Support Vector Machine

1) Prepare a classification model using SVM for salary data

Data Description:

age -- age of a person

workclass -- A work class is a grouping of work

education -- Education of an individuals

maritalstatus -- Marital status of an individulas

occupation -- occupation of an individuals

relationship --

race -- Race of an Individual

sex -- Gender of an Individual

capitalgain -- profit received from the sale of an investment

capitalloss -- A decrease in the value of a capital asset

hoursperweek -- number of hours work per week

native -- Native of an individual

Salary -- salary of an individual

Ans:-

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score, classification_report


data = pd.read_csv(‘SalaryData_Test(1)’)


print(data.head())
```

```
le = LabelEncoder()
categorical_columns = ['workclass', 'education', 'maritalstatus', 'occupation',
'relationship', 'race', 'sex', 'native']
for col in categorical_columns:
    data[col] = le.fit_transform(data[col])


X = data.drop('Salary', axis=1)
y = data['Salary']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)


svm_classifier = SVC(kernel='linear')


svm_classifier.fit(X_train, y_train)


y_pred = svm_classifier.predict(X_test)


accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')


print('Classification Report:')
print(classification_report(y_test, y_pred))
```

2) classify the Size_Categorie using SVM

month	month of the year: 'jan' to 'dec'
day	day of the week: 'mon' to 'sun'
FFMC	FFMC index from the FWI system: 18.7 to 96.20
DMC	DMC index from the FWI system: 1.1 to 291.3
DC	DC index from the FWI system: 7.9 to 860.6
ISI	ISI index from the FWI system: 0.0 to 56.10
temp	temperature in Celsius degrees: 2.2 to 33.30
RH	relative humidity in %: 15.0 to 100
wind	wind speed in km/h: 0.40 to 9.40
rain	outside rain in mm/m2 : 0.0 to 6.4

Size_Categorie the burned area of the forest (Small , Large)

Ans:-

Data Preprocessing:

Load and inspect the dataset.

Encode categorical variables (month and day) into numerical format.

Split the dataset into features (X) and the target variable (y).

Train-Test Split:

Split the dataset into training and testing sets.

Feature Scaling:

Standardize the numerical features (e.g., FFMC, DMC, DC, ISI, temp, RH, wind, rain) to ensure that they have the same scale. This step is crucial for SVM.

Model Training:

Use the training set to train an SVM classifier.

Choose the appropriate kernel function (e.g., linear, radial basis function) based on the characteristics of your data.

Model Evaluation:

Evaluate the model's performance on the test set using metrics such as accuracy, precision, recall, or F1-score.

Hyperparameter Tuning (Optional):

Fine-tune hyperparameters, such as the regularization parameter (C) and kernel parameters, to improve the model's performance.

Prediction:

Use the trained SVM model to make predictions on new or unseen data.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score

df = pd.read_csv('forestfires (1)')

df['month'] = pd.Categorical(df['month']).codes
df['day'] = pd.Categorical(df['day']).codes

X = df.drop('Size_Categorie', axis=1)
y = df['Size_Categorie']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)


svm_classifier = SVC(kernel='linear', C=1.0)  # You can experiment with different parameters
svm_classifier.fit(X_train_scaled, y_train)


y_pred = svm_classifier.predict(X_test_scaled)


accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)


print(f'Accuracy: {accuracy}')
print('Classification Report:')
print(report)
```