COMP4220: Machine Learning, Spring 2021, Assignment 6 (lec 6,8)

Due: Wednsday, April 28, 11pm

Please read lecture 6:Unsupervised Learning and Clustering and lecture 8:Neural Networks and Keras!

Please submit one pdf file for all questions.

You can type your answer for the first two questions in the below cell of each question using "Markdown" option!

**When turning in assignments after the due date, please clearly specify the number of late hours used.

P1. Describe two techniques to select the right number of clusters when using K-Means.

Two techniques used to select the right number of clusters when using K-Means are the Elbow method and the Average Silhouette method. The Elbow method involves first using K-means clustering as the clustering algorithm. Then for each k, the total within-cluster sum of square is calculated. The curve of this is plotted according to the number of clusters (k). The locating of the elbow like bend is considered the ideal number of clusters in this instance. In the Average Silhouette method k-means clustering is used as the clustering algorithm first. Then for each k the average silhouette of observations is calculated. The curve of this is plotted according to the number of clusters (k). In this instance, the maximum is considered the ideal number of clusters.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

P2. What is the difference between hard clustering and soft clustering?

In hard clustering, each data point is binary in a sense that it either completely belongs to a cluster or it does not. On the contrary, in soft clustering, a probability/likelihood of that data point is assingned for the cluster. in soft clustering, the item can exist in multiple clusters unlike in hard clustering.

P3. List five hyperparameters you can tweak in a basic neural network?

One hyperparameter you can tweak in a basic neural network is the number of layers. Increasing the number of layers can improve accuracy. Another hyperparameter that can be tweaked is the number of neurons per layer. Increasing this can also increase accuracy while decreasing it can cause underfitting. A third hyperparameter that can be tweaked is the activation functions. A fourth hyperparameter that can be tweaked is the weight initialization. Another example of a hyperparameter that can be tweaked is the learning rate which determines how quickly a network updates its parameters.

p4: What is backpropagation and how does it work?

Backpropagration is the practice of fine tuning the weights a neural net based on the error rate that was obtained from the previous iteration. The goal of backpropagation is to lower error rates and in turn make the model more reliable. At the very basic level, this process starts with designing the model, then foward propagation takes place, then finally with the error obtained backpropagating takes place. The error is fed back into the model, the optimization function helps tweak things for the next iteration. and then this repeats untill satisfied.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

P5. What are some of the main applications of clustering algorithms? (Name at least three applications)

Clustering algorithms can be used as a spam filter for mail. The algorithms looks at the different pieces that make up the email and classifies it as either spam or not. Clustering algorithms can also be used for marketing. Individuals with similar traits are grouped together marketed to based on their traits. They can even be used in diagnosing individuals, based on the present symptoms, clustering alborithms can draw conclusions as to what illness an individual might have.

P6. Cluster the following points, (18, 10), (21, 11), (22, 22), (24, 15), (26, 12), (26, 13), (27, 14), (30, 33), (31, 39), (35, 37), (39, 44), (40, 27), (41, 29), (42, 20), (44, 28), (46, 21), (47, 30), (48, 31), (49, 23), (54, 24) use the numbers as pairs of x and y values which represent their locations. All distances are measured with Euclidean distance.

In [1]:
```
# Importing the libraries
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import pandas as pd
```

P6.1 Train the Kmeans clustering algorithm on the data with (24, 15), (30, 33), and (54, 24) as the

initial cluster centers. Predict the cluster of each data point and show the result.

