(1) Select columns: Goal, students_reached, and funding_status and create a new data-frame. (1 point)

```
1  import numpy as np
2  import pandas as pd
3  import matplotlib.pyplot as plt
```

```
1  df = pd.read_excel("Crowdfunding_data_1000_projects (5).xlsx")
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-7-1e545c3baae7> in <module>()
----> 1 df = pd.read_excel("Crowdfunding_data_1000_projects (5).xlsx")

                                  6 frames
/usr/local/lib/python3.7/dist-packages/xlrd/__init__.py in open_workbook(filename,
logfile, verbosity, use_mmap, file_contents, encoding_override, formatting_info,
on_demand, ragged_rows)
    114         peek = file_contents[:peeksz]
    115     else:
--> 116         with open(filename, "rb") as f:
    117             peek = f.read(peeksz)
    118     if peek == b"PK\x03\x04": # a ZIP file

FileNotFoundError: [Errno 2] No such file or directory:
'Crowdfunding_data_1000_projects (5).xlsx'
```

```
1 df.head()
```

	Project_ID	school_latitude	school_longitude	school_city	school_state	school_zip
0	1	45.310140	-93.807736	Monticello	MN	55362
1	2	29.795216	-95.358101	Houston	TX	77009
2	3	37.754852	-122.426160	San Francisco	CA	94114
3	4	36.297083	-119.789619	Lemoore	CA	93245
4	5	33.946010	-118.223360	South Gate	CA	90280

```
df1 = df.loc[:,["Goal","students_reached","funding_status"]]
```

```
df1.head()
```

	Goal	students_reached	funding_status
0	887.15	12	completed
1	761.52	63	NotCompleted
2	266.55	88	completed
3	808.15	30	NotCompleted
4	1296.65	92	NotCompleted

```
a = {"completed":1,"NotCompleted":0}
df1.funding_status=df1.funding_status.map(a)
df1.head()
```

	Goal	students_reached	funding_status
0	887.15	12	1
1	761.52	63	0
2	266.55	88	1
3	808.15	30	0
4	1296.65	92	0

(2) Create random train and test data-frames in 75:25 ratio. (1 point)

```
x = df1.iloc[:,[0,1]]
y = df1.iloc[:,[-1]]
```

```
from sklearn.model_selection import train_test_split
```

```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.25)
```

(3) Using K-means, cluster the train data-frame into two clusters. Use Goal and students_reached columns (only

independent variables) for clustering (4 points)

```
1 from sklearn.cluster import KMeans
```

```
1 km=KMeans(n_clusters=2)
```

```
1 km.fit(xtrain)
```

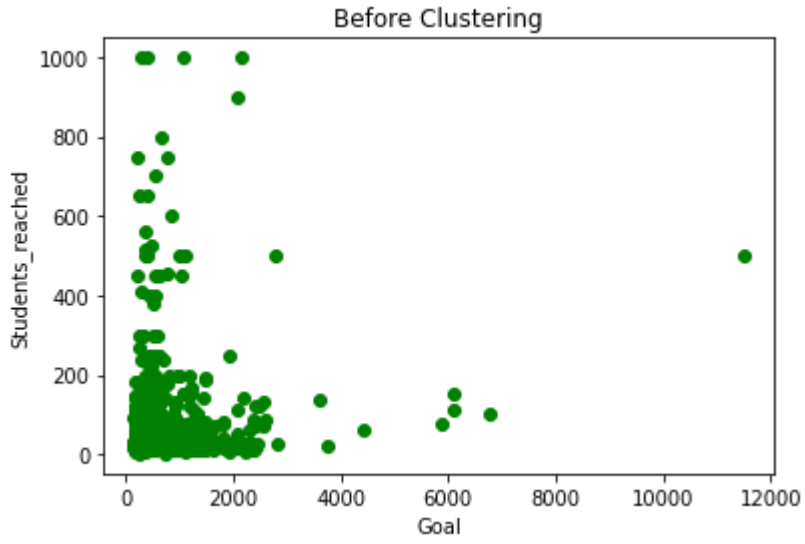
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

(4) Plot the scatter plots before and after clustering. (2 points)

```
1 ykm = km.predict(xtrain)
```

```
1 plt.scatter(xtrain.iloc[:,0],xtrain.iloc[:,1],c="g")
2 plt.title("Before Clustering")
3 plt.xlabel("Goal")
4 plt.ylabel("Students_reached")
```

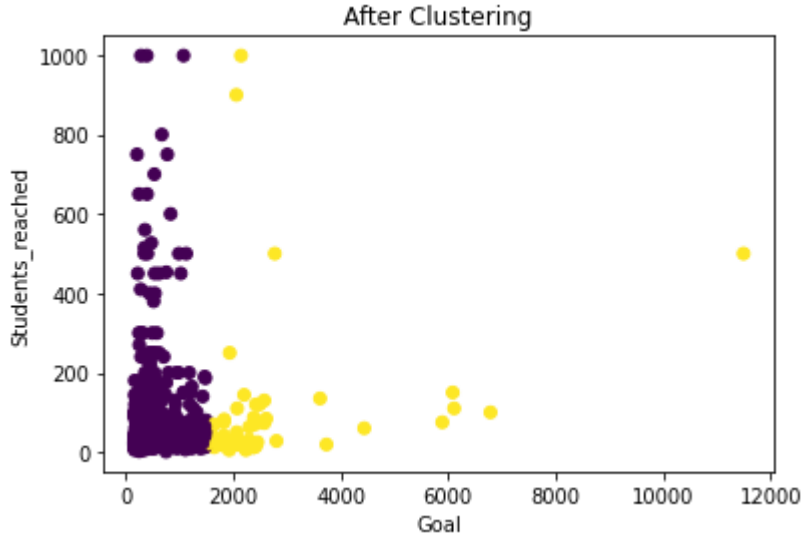
```
Text(0, 0.5, 'Students_reached')
```




```
1 plt.scatter(xtrain.iloc[:,0],xtrain.iloc[:,1],c=ykm)
2 plt.title("After Clustering")
```

```
3 plt.xlabel("Goal")
4 plt.ylabel("Students_reached")
```

```
Text(0, 0.5, 'Students_reached')
```



(5) Use predict() function and predict cluster labels for test data-frame. (2 points)

```
1 y_pred = km.predict(xtest)
```

```
1 y_pred#cluster labels for test dataframe
```

```
array([1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 1, 0, 1, 1, 0, 1], dtype=int32)
```

```
1
```

Colab paid products - Cancel contracts here