In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split, cross_val_score, KFold,
from sklearn.metrics import accuracy_score, classification_report, confusion_m
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import GradientBoostingClassifier
```

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

In [3]:
```
df.head(10)
```

Out[3]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	(
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-(
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-'
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-'
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	(
5	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.260314	-(
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	0.081213	(
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.807864	(
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.851084	-(
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.069539	-(

10 rows × 31 columns

In [4]:
```
df.shape
```

Out[4]:
```
(284807, 31)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64  
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

In [6]:
```
df.dtypes
```

Out[6]:
```
Time      float64
V1        float64
V2        float64
V3        float64
V4        float64
V5        float64
V6        float64
V7        float64
V8        float64
V9        float64
V10       float64
V11       float64
V12       float64
V13       float64
V14       float64
V15       float64
V16       float64
V17       float64
V18       float64
V19       float64
V20       float64
V21       float64
V22       float64
V23       float64
V24       float64
V25       float64
V26       float64
V27       float64
V28       float64
Amount    float64
Class       int64
dtype: object
```

In [7]:
```
df.isna().sum().sort_values()
```

Out[7]:
```
Time      0
V28       0
V27       0
V26       0
V25       0
V24       0
V23       0
V22       0
V21       0
V20       0
V19       0
V18       0
V17       0
V16       0
Amount    0
V15       0
V13       0
V12       0
V11       0
V10       0
V9        0
V8        0
V7        0
V6        0
V5        0
V4        0
V3        0
V2        0
V1        0
V14       0
Class     0
dtype: int64
```

In [9]:
```
df.describe()
```

Out[9]:

	Time	V1	V2	V3	V4	V5
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	3.918649e-15	5.682686e-16	-8.761736e-15	2.811118e-15	-1.552103e-15
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01

8 rows × 31 columns

In [10]:
```
df.describe().T
```

Out[10]:

	count	mean	std	min	25%	50%	
Time	284807.0	9.481386e+04	47488.145955	0.000000	54201.500000	84692.000000	139320
V1	284807.0	3.918649e-15	1.958696	-56.407510	-0.920373	0.018109	1
V2	284807.0	5.682686e-16	1.651309	-72.715728	-0.598550	0.065486	0
V3	284807.0	-8.761736e-15	1.516255	-48.325589	-0.890365	0.179846	1
V4	284807.0	2.811118e-15	1.415869	-5.683171	-0.848640	-0.019847	0
V5	284807.0	-1.552103e-15	1.380247	-113.743307	-0.691597	-0.054336	0
V6	284807.0	2.040130e-15	1.332271	-26.160506	-0.768296	-0.274187	0
V7	284807.0	-1.698953e-15	1.237094	-43.557242	-0.554076	0.040103	0
V8	284807.0	-1.893285e-16	1.194353	-73.216718	-0.208630	0.022358	0
V9	284807.0	-3.147640e-15	1.098632	-13.434066	-0.643098	-0.051429	0
V10	284807.0	1.772925e-15	1.088850	-24.588262	-0.535426	-0.092917	0
V11	284807.0	9.289524e-16	1.020713	-4.797473	-0.762494	-0.032757	0
V12	284807.0	-1.803266e-15	0.999201	-18.683715	-0.405571	0.140033	0
V13	284807.0	1.674888e-15	0.995274	-5.791881	-0.648539	-0.013568	0
V14	284807.0	1.475621e-15	0.958596	-19.214325	-0.425574	0.050601	0
V15	284807.0	3.501098e-15	0.915316	-4.498945	-0.582884	0.048072	0
V16	284807.0	1.392460e-15	0.876253	-14.129855	-0.468037	0.066413	0
V17	284807.0	-7.466538e-16	0.849337	-25.162799	-0.483748	-0.065676	0
V18	284807.0	4.258754e-16	0.838176	-9.498746	-0.498850	-0.003636	0
V19	284807.0	9.019919e-16	0.814041	-7.213527	-0.456299	0.003735	0
V20	284807.0	5.126845e-16	0.770925	-54.497720	-0.211721	-0.062481	0
V21	284807.0	1.473120e-16	0.734524	-34.830382	-0.228395	-0.029450	0
V22	284807.0	8.042109e-16	0.725702	-10.933144	-0.542350	0.006782	0
V23	284807.0	5.282512e-16	0.624460	-44.807735	-0.161846	-0.011193	0
V24	284807.0	4.456271e-15	0.605647	-2.836627	-0.354586	0.040976	0
V25	284807.0	1.426896e-15	0.521278	-10.295397	-0.317145	0.016594	0
V26	284807.0	1.701640e-15	0.482227	-2.604551	-0.326984	-0.052139	0
V27	284807.0	-3.662252e-16	0.403632	-22.565679	-0.070840	0.001342	0
V28	284807.0	-1.217809e-16	0.330083	-15.430084	-0.052960	0.011244	0
Amount	284807.0	8.834962e+01	250.120109	0.000000	5.600000	22.000000	77

