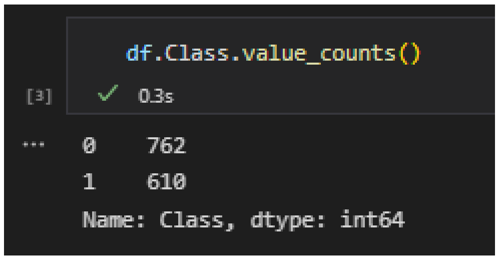
A. Dataset overview

Ans: Overall, datasets play a crucial role in the field of data science and are used in a wide range of applications, from simple data visualization to advanced machine learning algorithms.

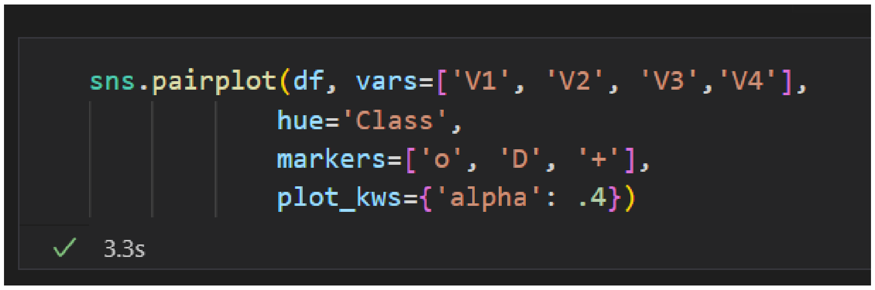
B. Data Preprocessing and Exploratory Data Analysis (EDA)



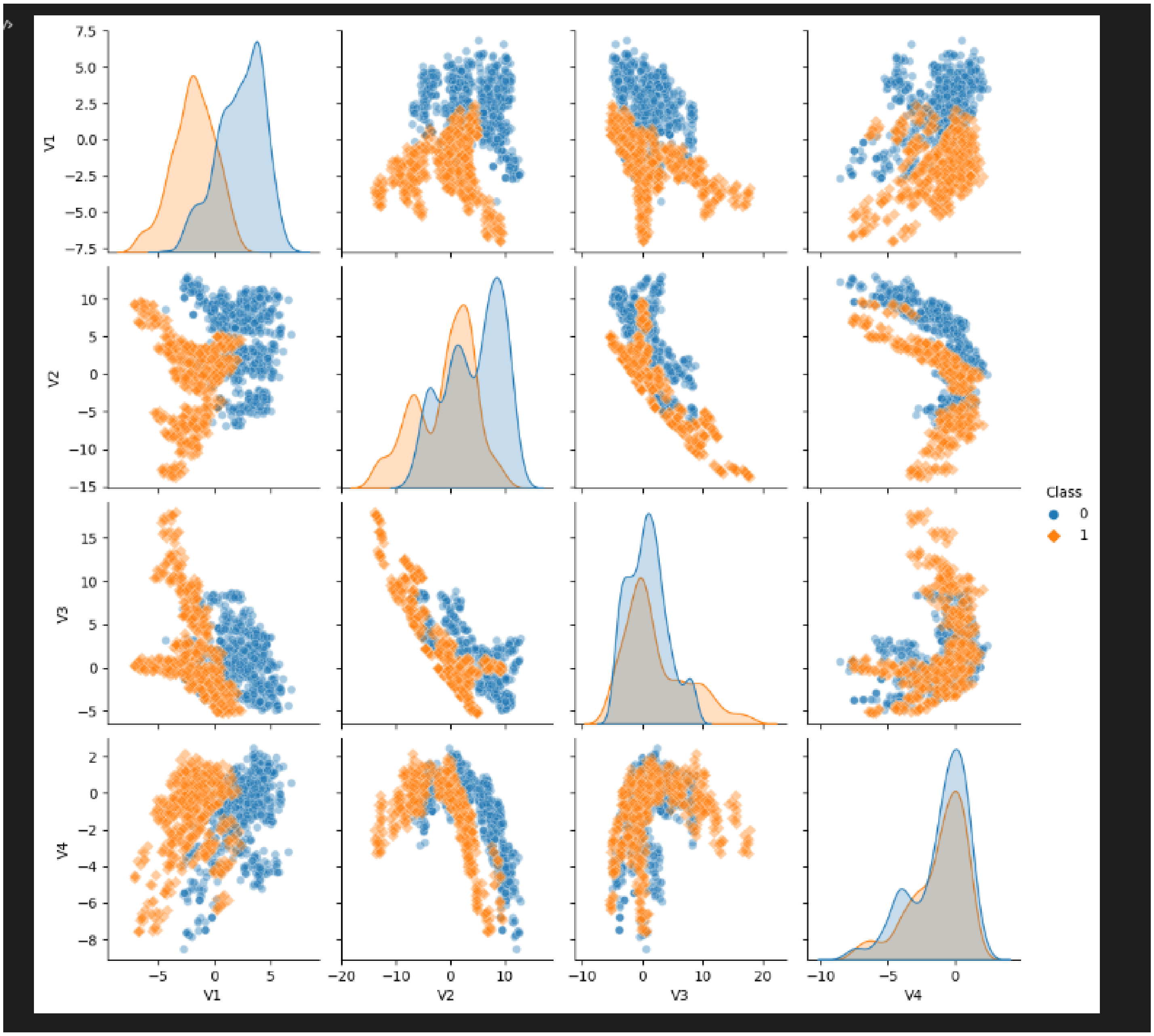
```
[3]  df.Class.value_counts()
     ✓ 0.3s

...  0    762
     1    610
     Name: Class, dtype: int64
```

