

# 1. Using KNN, (assuming K=2,3,4,5), classify point (50,40) and point (30,40), whether it belongs to class 1 or class 2.

In [7]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
test_data= pd.read_csv('test_new.csv')
train_data = pd.read_csv('train_new.csv')
```

In [9]:

```
df1=test_data
df2=train_data
```

In [10]: df1.head()

Out[10]:

	dvcat	weight	dead	airbag	seatbelt	frontal	sex	ageOFocc	yearacc	yearVeh	abcat
0	40-54	18.895	alive	airbag	belted	0	m	25	2002	2000.0	deploy
1	25-39	266.532	alive	none	none	1	f	28	2001	1991.0	unavail
2	25-39	51.810	alive	airbag	belted	1	f	36	1999	1994.0	deploy
3	10-24	1567.626	alive	airbag	belted	0	f	24	2002	1994.0	nodeploy
4	25-39	31.342	alive	none	none	1	f	46	1997	1990.0	unavail

In [11]: df2.head()

Out[11]:

	dvcat	weight	dead	airbag	seatbelt	frontal	sex	ageOFocc	yearacc	yearVeh	abcat
0	55+	53.342	dead	airbag	belted	1	f	48	2002	1997	deploy
1	25-39	154.960	alive	none	none	1	m	26	2001	1968	unavail
2	55+	38.994	alive	none	none	1	f	51	2002	1994	unavail
3	25-39	168.568	alive	airbag	belted	1	m	27	1998	1996	deploy
4	10-24	27.751	alive	airbag	belted	0	m	26	2002	1997	nodeploy

```
In [12]: from sklearn.preprocessing import LabelEncoder
t= LabelEncoder()
df1["airbag"]=t.fit_transform(df1["airbag"])
df1["seatbelt"]=t.fit_transform(df1["seatbelt"])
df1["sex"]=t.fit_transform(df1["sex"])
from sklearn.preprocessing import LabelEncoder
u= LabelEncoder()
df2["airbag"]=u.fit_transform(df2["airbag"])
df2["seatbelt"]=u.fit_transform(df2["seatbelt"])
df2["sex"]=u.fit_transform(df2["sex"])
features_train = list(['airbag','seatbelt','sex'])
train_x = train_data[features_train].values
train_x
```

```
Out[12]: array([[0, 0, 0],
                [1, 1, 1],
                [1, 1, 0],
                ...,
                [0, 0, 0],
                [1, 1, 0],
                [1, 0, 1]])
```

```
In [13]: target_train = list(['dead'])
train_y = train_data[target_train].values
features_test = list(['airbag','seatbelt','sex'])
test_x = test_data[features_test].values
target_test = list(['dead'])
test_y= test_data[target_test].values
df1['dead']=df1['dead'].map({'alive':0,'dead':1})
df2['dead']=df1['dead'].map({'alive':0,'dead':1})
df2.head()
```

```
Out[13]:
```

	dvcat	weight	dead	airbag	seatbelt	frontal	sex	ageOFocc	yearacc	yearVeh	abcat
0	55+	53.342	NaN	0	0	1	0	48	2002	1997	deploy
1	25-39	154.960	NaN	1	1	1	1	26	2001	1968	unavail
2	55+	38.994	NaN	1	1	1	0	51	2002	1994	unavail
3	25-39	168.568	NaN	0	0	1	1	27	1998	1996	deploy
4	10-24	27.751	NaN	0	0	0	1	26	2002	1997	nodeploy

```
In [14]: scaler = StandardScaler()
```

In [15]:

```
# fit training set
scaler.fit(train_x)
StandardScaler()
# apply transform for both test and training sets
train_x = scaler.transform(train_x)
test_x = scaler.transform(test_x)
# logistic model
logistic=LogisticRegression()
# fit values for x and y
logistic.fit(train_x,train_y)
```

C:\Users\91830\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().  
y = column\_or\_1d(y, warn=True)

Out[15]: LogisticRegression()

In [16]:

