documentation.

Dependencies

Imbalanced-learn is tested to work under Python 3.6+. The dependency requirements are based on the

last scikit-learn release:

Scipy(>=0.19.1)

Numpy(>=1.13.3)

Scikit-learn(>=0.23)

Joblib(>=0.11)

Keras 2 (optional)

Tensorflow (optional)

Additionally, to run the examples, you need matplotlib(>=2.0.0) and pandas(>=0.22).

5.How to Train/Test

Train/Test is a method to measure the accuracy of your model .It is called Train/Test because you split the data set into two sets: a training set and a testing set.

80% for training, and 20% for testing.

You train the model using the training set.

You test the model using the testing set.

Train the model means create the model.

Test the model means test the accuracy of the model.

Split Into Train/Test

The training set should be a random selection of 80% of the original data.

The testing set should be the remaining 20%.

```
train_x = x[:80]

train_y = y[:80]

test_x = x[80:]

test_y = y[80:]
```

Display the Training Set

Display the same scatter plot with the training set.

Example

```
plt.scatter (train_x, train_y)

plt.show()
```

6.EXTRA EXPLANATION

ALGORITHM USED IN CREDIT CARD FRAUD DEDUCTION

1.Decision Tree

2.Predictive Analytics and Algorithms

3.Clustering Techniques

4.K-Nearest Neighbour Algorithm

5.Neural Networks

6.Naive Bayes Classifiers

7.Support Vector Machines (SVMs)

7.METRICS USED FOR CREDIT CARD FRAUD DETECTION

The most frequent metric used is ‘Accuracy’. High Accuracy doesn’t mean that the model is performing better in all the situations. It is not always considered to be accurate as it sometimes is misleading in some situations like imbalanced class datasets.

ACCURACY METRICS

Accuracy is a metric that generally describes how the model performs across all classes. It is useful when all classes are of equal importance. It is calculated as the ratio between the number of correct predictions to the total number of predictions

CODE FOR ACCURACY OF A CLASSIFER

Importing All necessary libraries

from sklearn.metrics import accuracy_score

Calculating the accuracy of classifier

```
print(Accuracy of the classifier is: {accuracy_score(y_test, predictions)}")
```

IMPLEMENTATION OF ACCURACY METRICS

The Accuracy score is calculated by dividing the number of correct predictions by the total prediction number.

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

Another formula:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$