In [14]:
```
X = np.array([(18, 10), (21, 11), (22, 22), (24, 15), (26, 12), (26, 13
), (27, 14), (30, 33), (31, 39), (35, 37), (39, 44), (40, 27), (41, 29
), (42, 20), (44, 28), (46, 21), (47, 30), (48, 31), (49, 23), (54, 24
)])
Centroids = np.array([(24, 15), (30, 33), (54, 24)])
kmeans = KMeans(n_clusters = 3, init = Centroids).fit(X)
kmeans_pred = kmeans.fit_predict(X)
```

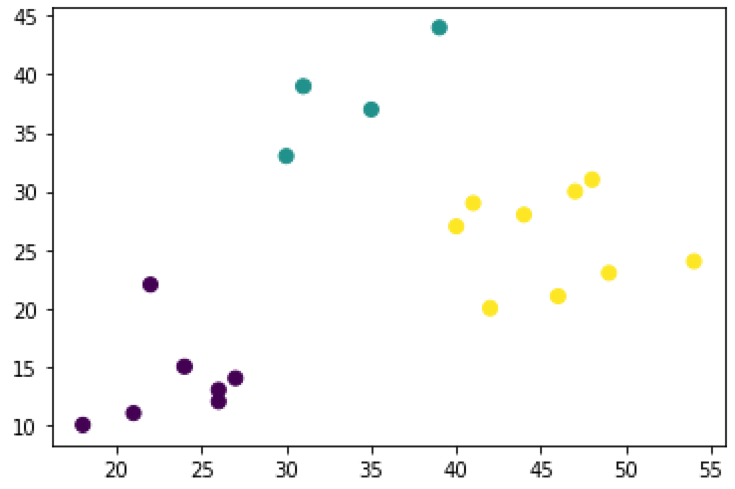
```
<ipython-input-14-719757e0061f>:3: RuntimeWarning: Explicit initial cen
ter position passed: performing only one init in k-means instead of n_i
nit=10
  kmeans = KMeans(n_clusters = 3, init = Centroids).fit(X)
C:\Users\jacob\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1
105: RuntimeWarning: Explicit initial center position passed: performin
g only one init in k-means instead of n_init=10
  return self.fit(X, sample_weight=sample_weight).labels_
```

P6.2 Visualize the clusters

In [16]:
```
plt.scatter(X[:, 0], X[:, 1], c=kmeans_pred, s=50, cmap='viridis')
```

Out[16]:
```
<matplotlib.collections.PathCollection at 0x16d3630e6d0>
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



P7. Programming assignment (KMeans clustering)

In [17]:
```
#Import the dataset

dataset = pd.read_csv('Mall_Customers.csv')
X = dataset.iloc[:, [3, 4]].values
```

P7. 1 Use the elbow method to find the optimal number of clusters through a loop (1 to 10 clusters) and visualize the result on a 2D plot

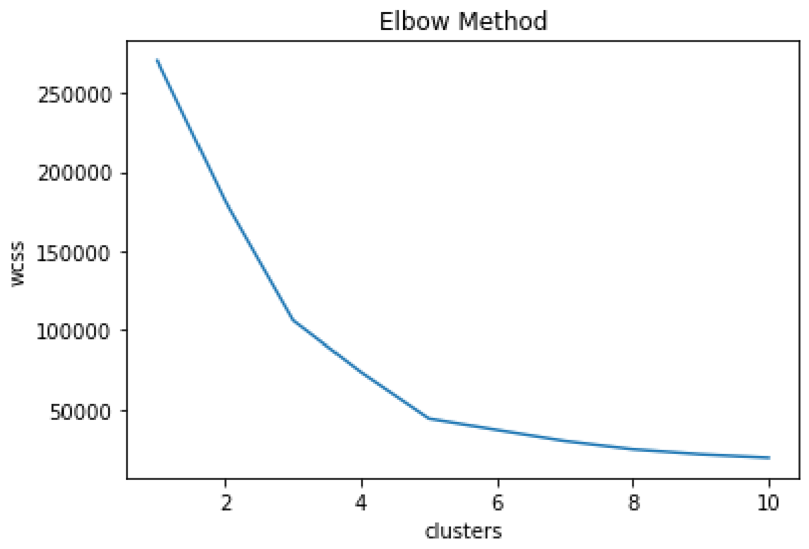
Set init = 'k-means++'

In [18]:
```
kmeans = dict(init = 'k-means++', max_iter = 300, n_init = 10, random_s
tate = 0)
wcss = [KMeans(n_clusters = i, **kmeans).fit(X).inertia_ for i in range
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
(1,11)]

