1-Import Libararies
2-Import data
3-Take care of na and missing data
4-Exept the dependent variable, convert all categorical variables to dummies
5-Define X and scale it by using StandardScaler(it has a templete codes)
6-Model the data (it has a templete codes)
7-Evaluate a model using Silhouette_Score (it has a templete codes) (close to 1 is better clustering, close to 0 no clustering)
8-Fit the optimal K on X-Scaled
9- Create a new column for cluster's labels
10- Groupby the clusters and show them in a graph
Example is used from Build a K-means model as part of Google Advanced Data Analytics Professional Certificate

In [1]:
```
# Import standard operational packages.
import numpy as np
import pandas as pd

# Important tools for modeling and evaluation.
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import StandardScaler

# Import visualization packages.
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
penguins = pd.read_csv(r"C:\Users\lasra\Downloads\penguins.csv")
```

In [4]:
```
penguins.head(n=10)
```

Out[4]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female
5	Adelie	Torgersen	39.3	20.6	190.0	3650.0	male
6	Adelie	Torgersen	38.9	17.8	181.0	3625.0	female
7	Adelie	Torgersen	39.2	19.6	195.0	4675.0	male
8	Adelie	Torgersen	34.1	18.1	193.0	3475.0	NaN
9	Adelie	Torgersen	42.0	20.2	190.0	4250.0	NaN

In [5]:
```
penguins["species"].uniqueque()
```
Out[5]:
```
array(['Adelie', 'Chinstrap', 'Gentoo'], dtype=object)
```

In [7]:
```
penguins["species"].value_counts(dropna=False)
```
Out[7]:
```
Adelie       152
Gentoo       124
Chinstrap     68
Name: species, dtype: int64
```

There are three types of species. Note the Chinstrap species is less common than the other species.
This has a chance to affect K-means clustering as K-means performs best with similar sized groupings.

In [9]:
```
penguins.isna().sum()
```
Out[9]:
```
species               0
island                0
bill_length_mm        2
bill_depth_mm         2
flipper_length_mm     2
body_mass_g           2
sex                  11
dtype: int64
```

In [10]:
```
penguins_subset= penguins.dropna(axis=0).reset_index(drop=True)
```

In [11]:
```
penguins_subset.isna().sum()
```
Out[11]:
```
species              0
island               0
bill_length_mm       0
bill_depth_mm        0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

In [12]:
```
penguins_subset.head()
```
Out[12]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female
3	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female
4	Adelie	Torgersen	39.3	20.6	190.0	3650.0	male

In [13]:
```
penguins_subset['sex']=penguins_subset["sex"].str.upper()
```

In [14]:
```
penguins_subset=pd.get_dummies(penguins_subset,drop_first=True, columns=["sex"])
```

In [16]:
```
penguins_subset=penguins_subset.drop(['island'], axis=1)
```

In [18]:
```
X=penguins_subset.drop(["species"], axis=1)
```

In [19]:
```
X_scaled=StandardScaler().fit_transform(X)
```

In [21]:
```
# Fit K-means and evaluate inertia for different values of k.

num_clusters = [i for i in range(2, 11)]

def kmeans_inertia(num_clusters, x_vals):
    """
    Accepts as arguments list of ints and data array.
    Fits a KMeans model where k = each value in the list of ints.
    Returns each k-value's inertia appended to a list.
    """
    inertia = []
    for num in num_clusters:
        kms = KMeans(n_clusters=num, random_state=42)
        kms.fit(x_vals)
        inertia.append(kms.inertia_)

    return inertia
```

In [22]:
```
inertia = kmeans_inertia(num_clusters, X_scaled)
inertia
```