```
# prediction from test data
prediction=logistic.predict(test_x)
# confusion matrix
confusion_matrix=confusion_matrix(prediction,test_y)
print(confusion_matrix)
```

```
[[8249  403]
 [   0    0]]
```

In [17]:

```
# calculating accuracy
accuracy_score=accuracy_score(prediction,test_y)
print(accuracy_score)
0.9534211742949606
# print misclassified values from prediction
print('Misclassified sample : %d'%(test_y !=prediction).sum())
```

```
0.9534211742949606
Misclassified sample : 3486756
```

## K=2

```
In [19]: from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=2)
from sklearn.metrics import accuracy_score,confusion_matrix
confusion_matrix = confusion_matrix(test_y, prediction)
print("\t","Predicted values")
print("Original values","\n",confusion_matrix)
```

```

          Predicted values
Original values
[[8249    0]
 [ 403    0]]
```

```
In [20]: print('Misclassified samples: %d' % (test_x != prediction))
# effect of x value on classifier
Misclassified_sample=[]
```

Misclassified samples: 1

C:\Users\91830\AppData\Local\Temp\ipykernel\_12380\1481535045.py:1: DeprecationWarning: elementwise comparison failed; this will raise an error in the future.

```
print('Misclassified samples: %d' % (test_x != prediction))
```

```
In [21]: # claculating error for k values
for i in range(1,20):
    knn=KNeighborsClassifier(n_neighbors=i)
    knn.fit(train_x,train_y)
    pred=knn.predict(test_x)
    Misclassified_sample.append(test_y !=pred)
print(Misclassified_sample)
```

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

## K=3

```
In [22]: from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=3)
from sklearn.metrics import accuracy_score,confusion_matrix
confusion_matrix = confusion_matrix(test_y, prediction)
print("\t","Predicted values")
print("Original values","\n",confusion_matrix)
```

```

          Predicted values
Original values
[[8249   0]
 [ 403   0]]
```

```
In [23]: print('Misclassified samples: %d' % (test_x != prediction))
# effect of x value on classifier
Misclassified_sample=[]
```

Misclassified samples: 1

C:\Users\91830\AppData\Local\Temp\ipykernel\_12380\4027423056.py:1: DeprecationWarning: elementwise comparison failed; this will raise an error in the future.

print('Misclassified samples: %d' % (test\_x != prediction))

```
In [24]: # claculating error for k values
for i in range(1,20):
    knn=KNeighborsClassifier(n_neighbors=i)
    knn.fit(train_x,train_y)
    pred=knn.predict(test_x)
    Misclassified_sample.append(test_y !=pred)
print(Misclassified_sample)
```

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

## K=4

```
In [25]: from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=4)
from sklearn.metrics import accuracy_score,confusion_matrix
confusion_matrix = confusion_matrix(test_y, prediction)
print("\t","Predicted values")
print("Original values","\n",confusion_matrix)
```

```

          Predicted values
Original values
[[8249   0]
 [ 403   0]]
```

```
In [26]: print('Misclassified samples: %d' % (test_x != prediction))
# effect of x value on classifier
Misclassified_sample=[]
```

Misclassified samples: 1

C:\Users\91830\AppData\Local\Temp\ipykernel\_12380\1481535045.py:1: DeprecationWarning: elementwise comparison failed; this will raise an error in the future.

print('Misclassified samples: %d' % (test\_x != prediction))

```
In [27]: # claculating error for k values
for i in range(1,20):
    knn=KNeighborsClassifier(n_neighbors=i)
    knn.fit(train_x,train_y)
    pred=knn.predict(test_x)
    Misclassified_sample.append(test_y !=pred)
print(Misclassified_sample)
```

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

In [ ]:

## K=5

```
In [28]: print('Misclassified samples: %d' % (test_x != prediction))
# effect of x value on classifier
Misclassified_sample=[]
```

Misclassified samples: 1

C:\Users\91830\AppData\Local\Temp\ipykernel\_12380\4027423056.py:1: DeprecationWarning: elementwise comparison failed; this will raise an error in the future.

print('Misclassified samples: %d' % (test\_x != prediction))

```
In [29]: # claculating error for k values
for i in range(1,20):
    knn=KNeighborsClassifier(n_neighbors=i)
    knn.fit(train_x,train_y)
    pred=knn.predict(test_x)
    Misclassified_sample.append(test_y !=pred)
print(Misclassified_sample)
```

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

return self.\_fit(X, y)

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

In [ ]:



## 2.Now, again use your 'crashTest.csv' file and build a classifier model using both Logistic regression and KNN.