	count	mean	std	min	25%	50%	
Class	284807.0	1.727486e-03	0.041527	0.000000	0.000000	0.000000	0

In [11]:
```
f = df[df['Class'] == 1]
v = df[df['Class'] == 0]
print("There are {} farudlent transactions".format(f.shape[0]))
print('There are {} valid transactions'.format(v.shape[0]))
```

```
There are 492 farudlent transactions
There are 284315 valid transactions
```

In [12]:
```
f.Amount.describe()
```

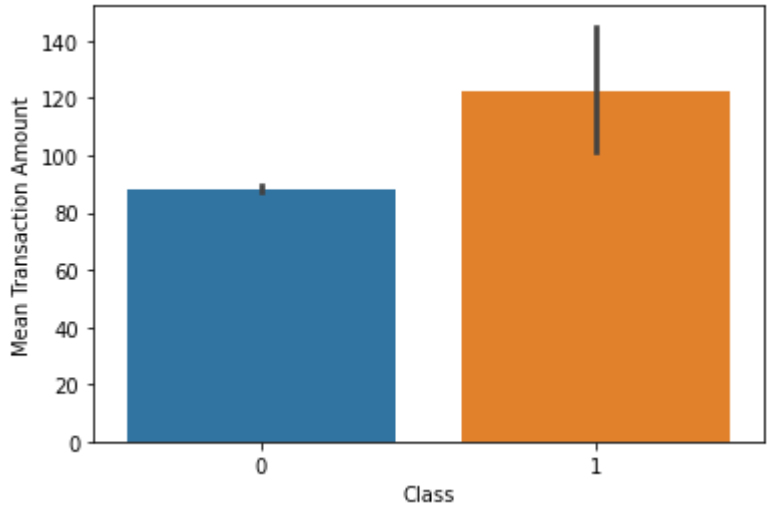
Out[12]:
```
count     492.000000
mean      122.211321
std       256.683288
min         0.000000
25%         1.000000
50%         9.250000
75%       105.890000
max      2125.870000
Name: Amount, dtype: float64
```

In [13]:
```
v.Amount.describe()
```

Out[13]:
```
count    284315.000000
mean         88.291022
std         250.105092
min           0.000000
25%           5.650000
50%          22.000000
75%          77.050000
max       25691.160000
Name: Amount, dtype: float64
```

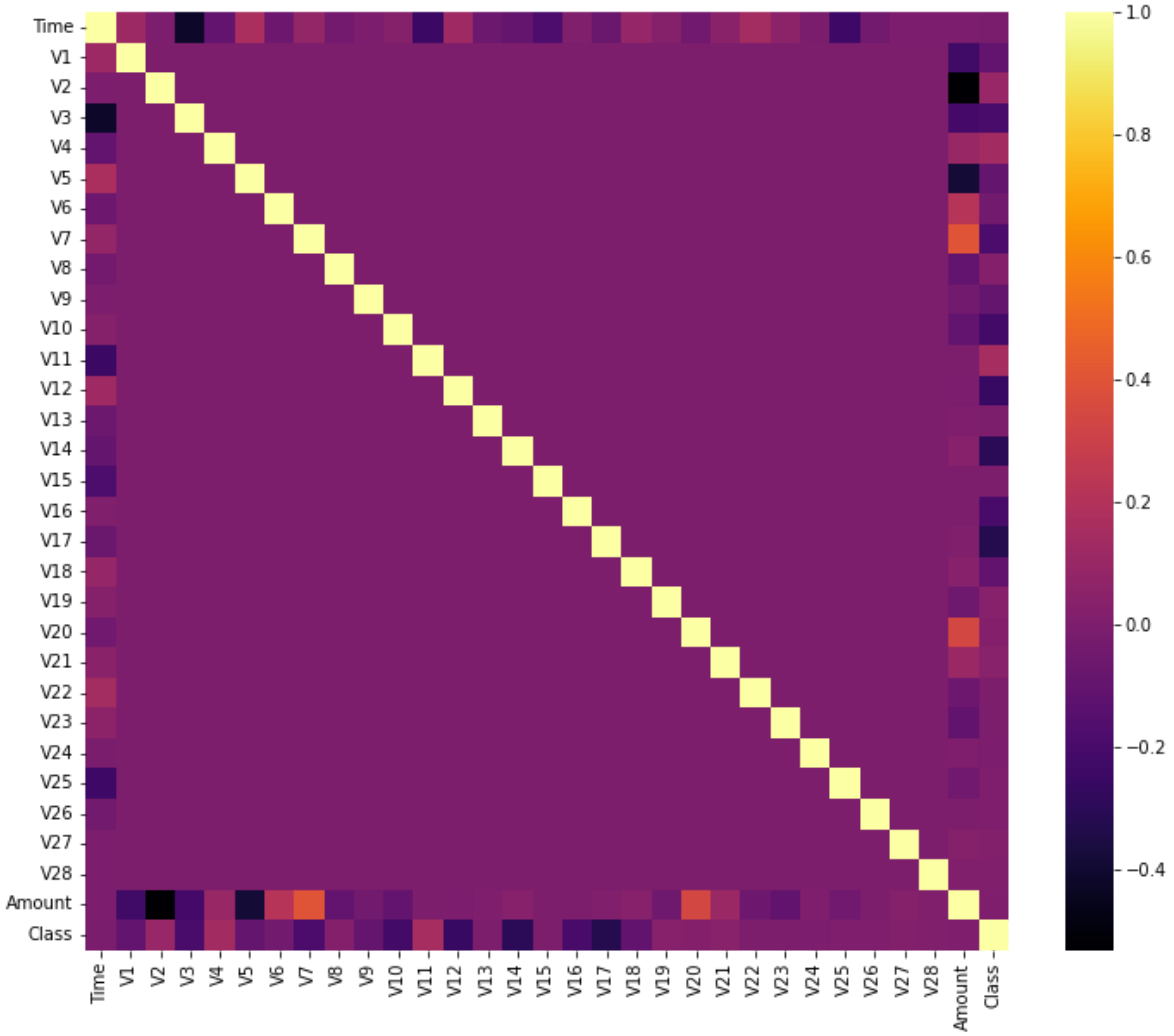
In [14]:
```
sns.barplot(data = df, x = 'Class', y = 'Amount')
plt.ylabel("Mean Transaction Amount")
```

Out[14]:
```
Text(0, 0.5, 'Mean Transaction Amount')
```



In [15]:
```
corr = df.corr()
fig = plt.figure(figsize = (12,10))
sns.heatmap(corr, cmap = 'inferno')
```

Out[15]: <AxesSubplot:>



In [16]:
```
scaler = StandardScaler()
df['Amount'] = scaler.fit_transform(df[['Amount']])
```

In [17]:
```
X = df.drop('Class', axis = 1).values
y = df['Class'].values
```

In [18]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random
```

In [ ]:
```
models = {'KNN': KNeighborsClassifier(),
          'LogReg': LogisticRegression(),
          'dtree': DecisionTreeClassifier(),
          'rforest': RandomForestClassifier(),
          'GB': GradientBoostingClassifier()
         }
names = []
scores = []
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    score = accuracy_score(y_pred, y_test)
    names.append(name)
    scores.append(score)
```

In [23]:
```
results = pd.DataFrame({'model': names, 'score': scores})
results.sort_values(by='score', ascending=False)
```

Out[23]:

	model	score
0	rforest	0.999579