plt.plot(range(1,11), wcss)
plt.title('Elbow Method')
plt.xlabel('clusters')
plt.ylabel('wcss')
plt.show()
```



P7. 2 Fit K-Means to the dataset and use the optimal number of clusters that you found in the previous part and show the result

In [19]:
```
kmeans = KMeans(n_clusters = 5, **kmeans)
k_means = kmeans.fit_predict(X)
```

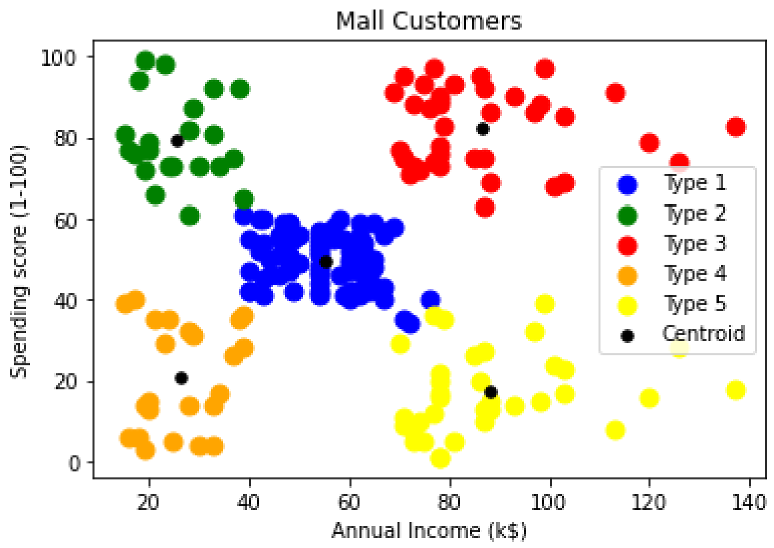
P7. 3 Visuale the clusters

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

(xlabel('Annual Income (k$)' and ylabel('Spending Score (1-100)'))

Use different colors for clusters and label them

In [24]:
```
plt.scatter(X[k_means == 0, 0], X[k_means == 0, 1], s = 80, c = 'blue',
label = 'Type 1')
plt.scatter(X[k_means == 1, 0], X[k_means == 1, 1], s = 80, c = 'green'
, label = 'Type 2')
plt.scatter(X[k_means == 2, 0], X[k_means == 2, 1], s = 80, c = 'red',
label = 'Type 3')
plt.scatter(X[k_means == 3, 0], X[k_means == 3, 1], s = 80, c = 'orang
e', label = 'Type 4')
plt.scatter(X[k_means == 4, 0], X[k_means == 4, 1], s = 80, c = 'yello
w', label = 'Type 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1
], s = 30, c = 'black', label = 'Centroid')
plt.title('Mall Customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending score (1-100)')
plt.legend()
plt.show()
```



Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

P8. Programming Assignment (Artificial Neural Network-ANN)

The problem of classifying customers staying in the bank or leaving!

Part 1 - Data Preprocessing

In [17]:
```
# Importing the libraries
import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.compose import ColumnTransformer
import keras
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix
tf.__version__
```

Out[17]:
```
'2.4.1'
```

In [18]:
```
# Importing the dataset
dataset = pd.read_csv('Churn_Modelling.csv')
X = dataset.iloc[:, 3:13].values # Git rid of useless columns # start
 "CreditScore" column
y = dataset.iloc[:, 13].values   # The last column "Exited" is our depe
ndent variable
print(X)
print(y)
```

```
[[619 'France' 'Female' ... 1 1 101348.88]
 [608 'Spain' 'Female' ... 0 1 112542.58]
 [502 'France' 'Female' ... 1 0 113931.57]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
 ...
 [709 'France' 'Female' ... 0 1 42085.58]