```
In [52]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
df=pd.read_csv('train_new.csv')
df.head()
```

```
Out[52]:
```

	dvcat	weight	dead	airbag	seatbelt	frontal	sex	ageOFocc	yearacc	yearVeh	abcat
0	55+	53.342	dead	airbag	belted	1	f	48	2002	1997	deploy
1	25-39	154.960	alive	none	none	1	m	26	2001	1968	unavail
2	55+	38.994	alive	none	none	1	f	51	2002	1994	unavail
3	25-39	168.568	alive	airbag	belted	1	m	27	1998	1996	deploy
4	10-24	27.751	alive	airbag	belted	0	m	26	2002	1997	nodeploy

```
In [50]: df['dvcat'].unique()
```

```
Out[50]: array([4, 2, 1, 3, 0], dtype=int64)
```

In [53]:

```

df['dvcat']=df['dvcat'].map({'1-9km/h':0,'10-24':1,'25-39':2,'40-54':3,'55+':4})
df['dead']=df['dead'].map({'dead':0,'alive':1})
df['airbag']=df['airbag'].map({'airbag':1,'none':0})
df['seatbelt']=df['seatbelt'].map({'none':0,'belted':1})
df['sex']=df['sex'].map({'m':0,'f':1})
df['abcat']=df['abcat'].map({'unavail':0,'deploy':1,'nodeploy':2})
df['occRole']=df['occRole'].map({'pass':0,'driver':1})
df = df.drop(['caseid'],axis=1)
k = list(df.columns)
features = list(set(k)-set(['injSeverity']))
output = list(['injSeverity'])
x = df.loc[:,features]
y = df.loc[:,output]
X_train,X_test,y_train,y_test =train_test_split(x,y,test_size=0.3,random_state=42)
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
knn = KNeighborsClassifier(n_neighbors=5,metric='minkowski', p=2)
model = knn.fit(X_train,y_train)

```

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

```

return self._fit(X, y)

```

In [54]:

```

y_pred = knn.predict(X_test)
print(y_pred)

```

```

[1 2 3 ... 1 0 1]

```

```
In [56]: from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
# trying for best fit of k values
for i in range(1,100):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    pred_i = knn.predict(X_test)
    print(i)
    print(pred_i)
    print(accuracy_score(y_test,pred_i))
```

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

```
    return self._fit(X, y)
```

```
1
[1 2 2 ... 3 0 1]
0.37703984819734343
```

C:\Users\91830\anaconda3\lib\site-packages\sklearn\neighbors\\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

```
    return self._fit(X, y)
```

```
2
[1 2 0 ... 1 0 1]
0.34497153700189753
```

```
In [57]: from sklearn.linear_model import LogisticRegression
lgr = LogisticRegression()
lgr.fit(X_train,y_train)
```

C:\Users\91830\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

```
    y = column_or_1d(y, warn=True)
```

C:\Users\91830\anaconda3\lib\site-packages\sklearn\linear\_model\\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max\_iter) or scale the data as shown in:  
<https://scikit-learn.org/stable/modules/preprocessing.html> (<https://scikit-learn.org/stable/modules/preprocessing.html>)

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression) ([https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression))

```
    n_iter_i = _check_optimize_result(
```

```
Out[57]: LogisticRegression()
```

```
In [58]: pred = lgr.predict(X_test)
accuracy_score(y_test,pred)
```

Out[58]: 0.49146110056925996

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: