

# HW

May 11, 2018

## 1 Short-Answer:

## 2 Question 1

```
In [ ]: #k-Nearest Neighbors: Fit
        Import KNeighborsClassifier from sklearn.neighbors

        #Create arrays for the features and the response variable
        y = df['party'].values
        X = df.drop('party', axis=1).values

        #Create a k-NN classifier with 6 neighbors

        #_____

        #Fit the classifier to the data
        #_____

        #Answer
        # Import KNeighborsClassifier from sklearn.neighbors
        from sklearn.neighbors import KNeighborsClassifier

        # Create arrays for the features and the response variable
        y = df['party'].values
        X = df.drop('party', axis=1).values

        # Create a k-NN classifier with 6 neighbors
        knn = KNeighborsClassifier(n_neighbors=6)

        # Fit the classifier to the data
        knn.fit(X,y)
```

## 3 Question 2

```
In [ ]: # Import necessary modules
        from sklearn import datasets
```

```

import matplotlib.pyplot as plt

# Load the digits dataset: digits
#-----

# Print the keys and DESCR of the dataset
#-----
#-----

# Print the shape of the images and data keys
#-----
#-----

#Answer:
# Import necessary modules
from sklearn import datasets
import matplotlib.pyplot as plt

# Load the digits dataset: digits
digits = datasets.load_digits()

# Print the keys and DESCR of the dataset
print(digits.keys())
print(digits.DESCR)

# Print the shape of the images and data keys
print(digits.images.shape)
print(digits.data.shape)

```

## 4 Question 3

```

In [ ]: # Import necessary modules
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split

# Create feature and target arrays
X = digits.data
y = digits.target

# Split into training and test set
#-----

# Create a k-NN classifier with 7 neighbors: knn
#-----

```

```

# Fit the classifier to the training data
#-----

# Print the accuracy
#-----

#Answer:
# Import necessary modules
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split

# Create feature and target arrays
X = digits.data
y = digits.target

# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state=42)

# Create a k-NN classifier with 7 neighbors: knn
knn = KNeighborsClassifier(n_neighbors=7)

# Fit the classifier to the training data
knn.fit(X_train,y_train)

# Print the accuracy
print(knn.score(X_test, y_test))

```

## 5 Question 4

```

In [ ]: # Import necessary modules
import keras
from keras.layers import Dense
from keras.models import Sequential

# Save the number of columns in predictors: n_cols
n_cols = predictors.shape[1]

# Set up the model: model
#-----

# Add the first layer
#-----

# Add the second layer

```

```

#-----

# Add the output layer
#-----

#Answer:
# Import necessary modules
import keras
from keras.layers import Dense
from keras.models import Sequential

# Save the number of columns in predictors: n_cols
n_cols = predictors.shape[1]

# Set up the model: model
model = Sequential()

# Add the first layer
model.add(Dense(50, activation='relu', input_shape=(n_cols,)))

# Add the second layer
model.add(Dense(32, activation='relu'))

# Add the output layer
model.add(Dense(1))

```

## 6 Question 5

```

In [ ]: from sklearn.metrics import classification_report
        from sklearn.metrics import confusion_matrix
        from sklearn.neighbors import KNeighborsClassifier
        from sklearn.model_selection import train_test_split

# Create training and test set
#-----

# Instantiate a k-NN classifier: knn
#-----

# Fit the classifier to the training data
#-----

# Predict the labels of the test data: y_pred
#-----

```

```

# Generate the confusion matrix and classification report
#-----
#-----

#Answer:
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

# Create training and test set
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.4, random_state=42)

# Instantiate a k-NN classifier: knn
knn = KNeighborsClassifier(n_neighbors=6)

# Fit the classifier to the training data
knn.fit(X_train,y_train)

# Predict the labels of the test data: y_pred
y_pred = knn.predict(X_test)

# Generate the confusion matrix and classification report
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

```

## 7 Dataset Mini Question

Load the Wine Data Set in Sklearn. Split the data into test/train split using seed=40. Classify using KNN. Print the score.

```

In [ ]: #Answer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

# Load the wine data base
wine_data = load_wine()
# Get the X and Target Labels out
wine_x = wine_data['data']
wine_target = wine_data['target']

# Split into test,train set
X_train, X_test, y_train, y_test = train_test_split( wine_x, wine_target, test_size=0.3,

```

```
# Create a k-NN classifier with 6 neighbors: knn
knn = KNeighborsClassifier(n_neighbors=6)
# Fit the training data
knn.fit(X_train, y_train)
# Predict the X
y_predict=knn.predict(X_test)
# Print report
print(classification_report(y_predict, y_test))
```