DATA: 23/04/2022 EXP No: 08	Title of the Lab Implementation of Machine Learning Algorithms for an Application	Name:Avinash reddy Vasipalli Registration Number: RA1911027010007 Section: N1 Lab Batch: 1 Day Order: 2

Aim: To understand phone purchase behavior of customers using KNN, which is a supervised Machine Learning algorithm.

ALGORITHM:

K-nearest neighbours (KNN) algorithm uses 'feature similarity' to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps −

Step 1 − For implementing any algorithm, we need dataset. So, during the first step of KNN, we must load the training as well as test data.

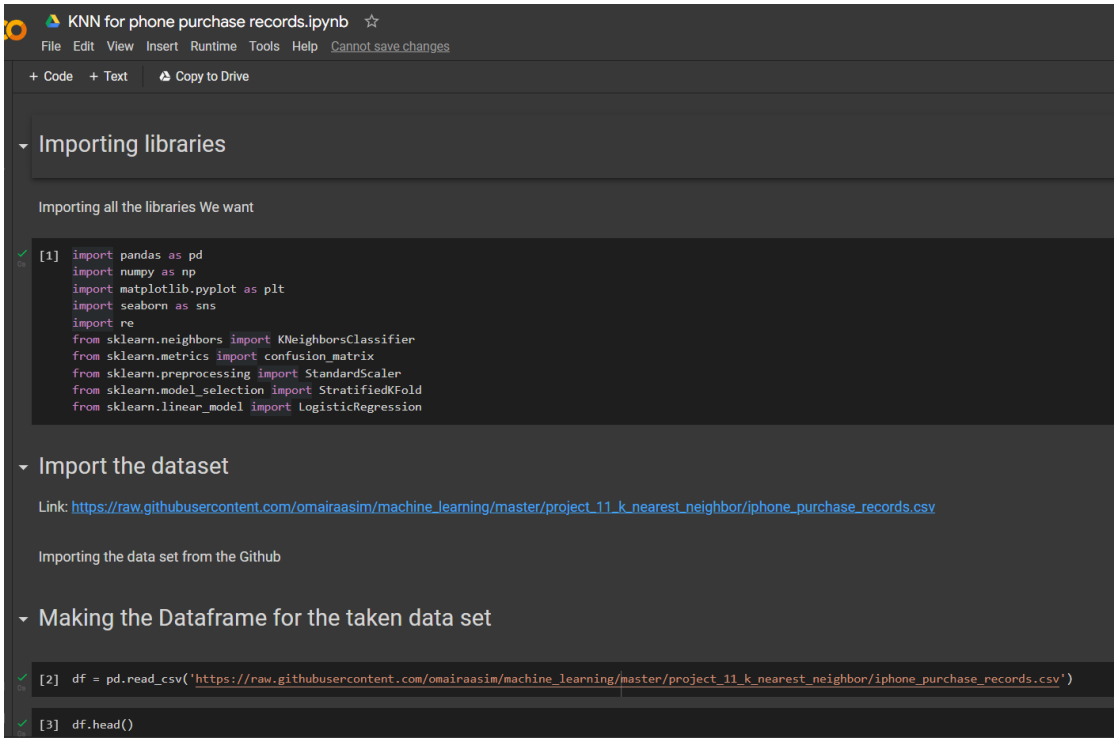
Step 2 − Next, we need to choose the value of K i.e., the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following −

- Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
- Now, based on the distance value, sort them in ascending order.
- Next, it will choose the top K rows from the sorted array.
- Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 – End

Source Code:



KNN for phone purchase records.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

```
[5] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Gender           400 non-null    object
 1   Age              400 non-null    int64
 2   Salary           400 non-null    int64
 3   Purchase Iphone  400 non-null    int64
dtypes: int64(3), object(1)
memory usage: 12.6+ KB
```

```
[6] df.describe()
```

	Age	Salary	Purchase Iphone
count	400.000000	400.000000	400.000000
mean	37.655000	69742.500000	0.357500
std	10.482877	34096.960282	0.479864
min	18.000000	15000.000000	0.000000
25%	29.750000	43000.000000	0.000000
50%	37.000000	70000.000000	0.000000
75%	46.000000	88000.000000	1.000000
max	60.000000	150000.000000	1.000000

```
[7] df.Gender.value_counts()
```

```
Female    204
Male      196
Name: Gender, dtype: int64
```

Df.Loc is used for the accessing the rows and columns.

```
[8] df.loc[df['Purchase Iphone']==1,"Gender"].value_counts()
```

KNN for phone purchase records.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

```
[11] from sklearn.preprocessing import LabelEncoder
```

```
[12] enc = LabelEncoder()
```

```
[13] X.Gender = enc.fit_transform(X.Gender)
```

```
[14] X
```

	Gender	Age	Salary
0	1	19	19000
1	1	35	20000
2	0	26	43000
3	0	27	57000
4	1	19	76000
...	...	...	...
395	0	46	41000
396	1	51	23000
397	0	50	20000
398	1	36	33000
399	0	49	36000

400 rows × 3 columns

```
[15] X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Gender  400 non-null    int64
 1   Age     400 non-null    int64
 2   Salary  400 non-null    int64
dtypes: int64(3)
memory usage: 9.5 KB
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import StratifiedKFold
from sklearn.linear_model import LogisticRegression


df = pd.read_csv('https://raw.githubusercontent.com/omairaasim/machine_learning/mas
ter/project_11_k_nearest_neighbor/iphone_purchase_records.csv')
df.head()
df.shape
df.info ()
df.describe()
df.Gender.value_counts()
df.loc[df['Purchase Iphone']==1,"Gender"].value_counts()
```

//Splitting of data

```
df.head(2)
X = df.iloc[:,:-1]
y = df.iloc[:,-1]
```

//Label Encoding

```
from sklearn.preprocessing import LabelEncoder
enc = LabelEncoder()
X.Gender = enc.fit_transform(X.Gender)
X.info ()
```

//Splitting the data into sets

```
skf = StratifiedKFold(n_splits=5)
for train_index,test_index in skf.split(X,y):
  X_train, X_test = X.iloc[train_index], X.iloc[test_index]
  y_train, y_test = y.iloc[train_index], y.iloc[test_index]
```

//Feature Scaling

```
scale = StandardScaler()
X_train = scale.fit_transform(X_train)
X_test = scale.fit_transform(X_test)
```

//Model Selection

```
log = LogisticRegression()
knn = KNeighborsClassifier(n_neighbors=5)
```

//Training the model

```
log.fit(X_train, y_train)
knn.fit(X_train, y_train)
```

//Test the Model

```
y_log_pred = log.predict(X_test)
y_knn_pred = knn.predict(X_test)
```

```
newdf = pd.DataFrame({"Actual":y_test, "Predicted":y_knn_pred})
newdf.head()
confusion_matrix(y_test, y_knn_pred)
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_log_pred)
lis = [i for i in range (2,101) if i%2==0]
acc= []
dic = {}
for i in lis:
  knn = KNeighborsClassifier(n_neighbors=i)
  knn.fit(X_train, y_train)
  y_knn_pred = knn.predict(X_test)
  acc.append(accuracy_score(y_test,y_knn_pred))
  # dic[i] = accuracy_score(y_test,y_knn_pred)

print(max(acc))
```

Output:

```
[28] accuracy_score(y_test, y_log_pred)

     0.725

[29] lis = [i for i in range(2,101) if i%2==0]

[30] acc=[]
     dic = {}
     for i in lis:
       knn = KNeighborsClassifier(n_neighbors=i)
       knn.fit(X_train, y_train)
       y_knn_pred = knn.predict(X_test)
       acc.append(accuracy_score(y_test,y_knn_pred))
       # dic[i] = accuracy_score(y_test,y_knn_pred)

     print(max(acc))
     # 0.8875 = 89%
     # 0.875 = 88%

     0.875
```

GitHub link - https://github.com/Avi-2362/18CSC305J-AI-Lab/blob/main/Machine%20Learning%20Algorithms%20for%20an%20Application/Machine%20Learning%20Algorithms%20for%20an%20Application.ipynb