Supervised Learning-Classification Models\SVM-For-Social-Network-ads.py

```
# ----- Support Vector Machine (SVM) ----- #
 2
   # Importing the libraries
 3
   import numpy as np
   import matplotlib.pyplot as plt
   import pandas as pd
   import pickle
 8
    # Importing the dataset
 9
   dataset = pd.read csv(r"C:\Users\Jan Saida\OneDrive\Documents\Desktop\Excel sheets\Social Network Ads.csv")
11 X = dataset.iloc[:, [2, 3]].values
   v = dataset.iloc[:, -1].values
12
13
   # Splitting the dataset into the Training set and Test set
   from sklearn.model selection import train test split
15
   X train, X test, y train, y test = train test split(X, y, test size = 0.20, random state = 0)
16
17
18 # Feature Scaling
19 from sklearn.preprocessing import StandardScaler
20 sc = StandardScaler()
21 X train = sc.fit transform(X train)
22 X test = sc.transform(X test)
23
24 # Training the SVM model on the Training set
25 from sklearn.svm import SVC
   classifier = SVC()
26
    classifier.fit(X train, y train)
27
28
29
30
   # Predicting the Test set results
   y pred = classifier.predict(X test)
32
33
34 | # Making the Confusion Matrix
```

```
35 from sklearn.metrics import confusion matrix
   cm = confusion matrix(y test, y pred)
36
   print(cm)
37
38
39 # This is to get the Models Accuracy
   from sklearn.metrics import accuracy_score
   ac = accuracy score(y test, y pred)
   print(ac)
42
43
   bias = classifier.score(X train,y train)
44
   print(bias)
45
46
   variance = classifier.score(X test,y test)
47
   print(variance)
48
49
   # This is to get the Classification Report
50
   from sklearn.metrics import classification report
51
   cr = classification report(y test, y pred)
52
53
54
55 # Pickling the trained model to a file
   with open('svm model.pkl', 'wb') as model file:
56
        pickle.dump(classifier, model file)
57
        print("Model saved to 'svm model.pkl'")
58
```

~\AppData\Roaming\Code\logs\20250125T225615\window1\exthost\output_logging_20250125T225618\10-Code.log

```
[Running] python -u "c:\VS Code\Supervised Learning-Classification Models\SVM-For-Social-Network-ads.py"
[[55 3]
        [1 21]]
4      0.95
5      0.903125
6      0.95
7      Model saved to 'svm_model.pkl'
8
9      [Done] exited with code=0 in 4.864 seconds
10
11
```

Streamlit\SVM-Model-Socaial-Network-ads-app.py

```
# frontend streamlit.py
2
3
    import streamlit as st
    import pickle
    import numpy as np
    from sklearn.preprocessing import StandardScaler
7
   # Load the pickled SVM model
    with open('svm model.pkl', 'rb') as model file:
9
        classifier = pickle.load(model file)
10
        scaler = StandardScaler()
11
12
13
   # Title of the Web App
    st.title("Support Vector Machine (SVM) Prediction Web App")
15
   # Description
16
   st.write("""
17
   This is a simple web app to predict the outcome of the Social Network Ads dataset using a trained Support Vector Machine (SVM) model.
18
    You can input features such as Age and Estimated Salary, and the app will predict whether the person will buy the product (1) or not (0).
19
20
21
   # User inputs for Age and Estimated Salary
22
    age = st.number input("Enter Age", min value=18, max value=100, step=1)
23
    salary = st.number input("Enter Estimated Salary", min value=10000, max value=150000, step=100)
24
25
    # Button to make a prediction
26
    if st.button("Make Prediction"):
27
       # Prepare the input data
28
       user input = np.array([[age, salary]])
29
30
        # Feature Scaling (same as in training)
31
        user input scaled = scaler.fit transform(user input)
32
33
        # Predict the result using the loaded model
34
```

```
prediction = classifier.predict(user_input_scaled)

# Display the prediction

if prediction == 1:
    st.write("Prediction: The person will buy the product (1).")

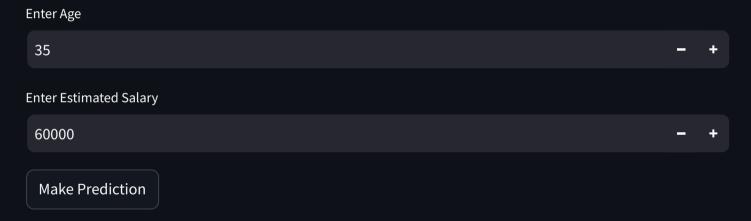
else:

st.write("Prediction: The person will not buy the product (0).")

22
43
```

Support Vector Machine (SVM) Prediction Web App

This is a simple web app to predict the outcome of the Social Network Ads dataset using a trained Support Vector Machine (SVM) model. You can input features such as Age and Estimated Salary, and the app will predict whether the person will buy the product (1) or not (0).



Prediction: The person will not buy the product (0).