We analyze the data by using min, max, describe and column to find the relationship of all data. And then we focus on the Class column As you can see in the picture we know that the value in the class column has only 2 type which is 1 and 0. So, the value shouldn’t greater than 1

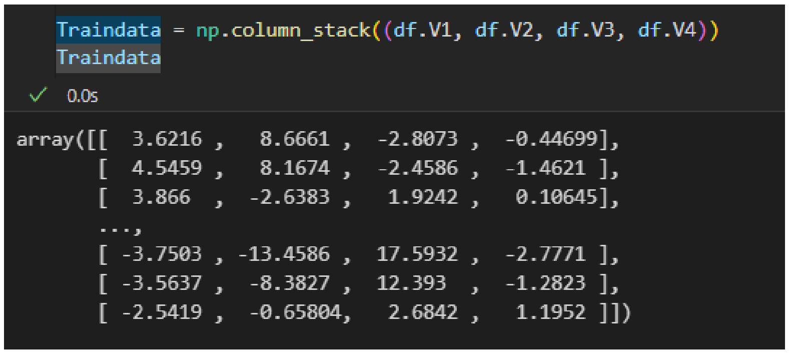


```
sns.pairplot(df, vars=['V1', 'V2', 'V3','V4'],
             hue='Class',
             markers=['o', 'D', '+'],
             plot_kws={'alpha': .4})
✓ 3.3s
```



Then we analyze the data byusing sns.pairplot function to see visual representation of the data and we can see that there are two type of cluster the data cluster scatter that is blue are belong to class 0 and The cluster scatter that is orange are belong to class 1

C. Training Model

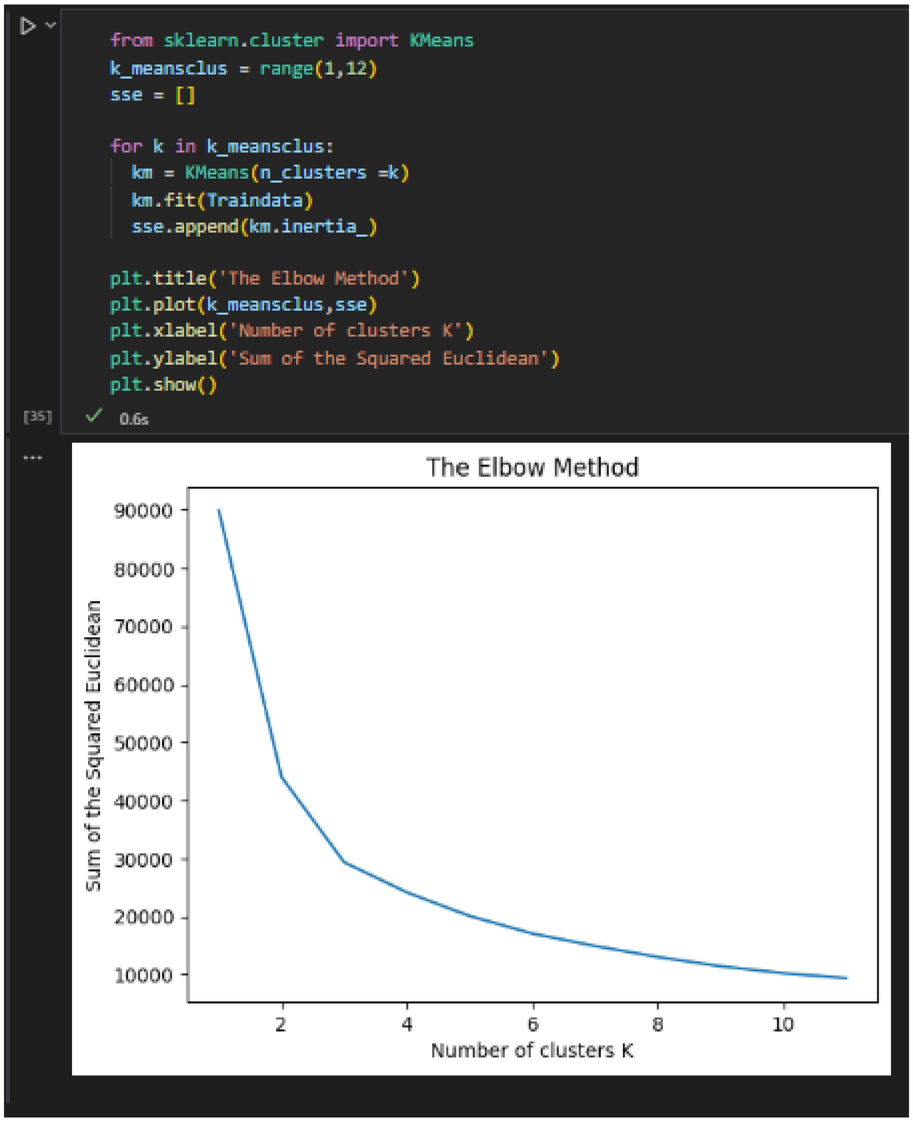


```
Traindata = np.column_stack((df.V1, df.V2, df.V3, df.V4))
Traindata
```

```
✓ 0.0s
```

```
array([[  3.6216 ,   8.6661 ,  -2.8073 ,  -0.44699],
       [  4.5459 ,   8.1674 ,  -2.4586 ,  -1.4621 ],
       [  3.866  ,  -2.6383 ,   1.9242 ,   0.10645],
       ...,
       [ -3.7503 , -13.4586 ,  17.5932 ,  -2.7771 ],
       [ -3.5637 ,  -8.3827 ,  12.393  ,  -1.2823 ],
       [ -2.5419 ,  -0.65804,   2.6842 ,   1.1952 ]])
```

First we make the training data Which is the one that we will be training or predicting by using this method
Then we want to find the optimal value of K. The value chosen for K is viewed at the “elbow junction of the graph”.



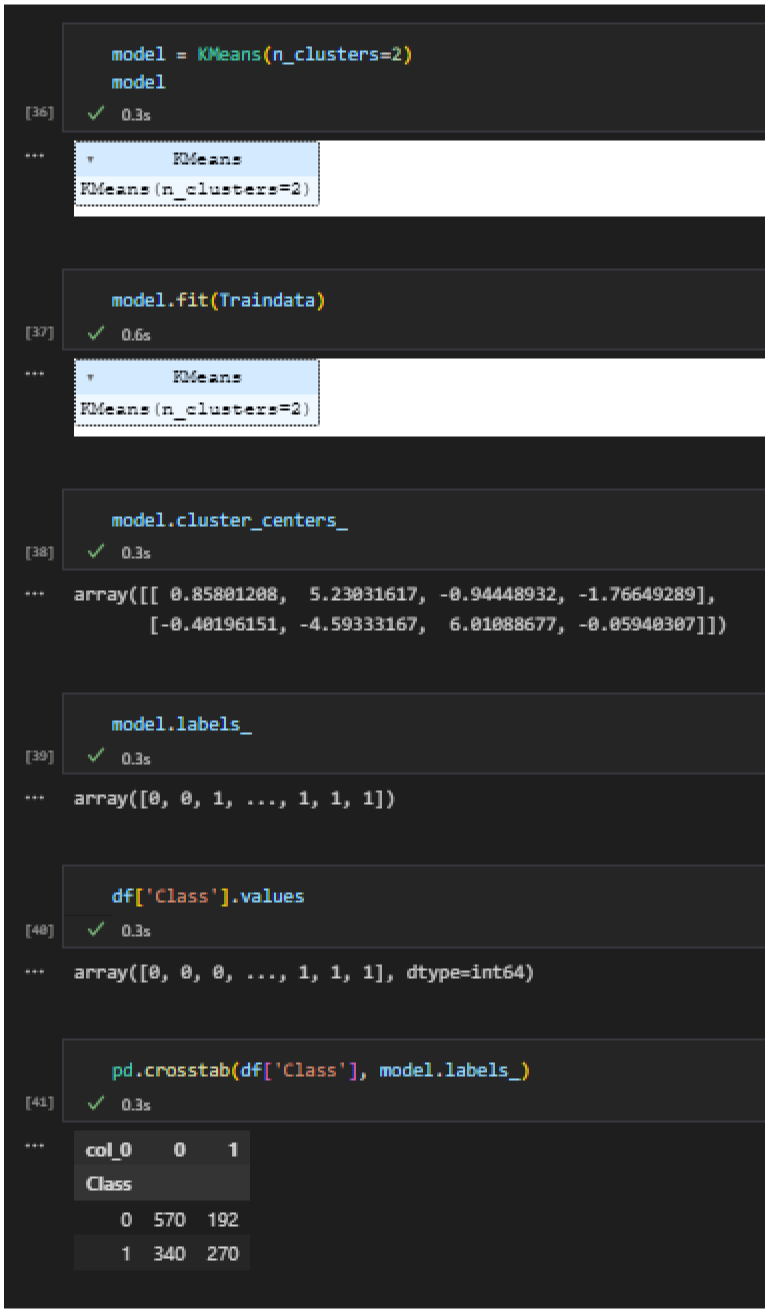
```
from sklearn.cluster import KMeans
k_meansclus = range(1,12)
sse = []

for k in k_meansclus:
  km = KMeans(n_clusters =k)
  km.fit(Traindata)
  sse.append(km.inertia_)

plt.title('The Elbow Method')
plt.plot(k_meansclus,sse)
plt.xlabel('Number of clusters K')
plt.ylabel('Sum of the Squared Euclidean')
plt.show()
```

```
[35]  ✓ 0.6s
```

After that we use the K-Value estimated to train our model and compute out the value.



```
model = KMeans(n_clusters=2)
model
```
[36] ✓ 0.3s

```
KMeans
KMeans(n_clusters=2)
```

```
model.fit(Traindata)
```
[37] ✓ 0.6s

```
KMeans
KMeans(n_clusters=2)
```

```
model.cluster_centers_
```
[38] ✓ 0.3s

```
array([[ 0.85801208,  5.23031617, -0.94448932, -1.76649289],
       [-0.40196151, -4.59333167,  6.01088677, -0.05940307]])
```

```
model.labels_
```
[39] ✓ 0.3s

```
array([0, 0, 1, ..., 1, 1, 1])
```

```
df['Class'].values
```
[40] ✓ 0.3s

```
array([0, 0, 0, ..., 1, 1, 1], dtype=int64)
```

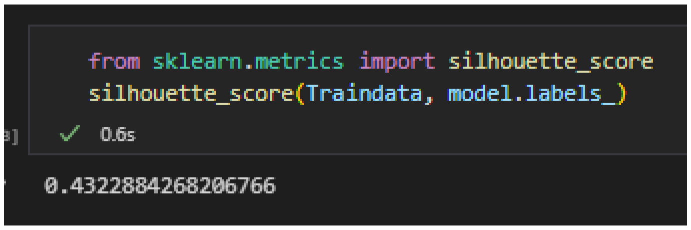
```
pd.crosstab(df['Class'], model.labels_)
```
[41] ✓ 0.3s

col_0	0	1
Class		
0	570	192
1	340	270

After we using the code pd.crosstab(df['Class'], model.labels_) we know that for the Class 0 there is the 570 is correct prediction but another 192 is incorrect prediction and for the Class 1 there is there is the 340 is correct prediction but the rest 270 is incorrect prediction

D. Model Evaluation

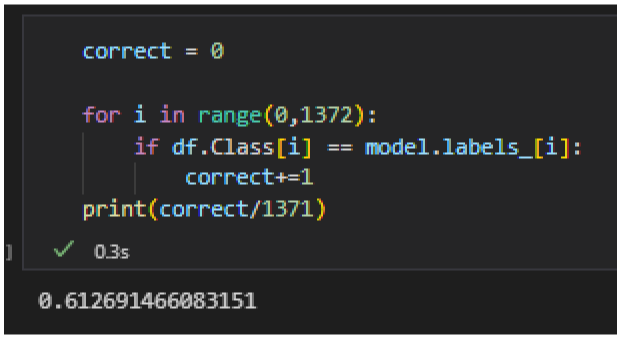
I use the silhouette score and the correctness value to assessing the model



```
from sklearn.metrics import silhouette_score
silhouette_score(Traindata, model.labels_)
```
✓ 0.6s

```
0.4322884268206766
```

- The silhouette score of 0.43 is acceptable



```
correct = 0

for i in range(0,1372):
    if df.Class[i] == model.labels_[i]:
        correct+=1
print(correct/1371)
```

```
0.612691466083151
```

- The accuracy score is 0.61 in accuracy. Which means the accuracy is 61%

E. Conclusion

After testing with different number of clusters I found out that 2 clusters give the best results which is 0.61 for accuracy sore So if you can analyze the number of clusters so it will give you the best accuracy score