```
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
```

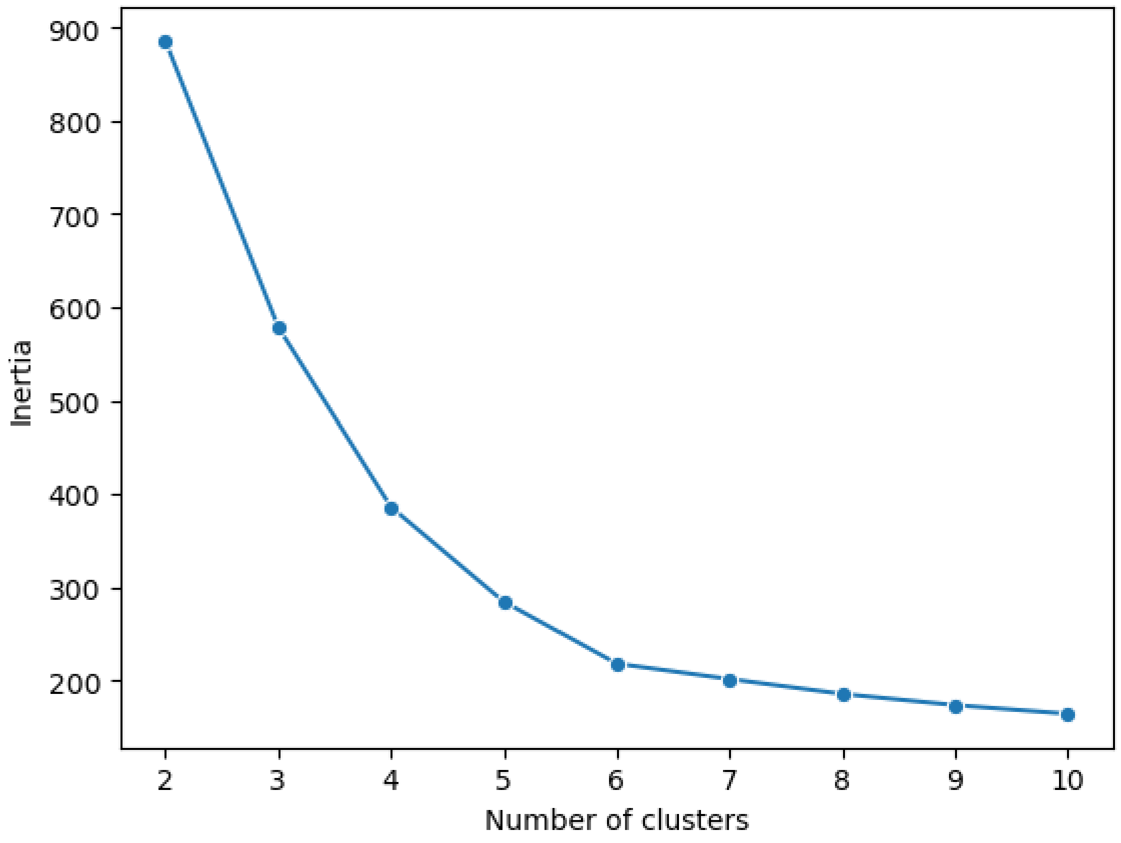
```
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
```

Out[22]:

```
[885.6224143652249,
 578.8284278107236,
 386.14534424773296,
 284.5464837898288,
 217.92858573807678,
 201.39287843423267,
 185.461310432323,
 173.45452114979847,
 164.1200152026071]
```

In [23]:

```python
# Create a line plot.

plot = sns.lineplot(x=num_clusters, y=inertia, marker = 'o')
plot.set_xlabel("Number of clusters");
plot.set_ylabel("Inertia");
```



In [24]:
```
# Evaluate silhouette score.
# Write a function to return a list of each k-value's score.

def kmeans_sil(num_clusters, x_vals):
    """
    Accepts as arguments list of ints and data array.
    Fits a KMeans model where k = each value in the list of ints.
    Calculates a silhouette score for each k value.
    Returns each k-value's silhouette score appended to a list.
    """
    sil_score = []
    for num in num_clusters:
        kms = KMeans(n_clusters=num, random_state=42)
        kms.fit(x_vals)
        sil_score.append(silhouette_score(x_vals, kms.labels_))

