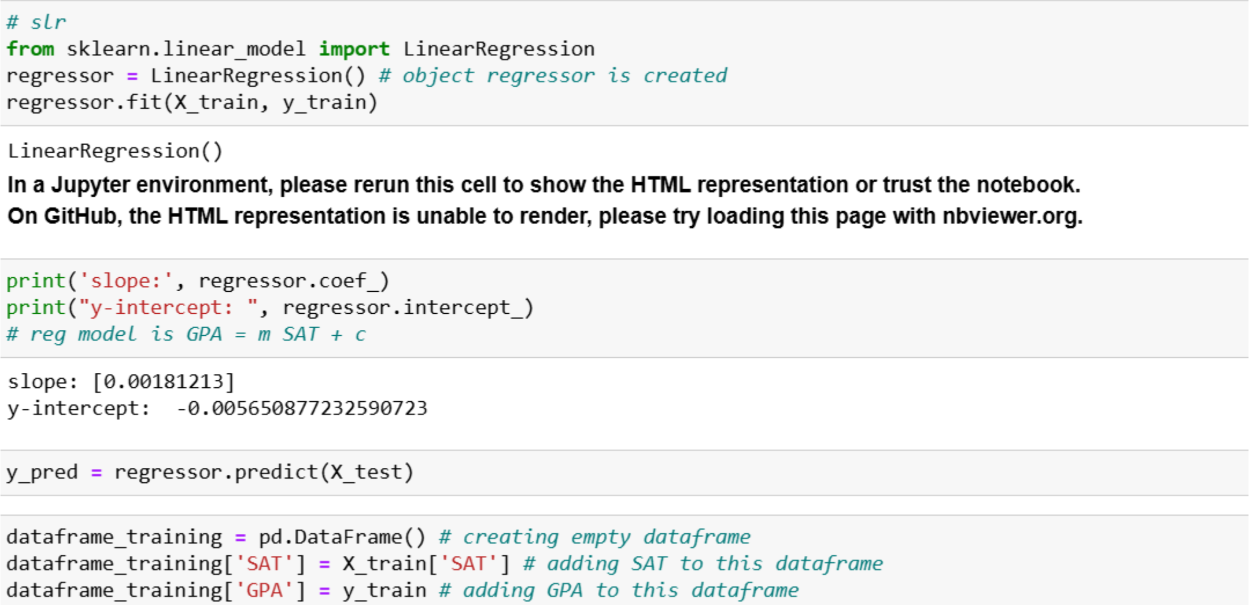


```
# slr
from sklearn.linear_model import LinearRegression
regressor = LinearRegression() # object regressor is created
regressor.fit(X_train, y_train)
```

```
LinearRegression()
```

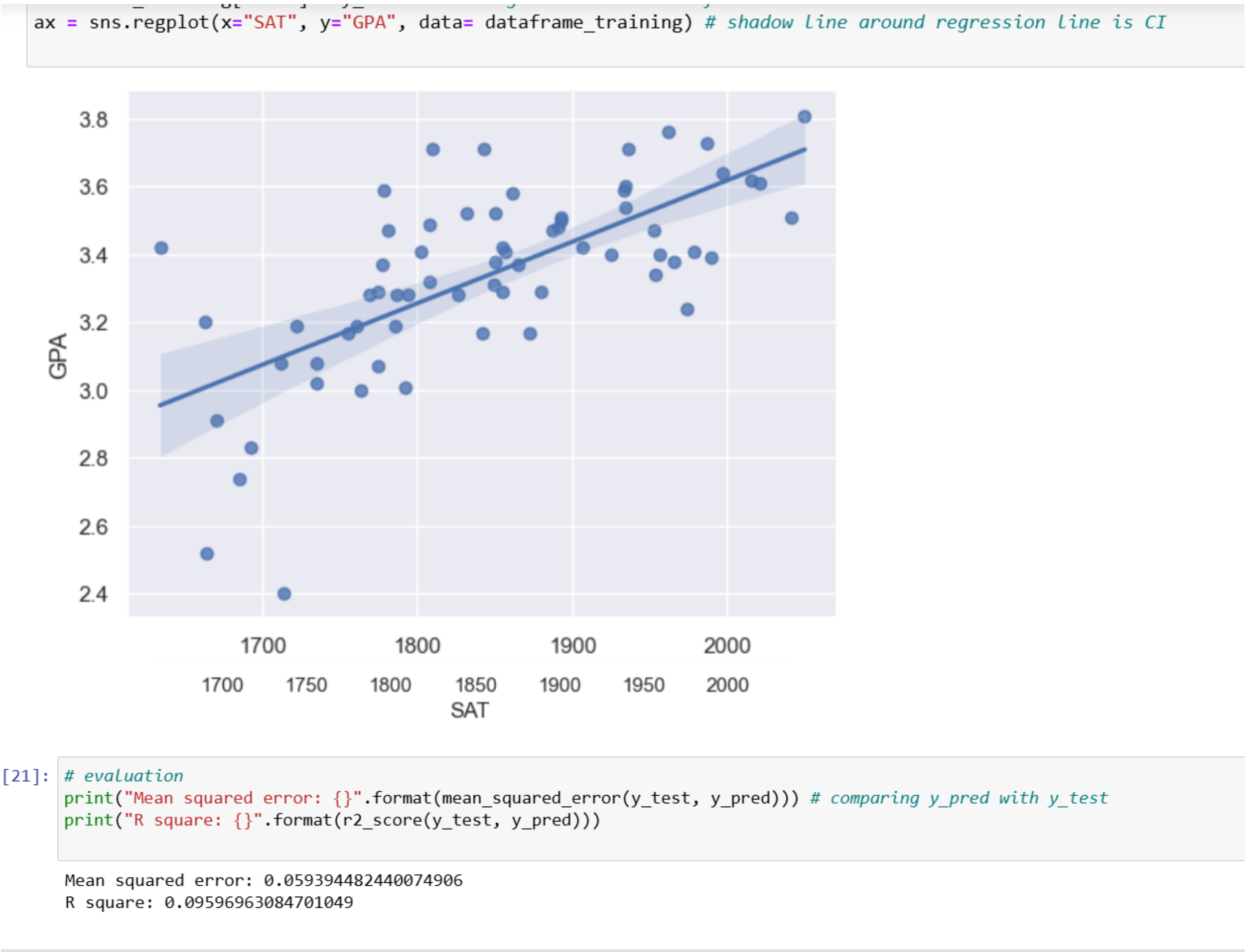
**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

```
print('slope:', regressor.coef_)
print("y-intercept: ", regressor.intercept_)
# reg model is GPA = m SAT + c
```

```
slope: [0.00181213]
y-intercept:  -0.005650877232590723
```

```
y_pred = regressor.predict(X_test)
```

```
dataframe_training = pd.DataFrame() # creating empty dataframe
dataframe_training['SAT'] = X_train['SAT'] # adding SAT to this dataframe
dataframe_training['GPA'] = y_train # adding GPA to this dataframe
```



```
ax = sns.regplot(x="SAT", y="GPA", data= dataframe_training) # shadow line around regression line is CI
```

```
[21]: # evaluation
      print("Mean squared error: {}".format(mean_squared_error(y_test, y_pred))) # comparing y_pred with y_test
      print("R square: {}".format(r2_score(y_test, y_pred)))
```

```
Mean squared error: 0.059394482440074906
R square: 0.09596963084701049
```