 [772 'Germany' 'Male' ... 1 0 92888.52]
 [792 'France' 'Female' ... 1 0 38190.78]]
[1 0 1 ... 1 1 0]
```

In [19]:

```
# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_1 = LabelEncoder()
X[:, 1] = labelencoder_X_1.fit_transform(X[:, 1])
labelencoder_X_2 = LabelEncoder()
X[:, 2] = labelencoder_X_2.fit_transform(X[:, 2])
ct = ColumnTransformer([("Geogrophy", OneHotEncoder(), [1])], remainder
= 'passthrough')
X = ct.fit_transform(X)
X = X[:, 1:]

print(X)
print(y)
```

```
[[0.0 0.0 619 ... 1 1 101348.88]
 [0.0 1.0 608 ... 0 1 112542.58]
 [0.0 0.0 502 ... 1 0 113931.57]
 ...
 [0.0 0.0 709 ... 0 1 42085.58]
 [1.0 0.0 772 ... 1 0 92888.52]
 [0.0 0.0 792 ... 1 0 38190.78]]
[1 0 1 ... 1 1 0]
```

P8.1.1 Split the dataset into the Training set and Test set (test_size = 0.2)

In [20]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2, random_state = 0)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

P8.1.2 Apply Feature Scaling to all features before training a neural network

In [21]:
```
from sklearn.preprocessing import StandardScaler
standard_scaler = StandardScaler()
X_train = standard_scaler.fit_transform(X_train)
X_test = standard_scaler.transform(X_test)
```

Part 2 - Now let's make the ANN!

In [22]:
```
# Importing the Keras libraries and packages
import keras
from keras.models import Sequential
from keras.layers import Dense
```

In [23]:
```
# Initialising the ANN
classifier = Sequential()
```

P8.2.1 Add the input layer and the first hidden layer

In [24]:
```
# hint: (units = 6, kernel_initializer = 'uniform', activation = 'rel
u', input_dim = 11)
classifier.add(Dense(units = 6, kernel_initializer = 'uniform', activat
ion = 'relu', input_dim = 11))
```

P8.2.2 Add the second hidden layer

In [25]:
```
#  hint: (units = 6, kernel_initializer = 'uniform', activation = 'rel
u')
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
classifier.add(Dense(units = 6, kernel_initializer = 'uniform', activat
ion = 'relu'))
```

P8.2.3 Add the output layer

In [26]:
```
# (units = 1, kernel_initializer = 'uniform', activation = 'sigmoid')
classifier.add(Dense(units = 1, kernel_initializer = 'uniform', activat
ion = 'sigmoid'))
```

P8.2.4 Compile the ANN

In [27]:
```
# hint: (optimizer = 'adam', loss = 'binary_crossentropy', metrics =
 ['accuracy']))
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', me
trics = ['accuracy'])
```

P8.2.3 Fit the ANN to the Training set

In [29]:
```
classifier.fit(X_train, y_train)
```

```
250/250 [==============================] - 1s 616us/step - loss: 0.6490
- accuracy: 0.7766
```

Out[29]:
```
<tensorflow.python.keras.callbacks.History at 0x1cc5ebf9d60>
```

Part 3 - Making the predictions and evaluating the model

P8.3.1 Predict the Test set results

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [30]:
```
# hint: just consider y_pred > 0.5  (y_pred = (y_pred > 0.5))
yPred = classifier.predict(X_test)
yPred = (yPred > 0.5)
```

P8.3.2 Make the Confusion Matrix and show the result

In [33]:
```
from sklearn.metrics import confusion_matrix
confusionMatrix = confusion_matrix(y_test, yPred)
confusionMatrix
```

Out[33]:
```
array([[1595,    0],
       [ 405,    0]], dtype=int64)
```

In [ ]:
```
```

In [ ]:
```
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD