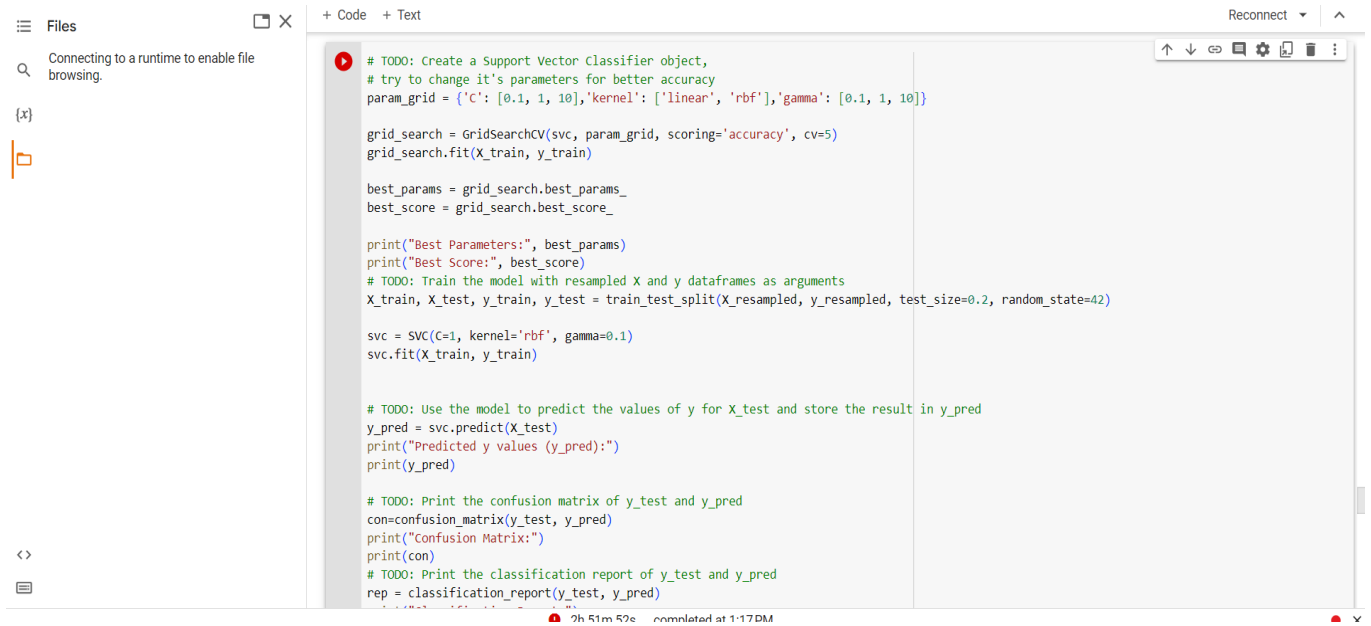


For this part, I tried several times and it really took a long time to run, but in the end it didn't work



```
# TODO: Create a Support Vector Classifier object,
# try to change it's parameters for better accuracy
param_grid = {'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf'], 'gamma': [0.1, 1, 10]}

grid_search = GridSearchCV(svc, param_grid, scoring='accuracy', cv=5)
grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_
best_score = grid_search.best_score_

print("Best Parameters:", best_params)
print("Best Score:", best_score)
# TODO: Train the model with resampled X and y dataframes as arguments
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.2, random_state=42)

svc = SVC(C=1, kernel='rbf', gamma=0.1)
svc.fit(X_train, y_train)

# TODO: Use the model to predict the values of y for X_test and store the result in y_pred
y_pred = svc.predict(X_test)
print("Predicted y values (y_pred):")
print(y_pred)

# TODO: Print the confusion matrix of y_test and y_pred
con=confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(con)
# TODO: Print the classification report of y_test and y_pred
rep = classification_report(y_test, y_pred)
```

2h 51m 52s completed at 1:17 PM

This time It took two hours and fifty one minutes ://