Machine Learning :

Machine Learning is a type of AI techniques used by data scientists that allow computers to learn from data. With these techniques the computer will use algorithms to extract data and predict future trends.

Here I am using a Fraud dataset that contain historical infromation about fradudulent transcation which can used to detect fraud in online payments.

In [1]:
```
#import neccessary libraries
import pandas as pd
import numpy as np
```

DATA PREPARATION AND CLEANING :

In [2]:
```
df = pd.read_csv("onlinefraud.csv")
df
```

Out[2]:

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	i
0	1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.00	0.00	0	
1	1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.00	0.00	0	
2	1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065	0.00	0.00	1	
3	1	CASH_OUT	181.00	C840083671	181.00	0.00	C38997010	21182.00	0.00	1	
4	1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703	0.00	0.00	0	
...	...	...	...	...	...	...	...	...	...	...	...
1048570	95	CASH_OUT	132557.35	C1179511630	479803.00	347245.65	C435674507	484329.37	616886.72	0	
1048571	95	PAYMENT	9917.36	C1956161225	90545.00	80627.64	M668364942	0.00	0.00	0	
1048572	95	PAYMENT	14140.05	C2037964975	20545.00	6404.95	M1355182933	0.00	0.00	0	
1048573	95	PAYMENT	10020.05	C1633237354	90605.00	80584.95	M1964992463	0.00	0.00	0	
1048574	95	PAYMENT	11450.03	C1264356443	80584.95	69134.92	M677577406	0.00	0.00	0	

1048575 rows × 11 columns

In [3]:
```
# check for missing values
df.isnull().sum()
```

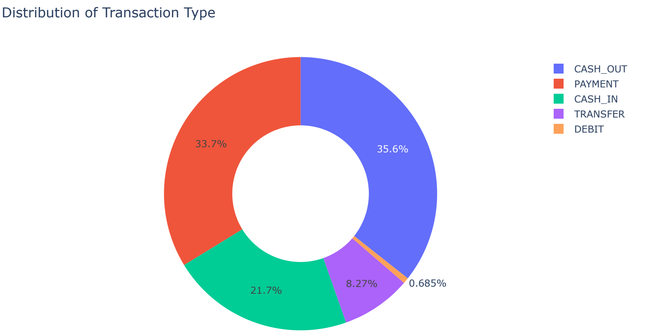
Out[3]:
```
step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
```

In [4]:
```
# type of transactions
df.type.value_counts()
```

Out[4]:
```
CASH_OUT    373641
PAYMENT     353873
CASH_IN     227130
TRANSFER     86753
DEBIT         7178
Name: type, dtype: int64
```

In [5]:
```
# show visual of transaction types
type = df['type'].value_counts()
transactions = type.index
quantity = type.values

import plotly.express as px
figure = px.pie(df,
             values=quantity,
             names=transactions,hole = 0.5,
             title="Distribution of Transaction Type")
figure.show()
```



In [6]:
```
# check correlation between features and "isFraud" column.

correlation = df.corr()
print(correlation["isFraud"].sort_values(ascending=False))
```

```
isFraud           1.000000
amount            0.128862
step              0.045030
oldbalanceOrg     0.003829
newbalanceDest   -0.000495
oldbalanceDest   -0.007552
newbalanceOrig   -0.009438
isFlaggedFraud         NaN
Name: isFraud, dtype: float64
```

In [7]:
```
# transform categorical features into numerical
df["type"] = df["type"].map({"CASH_OUT": 1, "PAYMENT": 2,
                             "CASH_IN": 3, "TRANSFER": 4,
                             "DEBIT": 5})

#transform values of "isFraud" to Fraud or No Fraud
df["isFraud"] = df["isFraud"].map({0: "No Fraud", 1: "Fraud"})
df
```

Out[7]:

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlagg
0	1	2	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.00	0.00	No Fraud	
1	1	2	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.00	0.00	No Fraud	
2	1	4	181.00	C1305486145	181.00	0.00	C553264065	0.00	0.00	Fraud	
3	1	1	181.00	C840083671	181.00	0.00	C38997010	21182.00	0.00	Fraud	
4	1	2	11668.14	C2048537720	41554.00	29885.86	M1230701703	0.00	0.00	No Fraud	
...	...	...	...	...	...	...	...	...	...	...	...
1048570	95	1	132557.35	C1179511630	479803.00	347245.65	C435674507	484329.37	616886.72	No Fraud	
1048571	95	2	9917.36	C1956161225	90545.00	80627.64	M668364942	0.00	0.00	No Fraud	
1048572	95	2	14140.05	C2037964975	20545.00	6404.95	M1355182933	0.00	0.00	No Fraud	
1048573	95	2	10020.05	C1633237354	90605.00	80584.95	M1964992463	0.00	0.00	No Fraud	
1048574	95	2	11450.03	C1264356443	80584.95	69134.92	M677577406	0.00	0.00	No Fraud	

1048575 rows × 11 columns

BUILDING MODEL

In [8]:
```
# split data into training set and test set
from sklearn.model_selection import train_test_split
x = np.array(df[["type", "amount", "oldbalanceOrg", "newbalanceOrig"]])
y = np.array(df[["isFraud"]])
```

In [9]:
```
# train the model
from sklearn.tree import DecisionTreeClassifier
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.10, random_state=42)
model = DecisionTreeClassifier()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))
```

```
0.9994277975929352
```

In [10]:
```
# classiy whether a transaction is fraud or not
features = np.array([[4, 9000.60, 9000.60, 0.0]])
print(model.predict(features))
```

```
['Fraud']
```