

```
In [2]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [3]: df = pd.read_csv('creditcard.csv')
```

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

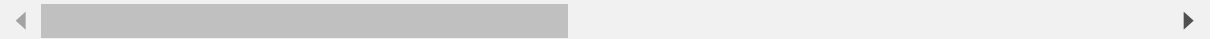
```
Out[4]: Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
              'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
              'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
              'Class'],
              dtype='object')
```

```
In [5]: df.head(3)
```

```
Out[5]:
```

	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	0.3
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.2
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.5

3 rows × 31 columns



```
In [6]: df.info()
```

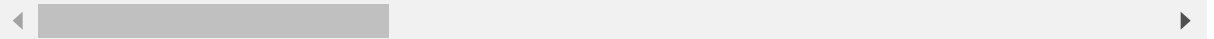
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Time        284807 non-null  float64
1   V1          284807 non-null  float64
2   V2          284807 non-null  float64
3   V3          284807 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 [7]: df.describe()
```

```
Out[7]:
```

	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	1.168375e-15	3.416908e-16	-1.379537e-15	2.074095e-15	9.604066e-16
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 [8]: df.isnull().sum()
```

```
Out[8]: Time      0
V1              0
V2              0
V3              0
V4              0
V5              0
V6              0
V7              0
V8              0
V9              0
V10             0
V11             0
V12             0
V13             0
V14             0
V15             0
V16             0
V17             0
V18             0
V19             0
V20             0
V21             0
V22             0
V23             0
V24             0
V25             0
V26             0
V27             0
V28             0
Amount          0
Class           0
dtype: int64
```

```
In [ ]: sns.pairplot(df,hue='Class',palette='Set1')
```

## Train Test and Split

```
In [10]: from sklearn.model_selection import train_test_split
```

```
In [11]: X = df.drop('Class',axis=1)
y = df['Class']
```

```
In [12]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
```

```
In [ ]:
```

## Decision Tress

```
In [13]: from sklearn.tree import DecisionTreeClassifier
```

```
In [14]: dtree = DecisionTreeClassifier(criterion='entropy', random_state=0)
```

```
In [15]: dtree.fit(X_train,y_train)
```

```
Out[15]: DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

## Prediction and Evaluation

```
In [16]: predictions = dtree.predict(X_test)
```

```
In [17]: predictions
```

```
Out[17]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

```
In [18]: from sklearn.metrics import classification_report,confusion_matrix
```

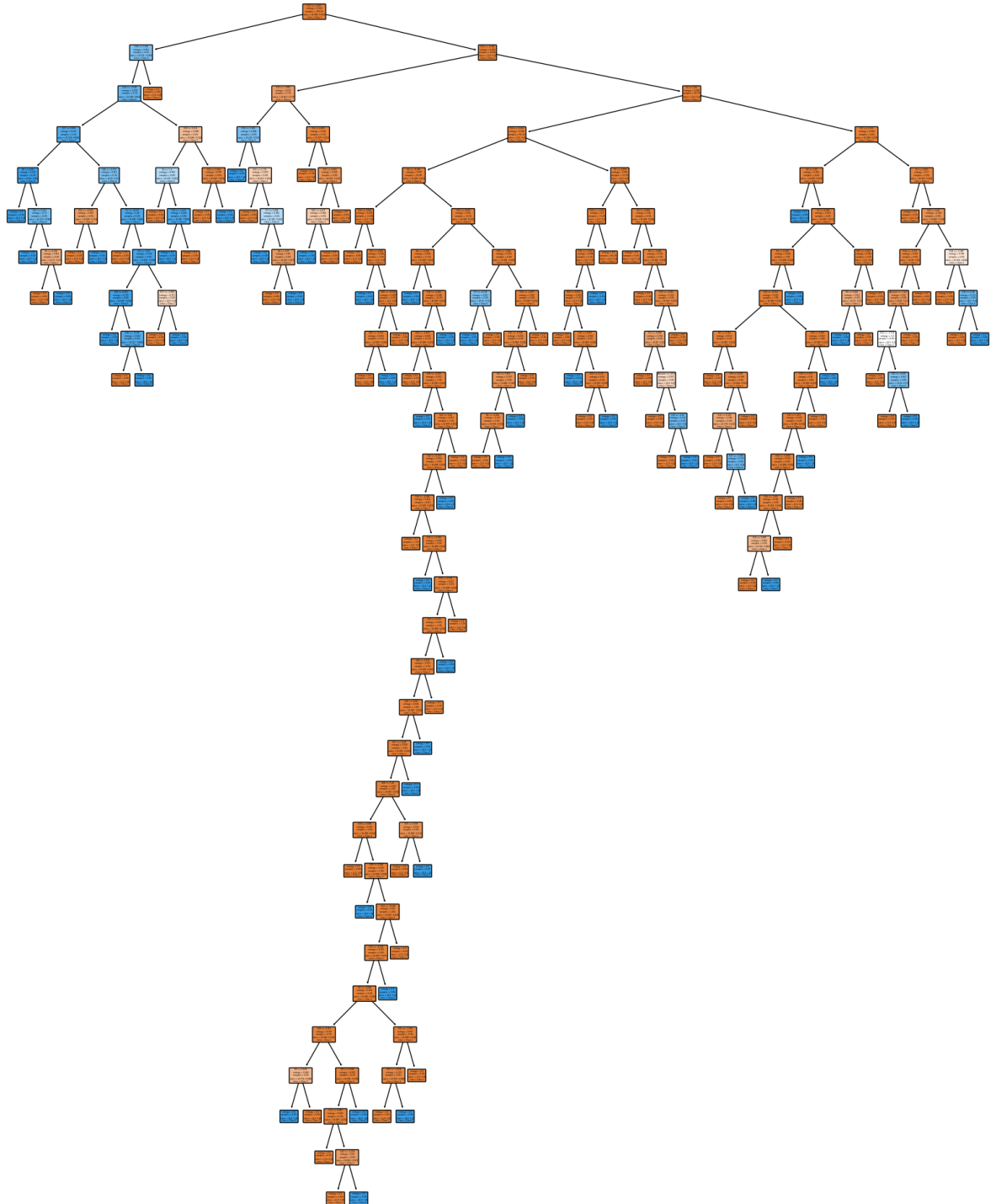
```
In [19]: print(classification_report(y_test,predictions))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	85285
1	0.83	0.81	0.82	158
accuracy			1.00	85443
macro avg	0.92	0.90	0.91	85443
weighted avg	1.00	1.00	1.00	85443

```
In [22]: print(confusion_matrix(y_test,predictions))
```

```
[[85259  26]
 [  30 128]]
```

```
In [24]: from sklearn import tree
plt.figure(figsize=(20,25))
tree.plot_tree(dtree,feature_names=X.columns,class_names=['Class-1', 'Class-0']
               filled=True, # Adds color according to class
               proportion=True
             )
plt.show()
```



In [ ]:

