M. Rohith
3122 21 5001 085

K-Nearest Neighbor algorithm

Program code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from numpy import set_printoptions
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, roc_auc_score, roc_curve,
confusion_matrix,f1_score, precision_score, recall_score
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import RandomUnderSampler
```

```
df = pd.read_csv("online_shoppers_intention.csv")
df.head()
df.info()
df.isnull().sum()
```

```
label_encoder = preprocessing.LabelEncoder()
df['Month'] = label_encoder.fit_transform(df['Month']);
df['VisitorType'] = label_encoder.fit_transform(df['VisitorType'])
df['Weekend'] = label_encoder.fit_transform(df['Weekend'])
df['Revenue'] = label_encoder.fit_transform(df['Revenue'])
```

```
df.describe().transpose()
sns.histplot(data = df, x = 'ProductRelated', bins = 10)
sns.countplot(data = df, x = 'Month')
sns.countplot(data = df, x = 'Revenue')
df.corr()
sns.heatmap(data = df)
sns.boxplot(data = df, x = 'VisitorType', y = 'BounceRates')
```

```
X = df.iloc[:,:17]
X
y = df.iloc[:,-1:]
y
```

```
smote = SMOTE()
```

```
rus = RandomUnderSampler(random_state=42, sampling_strategy =
'majority')
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state
= 42, test_size = 0.2)
```

```
print((y.value_counts()))
X_resampled, y_resampled = smote.fit_resample(X_train, y_train)
print((y_resampled.value_counts()))
```

```
# data normalization with sklearn
from sklearn.preprocessing import MinMaxScaler

# fit scaler on training data
norm = MinMaxScaler().fit(X_train)

# transform training data
X_train_norm = norm.transform(X_train)

# transform testing dataabs
X_test_norm = norm.transform(X_test)

# fit scaler on training data
norm = MinMaxScaler().fit(X_train)
```

```
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=11)
model2 = KNeighborsClassifier(n_neighbors=11)
model.fit(X_train, y_train)
model2.fit(X_resampled, y_resampled)
y_pred = model.predict(X_test)
y_pred2 = model2.predict(X_test)
```

```
import matplotlib.pyplot as plt
y_pred_proba = model.predict_proba(X_test)[::,1]
auc = roc_auc_score(y_test, y_pred_proba)

fpr, tpr, _ = roc_curve(y_test,  y_pred_proba)

plt.plot(fpr,tpr,label="with oversampling, auc="+str(auc), )

x = [0, 1]
y = [0, 1]


y_pred_proba2 = model2.predict_proba(X_test)[::,1]
```

M. Rohith
3122 21 5001 085

```
auc2 = roc_auc_score(y_test, y_pred_proba2)

fpr, tpr, _ = roc_curve(y_test,  y_pred_proba2)

plt.plot(fpr,tpr,label="without oversampling, auc="+str(auc2),
color='red')

print("Accuracy score without oversampling:",accuracy_score(y_test,
y_pred))
print("F1 score without oversampling:",f1_score(y_test, y_pred))
print("Precision without oversampling:",precision_score(y_test,
y_pred))
print("Recall without oversampling:",recall_score(y_test, y_pred))

print()

print("Accuracy score with oversampling:",accuracy_score(y_test,
y_pred2))
print("F1 score with oversampling:",f1_score(y_test, y_pred2))
print("Precision with oversampling:",precision_score(y_test, y_pred2))
print("Recall with oversampling:",recall_score(y_test, y_pred2))


plt.plot(x,y)
plt.legend(loc=4)

plt.show()
```

Github Link:

https://colab.research.google.com/drive/1bhTGBhOyMHqAlrtfT0BLsy1CRpWnXiw-?usp=sharing