    return sil_score


sil_score = kmeans_sil(num_clusters, X_scaled)
sil_score
```

```
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
```

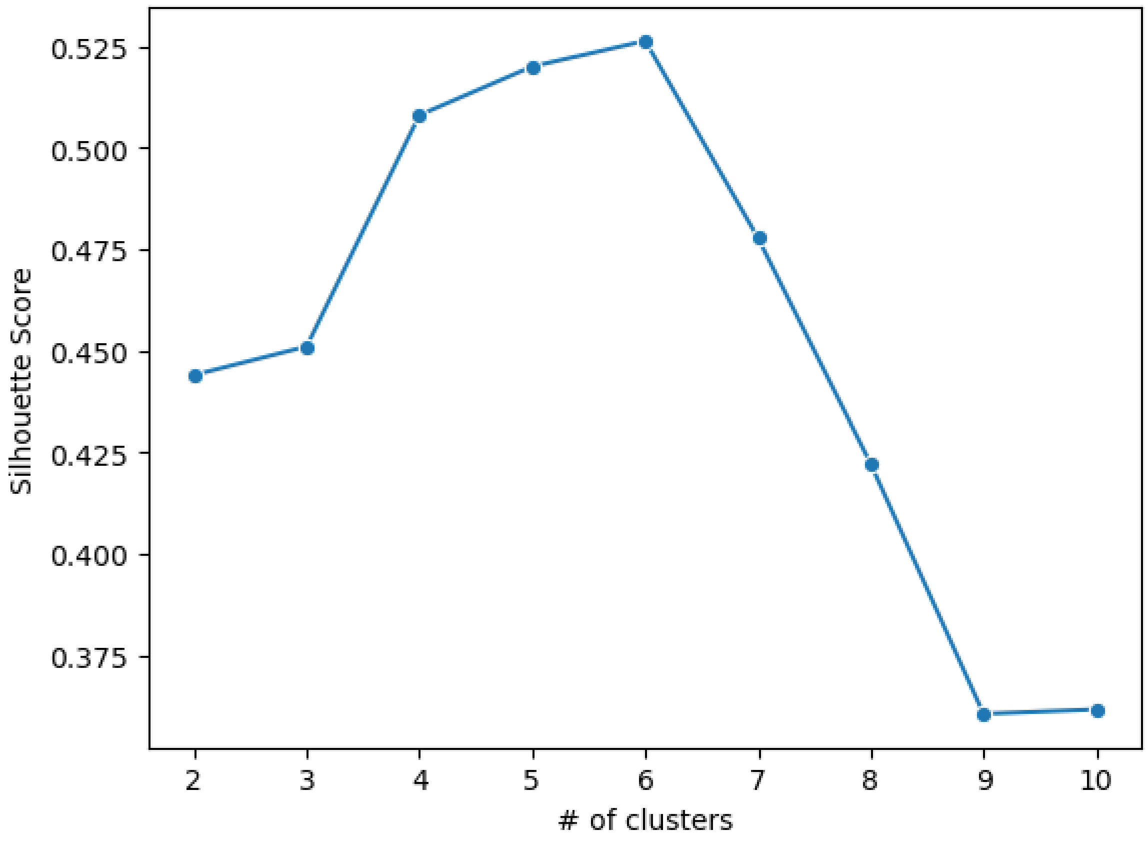
```
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
```

Out[24]:

```
[0.44398088353055243,
 0.45101024097188364,
 0.5080140996630784,
 0.519998574860868,
 0.5263224884981607,
 0.47774022332151733,
 0.4221920732643224,
 0.36062890821417276,
 0.3617250563420018]
```

In [25]:

```
# Create a line plot.

plot = sns.lineplot(x=num_clusters, y=sil_score, marker = 'o')
plot.set_xlabel("# of clusters");
plot.set_ylabel("Silhouette Score");
```



In [26]:
```
# Fit a 6-cluster model.

kmeans6 = KMeans(n_clusters=6, random_state=42)
kmeans6.fit(X_scaled)
```

```
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\lasra\anaconda3-2\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWar
ning: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=2.
  warnings.warn(
```

Out[26]:
```
▾ KMeans
KMeans(n_clusters=6, random_state=42)
```

In [27]:
```
# Print unique labels.

print('Unique labels:', np.unique(kmeans6.labels_))
```

```
Unique labels: [0 1 2 3 4 5]
```

In [28]:
```
# Create a new column `cluster`.

penguins_subset['cluster'] = kmeans6.labels_
penguins_subset.head()
```

Out[28]:

	species	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex_MALE	cluster
0	Adelie	39.1	18.7	181.0	3750.0	1	2
1	Adelie	39.5	17.4	186.0	3800.0	0	1
2	Adelie	40.3	18.0	195.0	3250.0	0	1
3	Adelie	36.7	19.3	193.0	3450.0	0	1
4	Adelie	39.3	20.6	190.0	3650.0	1	2

In [29]:

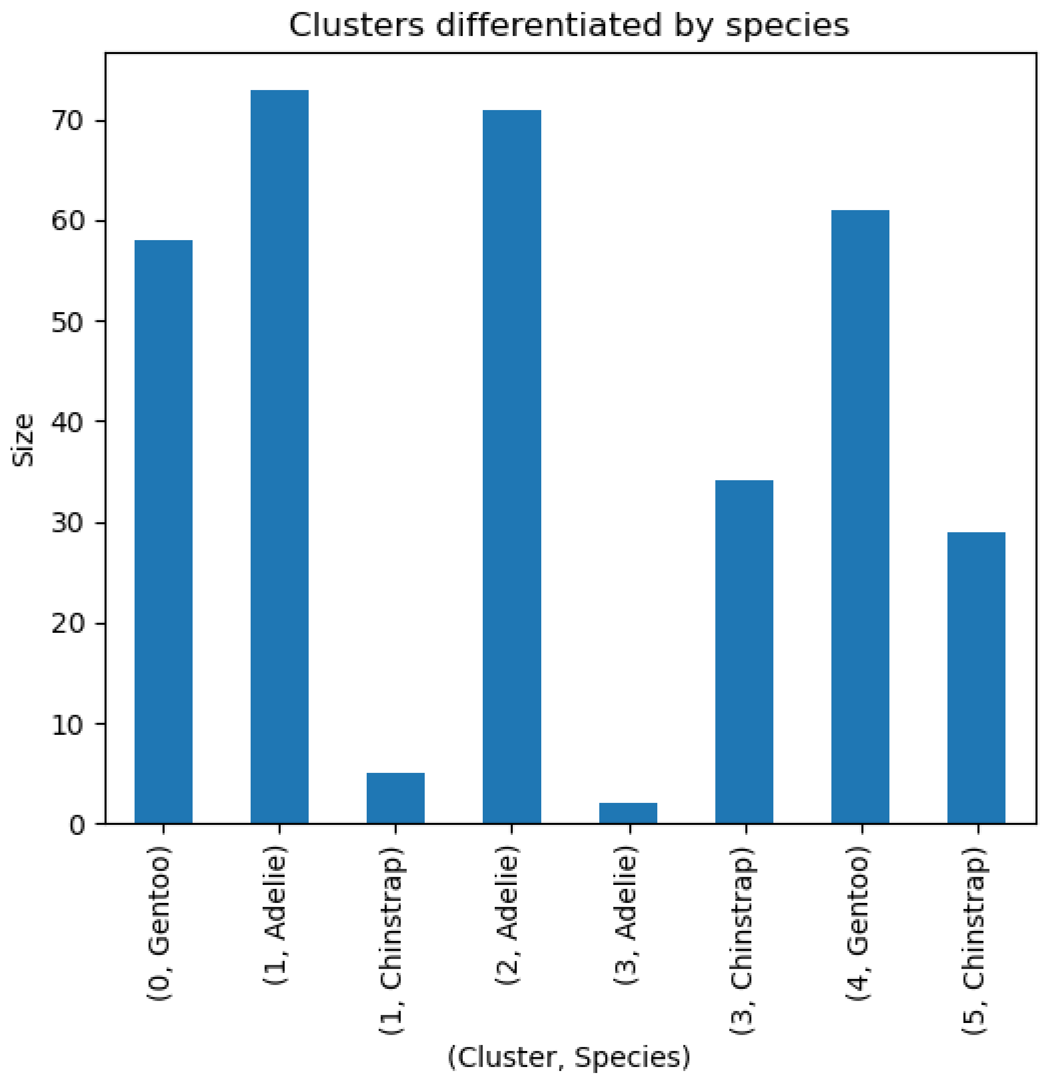
```
# Verify if any `cluster` can be differentiated by `species`.

penguins_subset.groupby(by=['cluster','species']).size()
```

Out[29]:

```
cluster  species  
0        Gentoo       58
1        Adelie       73
         Chinstrap     5
2        Adelie       71
3        Adelie        2
         Chinstrap    34
4        Gentoo       61
5        Chinstrap    29
dtype: int64
```

In [30]:

```
penguins_subset.groupby(by=['cluster', 'species']).size().plot.bar(title='Clusters dif
                                                                    figsize=(6, 5),
                                                                    ylabel='Size',
                                                                    xlabel='(Cluster, S
```



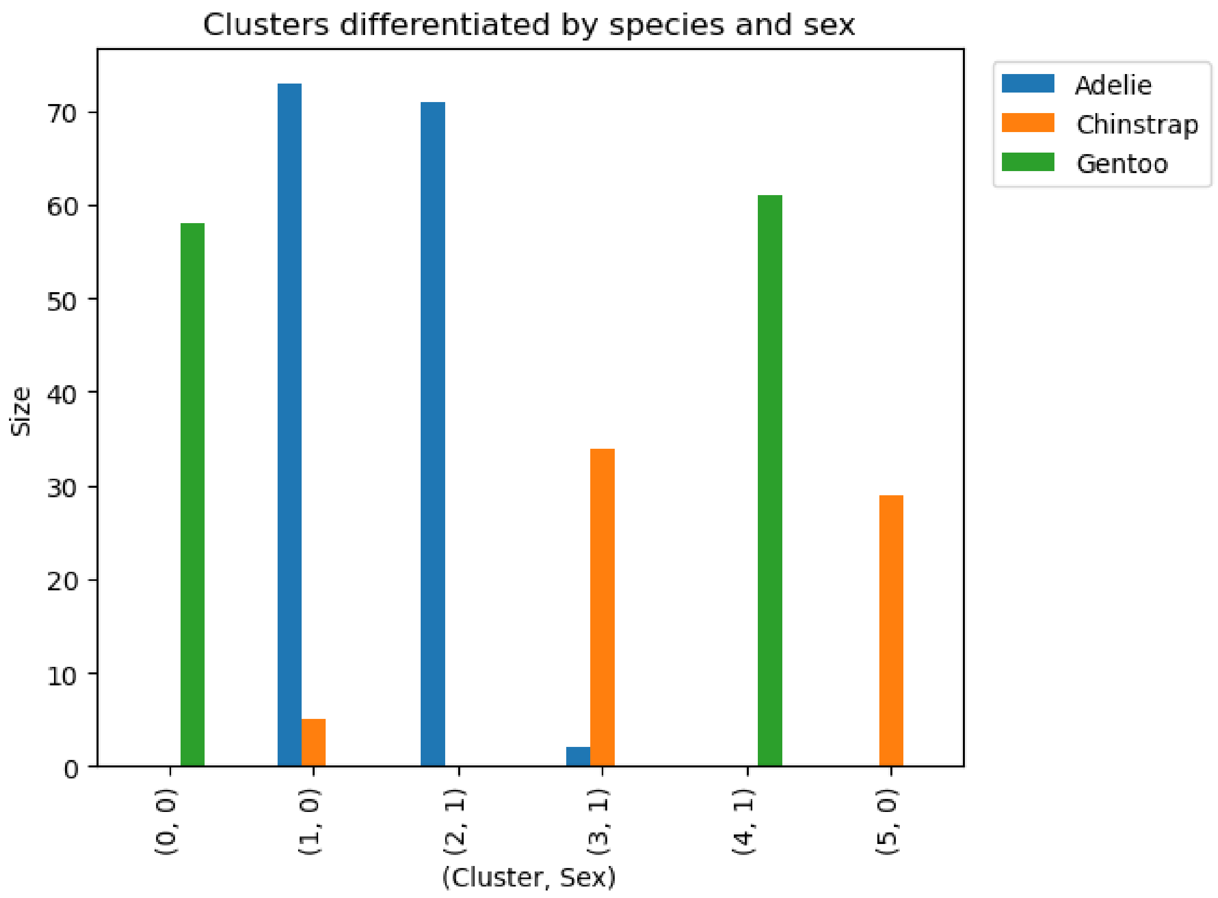
In [31]:
```
# Verify if each `cluster` can be differentiated by `species` AND `sex_MALE`.

penguins_subset.groupby(by=['cluster','species', 'sex_MALE']).size().sort_values(ascen
```

Out[31]:
```
cluster  species    sex_MALE
1        Adelie     0           73
2        Adelie     1           71
4        Gentoo     1           61
0        Gentoo     0           58
3        Chinstrap  1           34
5        Chinstrap  0           29
1        Chinstrap  0            5
3        Adelie     1            2
dtype: int64
```

Even though clusters 1 and 3 weren't all one species or sex, the groupby indicates that the algorithm produced clusters mostly differentiated by species and sex.

In [34]:
```
penguins_subset.groupby(by=['cluster','species','sex_MALE']).size().\
unstack(level = 'species', fill_value=0).plot.bar(title='Clusters differentiated by sp



plt.legend(bbox_to_anchor=(1.3, 1.0))
```

```
Out[34]: <matplotlib.legend.Legend at 0x1d0328aa8c0>
```


Clusters differentiated by species and sex
Size
70
60
50
40
30
20
10
0
(0, 0)
(1, 0)
(2, 1)
(3, 1)
(4, 1)
(5, 0)
(Cluster, Sex)
Adelie
Chinstrap
Gentoo

```
In [ ]:
```