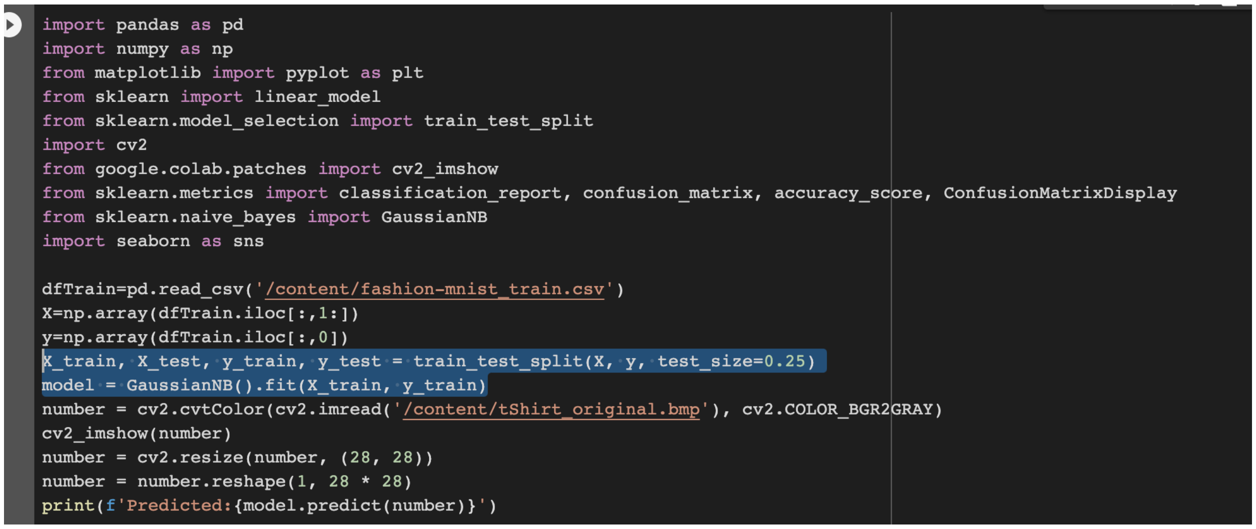
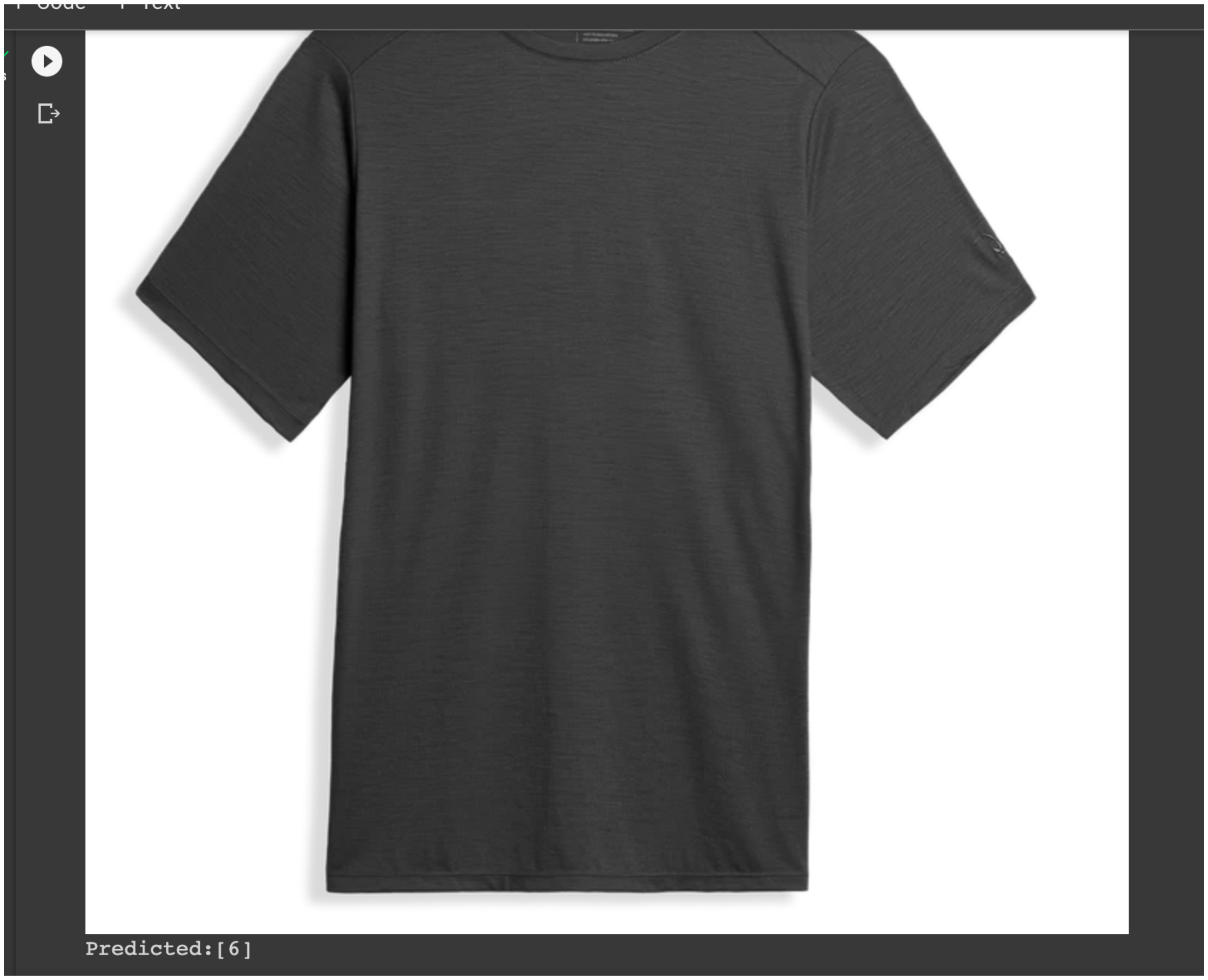
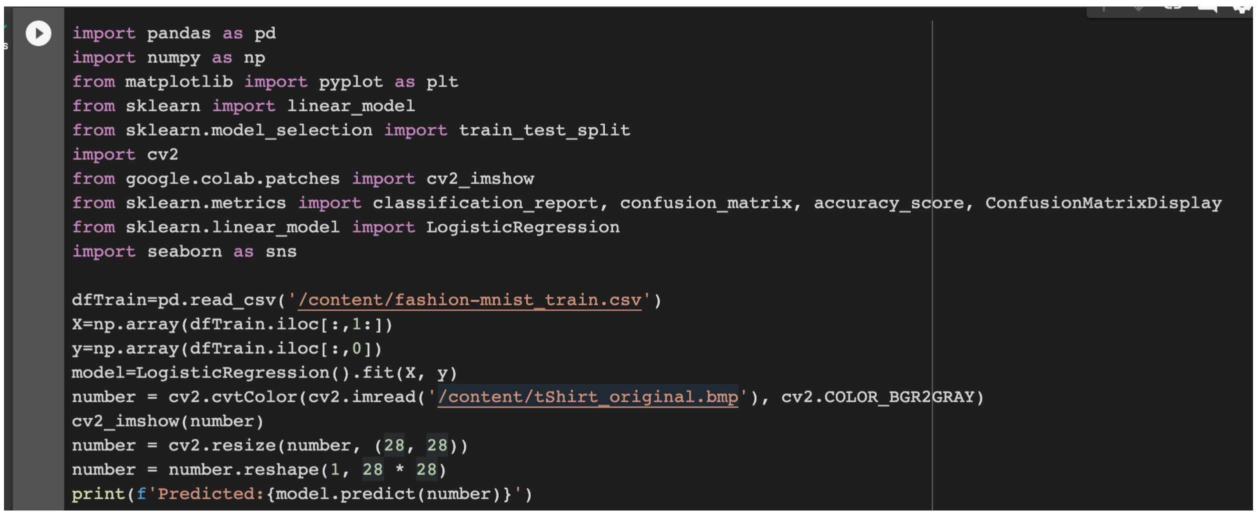


```python
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn import linear_model
from sklearn.model_selection import train_test_split
import cv2
from google.colab.patches import cv2_imshow
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, ConfusionMatrixDisplay
from sklearn.naive_bayes import GaussianNB
import seaborn as sns

dfTrain=pd.read_csv('/content/fashion-mnist_train.csv')
X=np.array(dfTrain.iloc[:,1:])
y=np.array(dfTrain.iloc[:,0])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
model = GaussianNB().fit(X_train, y_train)
number = cv2.cvtColor(cv2.imread('/content/tShirt_original.bmp'), cv2.COLOR_BGR2GRAY)
cv2_imshow(number)
number = cv2.resize(number, (28, 28))
number = number.reshape(1, 28 * 28)
print(f'Predicted:{model.predict(number)}')
```

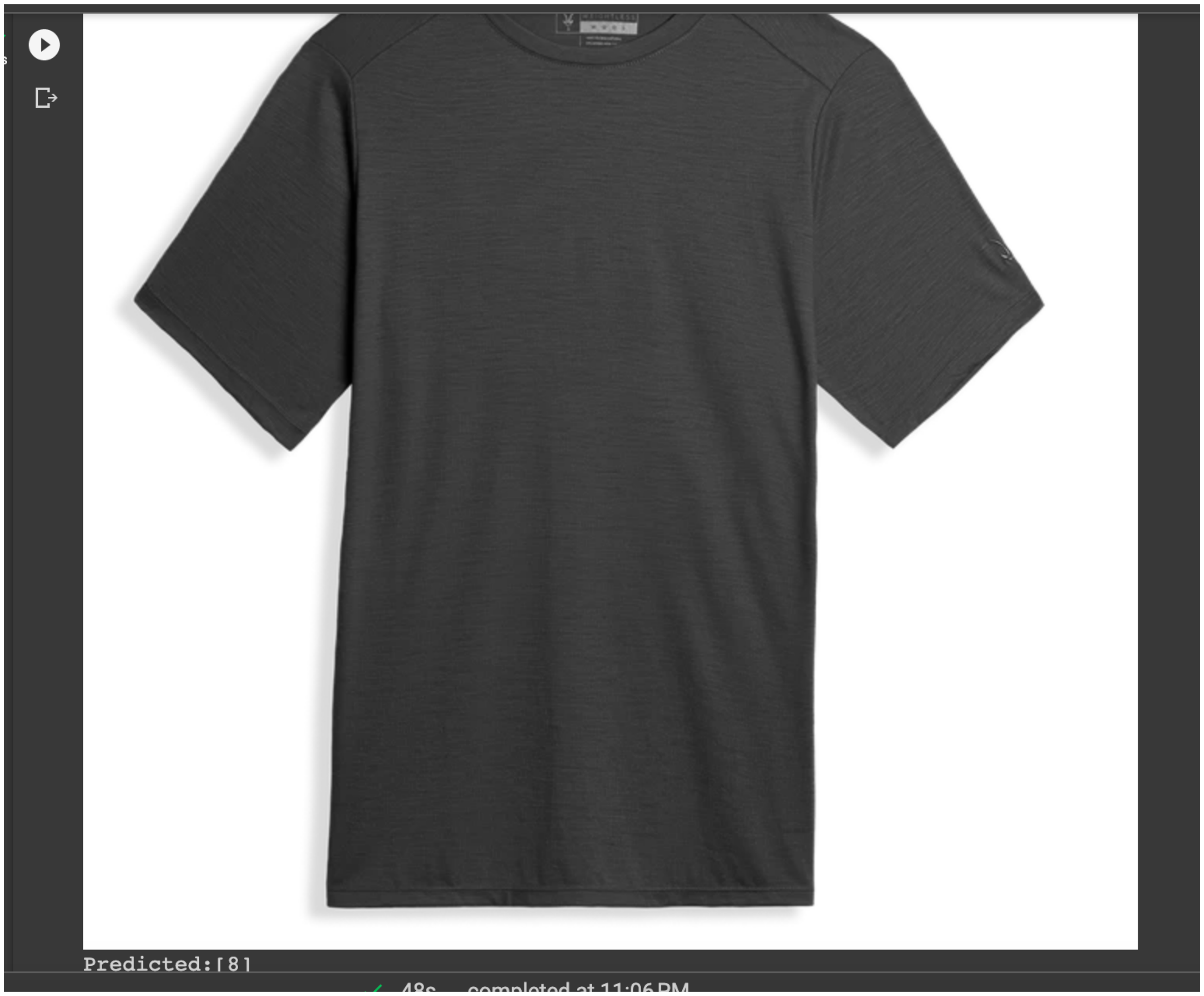


```
Predicted:[6]
```



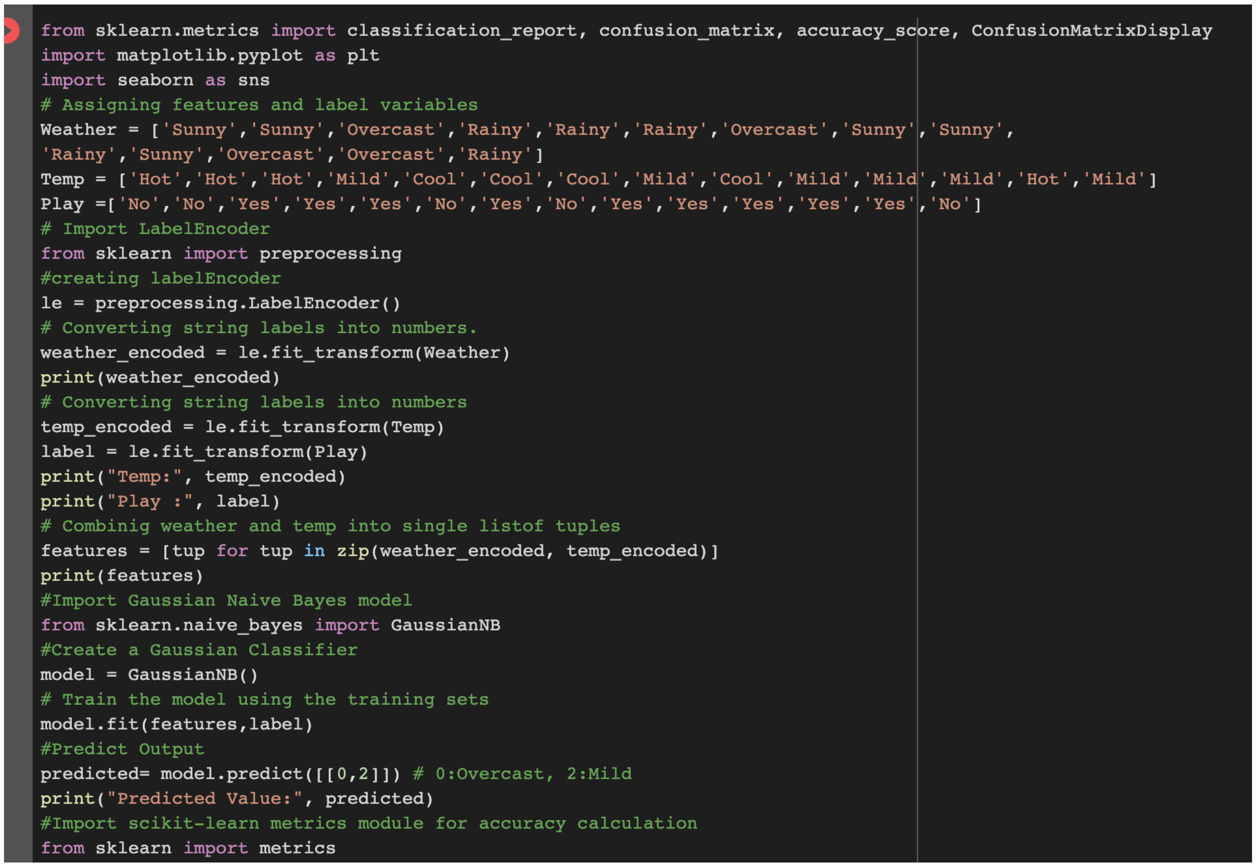
```python
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn import linear_model
from sklearn.model_selection import train_test_split
import cv2
from google.colab.patches import cv2_imshow
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, ConfusionMatrixDisplay
from sklearn.linear_model import LogisticRegression
import seaborn as sns

dfTrain=pd.read_csv('/content/fashion-mnist_train.csv')
X=np.array(dfTrain.iloc[:,1:])
y=np.array(dfTrain.iloc[:,0])
model=LogisticRegression().fit(X, y)
number = cv2.cvtColor(cv2.imread('/content/tShirt_original.bmp'), cv2.COLOR_BGR2GRAY)
cv2_imshow(number)
number = cv2.resize(number, (28, 28))
number = number.reshape(1, 28 * 28)
print(f'Predicted:{model.predict(number)}')
```

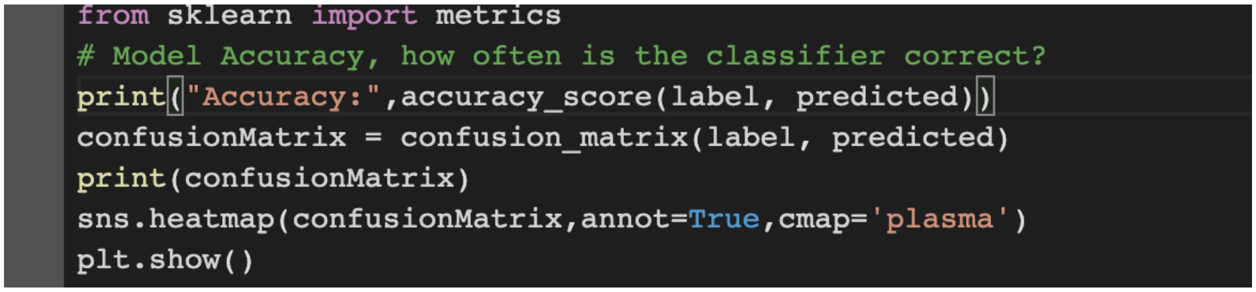


```
Predicted:[8]
```

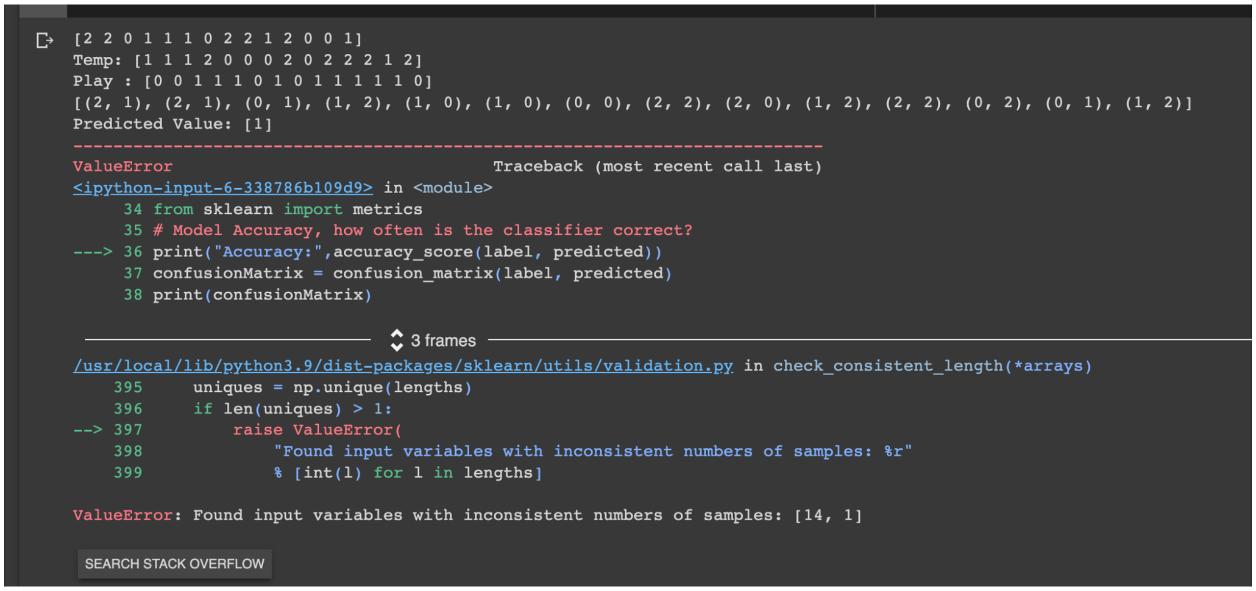
✓ 48s completed at 11:06 PM



```python
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, ConfusionMatrixDisplay
import matplotlib.pyplot as plt
import seaborn as sns
# Assigning features and label variables
Weather = ['Sunny','Sunny','Overcast','Rainy','Rainy','Rainy','Overcast','Sunny','Sunny',
'Rainy','Sunny','Overcast','Overcast','Rainy']
Temp = ['Hot','Hot','Hot','Mild','Cool','Cool','Cool','Mild','Cool','Mild','Mild','Mild','Hot','Mild']
Play =['No','No','Yes','Yes','Yes','No','Yes','No','Yes','Yes','Yes','Yes','Yes','No']
# Import LabelEncoder
from sklearn import preprocessing
#creating labelEncoder
le = preprocessing.LabelEncoder()
# Converting string labels into numbers.
weather_encoded = le.fit_transform(Weather)
print(weather_encoded)
# Converting string labels into numbers
temp_encoded = le.fit_transform(Temp)
label = le.fit_transform(Play)
print("Temp:", temp_encoded)
print("Play :", label)
# Combinig weather and temp into single listof tuples
features = [tup for tup in zip(weather_encoded, temp_encoded)]
print(features)
#Import Gaussian Naive Bayes model
from sklearn.naive_bayes import GaussianNB
#Create a Gaussian Classifier
model = GaussianNB()
# Train the model using the training sets
model.fit(features,label)
#Predict Output
predicted= model.predict([[0,2]]) # 0:Overcast, 2:Mild
print("Predicted Value:", predicted)
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
```



```python
from sklearn import metrics
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",accuracy_score(label, predicted))
confusionMatrix = confusion_matrix(label, predicted)
print(confusionMatrix)
sns.heatmap(confusionMatrix,annot=True,cmap='plasma')
plt.show()
```



```
[2 2 0 1 1 1 0 2 2 1 2 0 0 1]
Temp: [1 1 1 2 0 0 0 2 0 2 2 2 1 2]
Play : [0 0 1 1 1 0 1 0 1 1 1 1 1 0]
[(2, 1), (2, 1), (0, 1), (1, 2), (1, 0), (1, 0), (0, 0), (2, 2), (2, 0), (1, 2), (2, 2), (0, 2), (0, 1), (1, 2)]
Predicted Value: [1]
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-6-338786b109d9> in <module>
     34 from sklearn import metrics
     35 # Model Accuracy, how often is the classifier correct?
---> 36 print("Accuracy:",accuracy_score(label, predicted))
     37 confusionMatrix = confusion_matrix(label, predicted)
     38 print(confusionMatrix)

                              3 frames
/usr/local/lib/python3.9/dist-packages/sklearn/utils/validation.py in check_consistent_length(*arrays)
    395     uniques = np.unique(lengths)
    396     if len(uniques) > 1:
--> 397         raise ValueError(
    398             "Found input variables with inconsistent numbers of samples: %r"
    399             % [int(l) for l in lengths]

ValueError: Found input variables with inconsistent numbers of samples: [14, 1]
```

Output wouldnt work