Aryan Jagani

Week 6 in-class activities/Lab

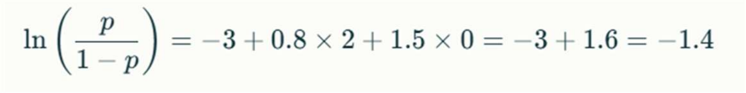
Task1a: Interpreting Logistic Regression model

Given a logistic regression model

$$\ln\left(\frac{p}{1-p}\right) = -3 + 0.8 \times \text{Hours\_Studied} + 1.5 \times \text{Review\_Session}$$

Answer the following questions:
(you may use the provided "logistic regression" notebook and AI assistant.)

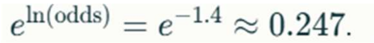
a. Thomas studied for two hours and did not attend the review session. What is his (1) log odds: Substitute Hours_Studied = 2 and Review_Session = 0 into the model.

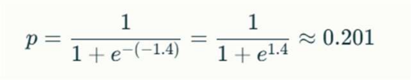


$$\ln\left(\frac{p}{1-p}\right) = -3 + 0.8 \times 2 + 1.5 \times 0 = -3 + 1.6 = -1.4$$

(2) odds, and
(3) likelihood of passing the exam?

- ➔ Log odds:
- ➔ Odds: The odds are calculated as: $e^{\ln(\text{odds})} = e^{-1.4} \approx 0.247.$
- ➔ Likelihood of passing the exam: The probability of passing is given by the logistic function:

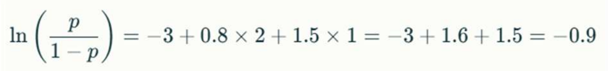




$$p = \frac{1}{1+e^{-(-1.4)}} = \frac{1}{1+e^{1.4}} \approx 0.201$$

b. If Thomas goes to the review session, what is the updated 1) log_odds,
(2) odds, and
(3) likelihood of passing the exam?

- ➔ Log_odds: Substitute Hours_Studied = 2 and Review_Session = 1 into the model.



$$\ln\left(\frac{p}{1-p}\right) = -3 + 0.8 \times 2 + 1.5 \times 1 = -3 + 1.6 + 1.5 = -0.9$$

- ➔ Odds: The odds are e^−0.9≈0.407
- ➔ Likelihood of passing the exam- likelihood of passing the exam is:

$$p\ = \backslash frac\{1\}\{1+e^{\{-(-0.9)\}}\} = \backslash frac\{1\}\{1+e^{\{0.9\}}\} \approx 0.286$$

c. If Thomas studied more or less hours, would the answer change?

  - ➔ Studying more hours increases the log odds, odds, and likelihood of passing because the coefficient for Hours_Studied is positive (0.8).
  - ➔ Studying fewer hours decreases these values.

d. How would you interpret the coefficient of review_session (1.5) from the above experiment?
  - ➔ The coefficient of 1.5 for Review_Session means that attending a review session increases the log odds of passing by 1.5. This translates to a multiplicative increase in odds by e^1.5≈4.482.

e. Using similar reasoning, how would you interpret the coefficient of hours_studied (0.8)
  - ➔ The coefficient of 0.8 for Hours_Studied means that for every additional hour studied, the log odds of passing increase by 0.8. This translates to a multiplicative increase in odds by e^0.8≈2.225.

f. How would you interpret the intercept?
  - ➔ The intercept of -3 represents the log odds of passing when both Hours_Studied and Review_Session are zero. This corresponds to an odds ratio of e^−3≈0.0498 and a probability of passing of about 1/1+e3≈0.047.

g. For someone who studied 8 hours, would you recommend him/her to attend the review session?
    - ➔ For someone who studied 8 hours, attending a review session would increase their log odds, odds, and likelihood of passing, as seen from the positive coefficient of Review_Session.

h. What type of students seems to benefit most from the review session?
  - ➔ Students who benefit most from attending a review session are those who have lower hours studied or are on the borderline of passing, as the review session provides a significant boost in log odds and probability of passing.

Task 1b: Build a logistic regression model

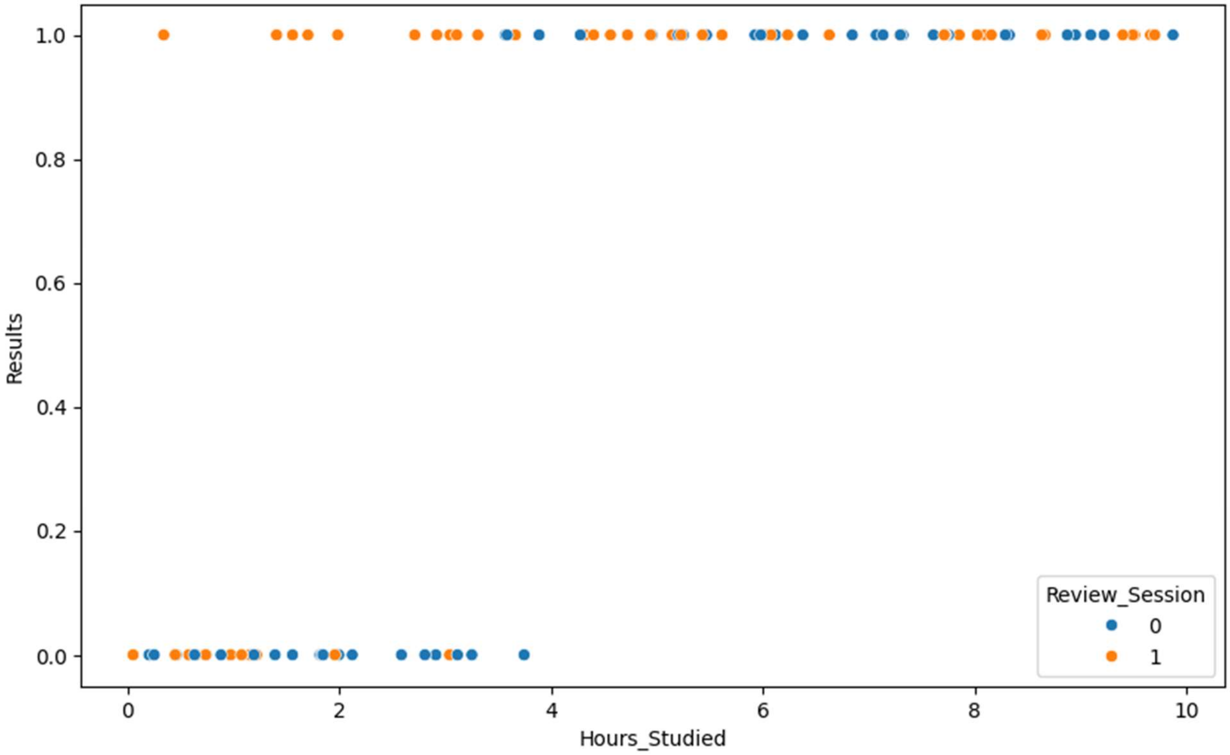
Using the dataset “student_data.csv,” write code to (1) create a visualization of the data , (2) fit a model using logistic regression, (3) output model coefficients and performance metrics such as accuracy and AUC and ROC; **NOTE: For this exercise, you will train and test on the same given dataset, instead of doing train/test split. Make sure you give the correct GPT prompt.**

  - ➔ **Github link-**

https://github.com/AryanJ09/IS733_Class/blob/main/01272025_CW/CW-03-10-2025/Task1_lab6.ipynb

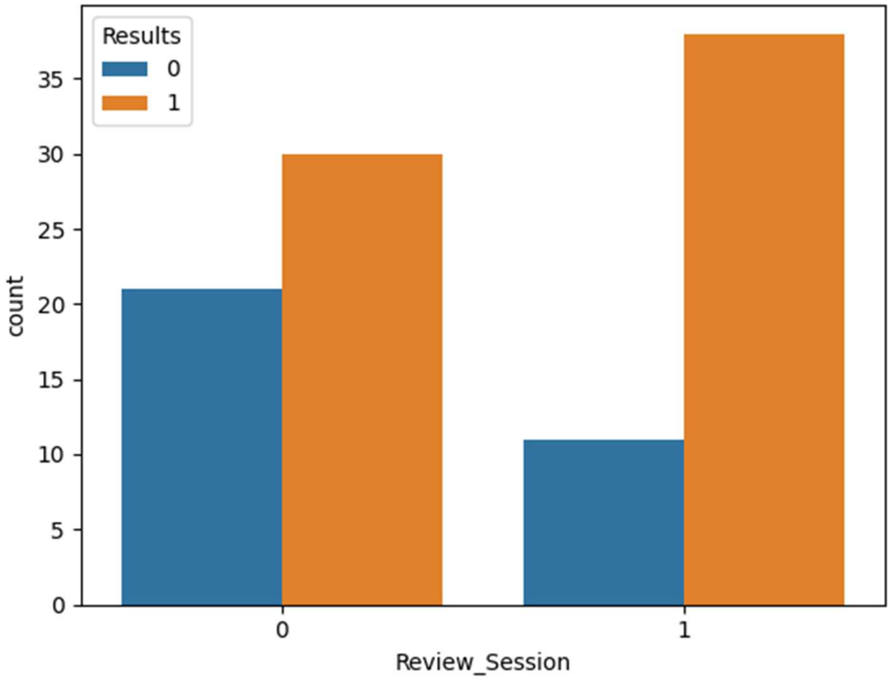
- Visualisation Of the data
- Scatter plot: This plot suggests that both studying more hours and attending review sessions positively influence success. However, attending review sessions might provide an additional advantage, as some students achieve success even with fewer study hours if they attended a session.
- This scatter plot visualizes the relationship between the number of hours studied (Hours_Studied) and the results (Results), which appear to be binary (0 or 1, likely indicating failure or success). The data points are further categorized based on whether a review session was attended (Review_Session), represented by two colors: blue for 0 (no review session) and orange for 1 (attended review session).

```
import seaborn as sns
# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x="Hours_Studied", y="Results",
hue="Review_Session", data=df)
plt.show()
```



➔ This **count plot** visualizes the relationship between **Review_Session** (whether students attended a review session) and **Results** (whether they passed or failed).

```
#Count Plot (for Review_Session)

sns.countplot(x="Review_Session", hue="Results", data=df)
plt.show()
```

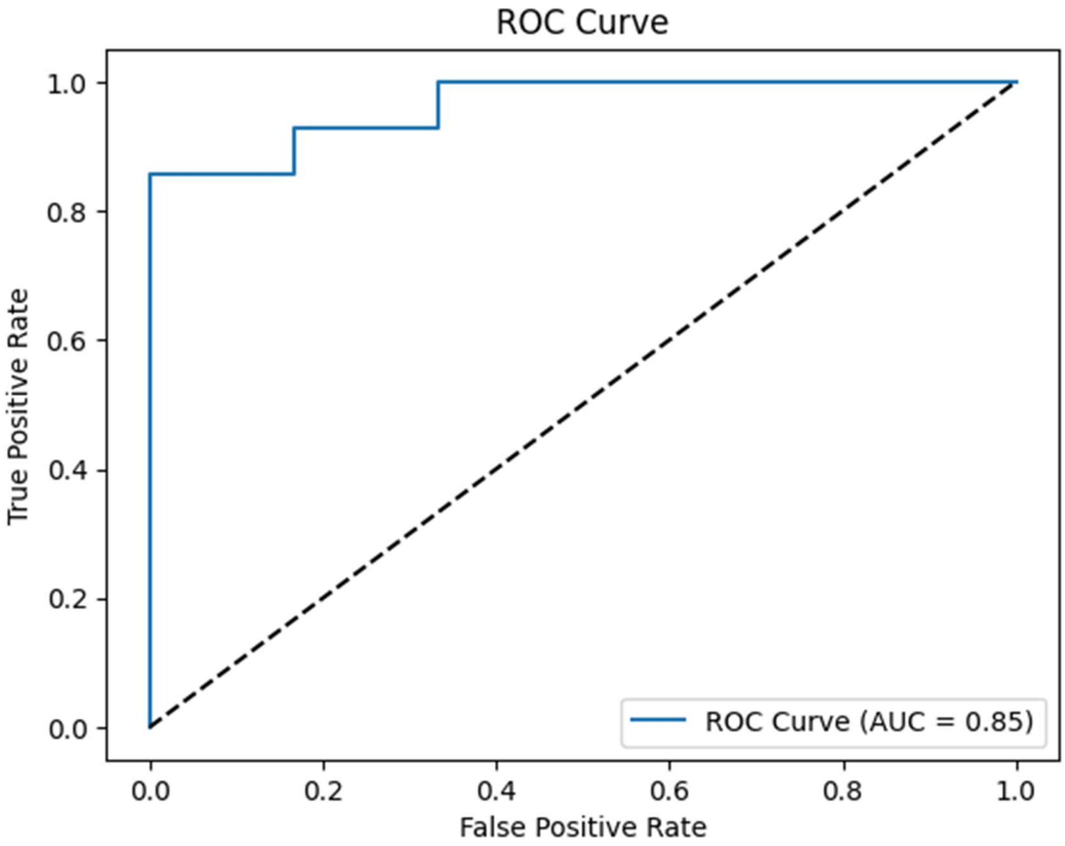


2) Fit a model using logistic Regression

→ Since the task involves training and testing on the same dataset, we will not split the data into training and testing sets.

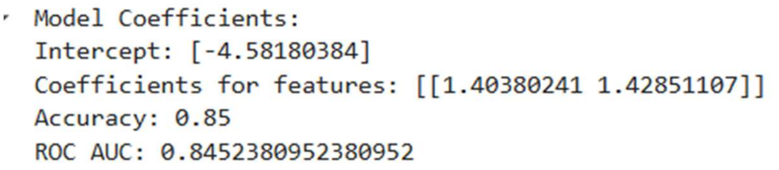
```
#fit a model using logistic regression

import matplotlib.pyplot as plt
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Initialize and train the Logistic Regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
```

```
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print(f"ROC AUC: {roc_auc}")

# Generate ROC curve
fpr, tpr, thresholds = roc_curve(y_test, model.predict_proba(X_test)[:,
1])
plt.plot(fpr, tpr, label=f'ROC Curve (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], 'k--')  # Random guess line
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend()
plt.show()
```

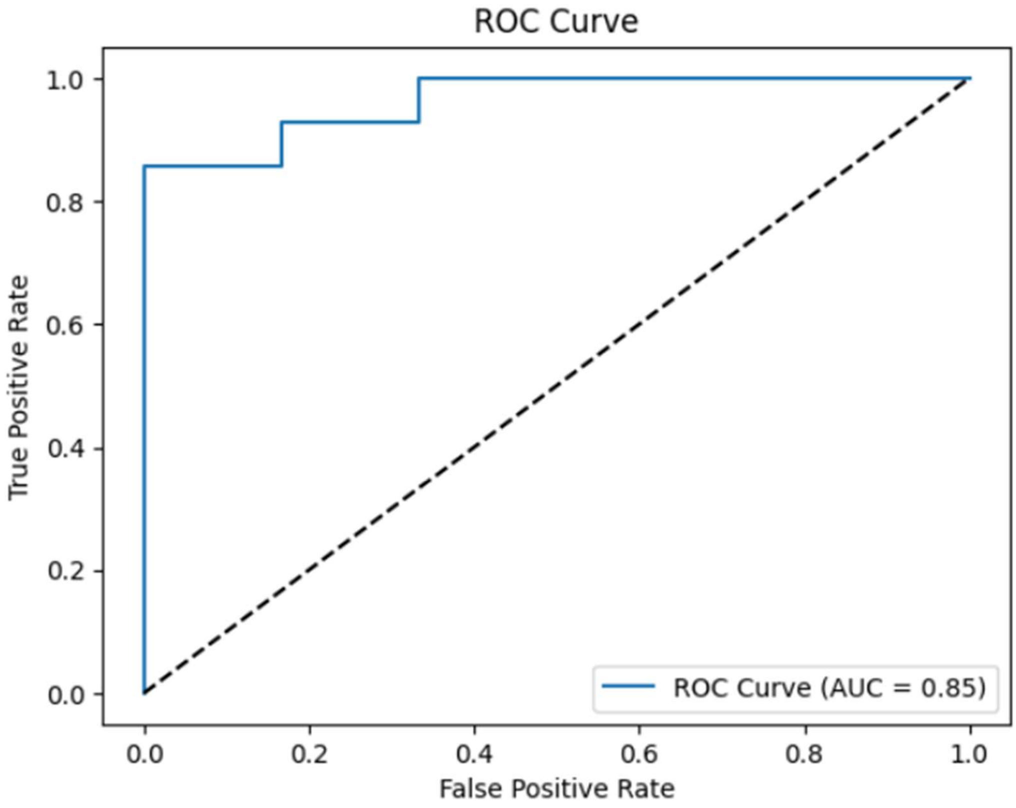


(3) output model coefficients and performance metrics such as accuracy and AUC and ROC

```
#output model coefficients and performance metrics such as accuracy and AUC and ROC

import matplotlib.pyplot as plt
# Print model coefficients
print("Model Coefficients:")
print(f"Intercept: {model.intercept_}")
print(f"Coefficients for features: {model.coef_}")

# ... (rest of your existing code) ...

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print(f"ROC AUC: {roc_auc}")
```



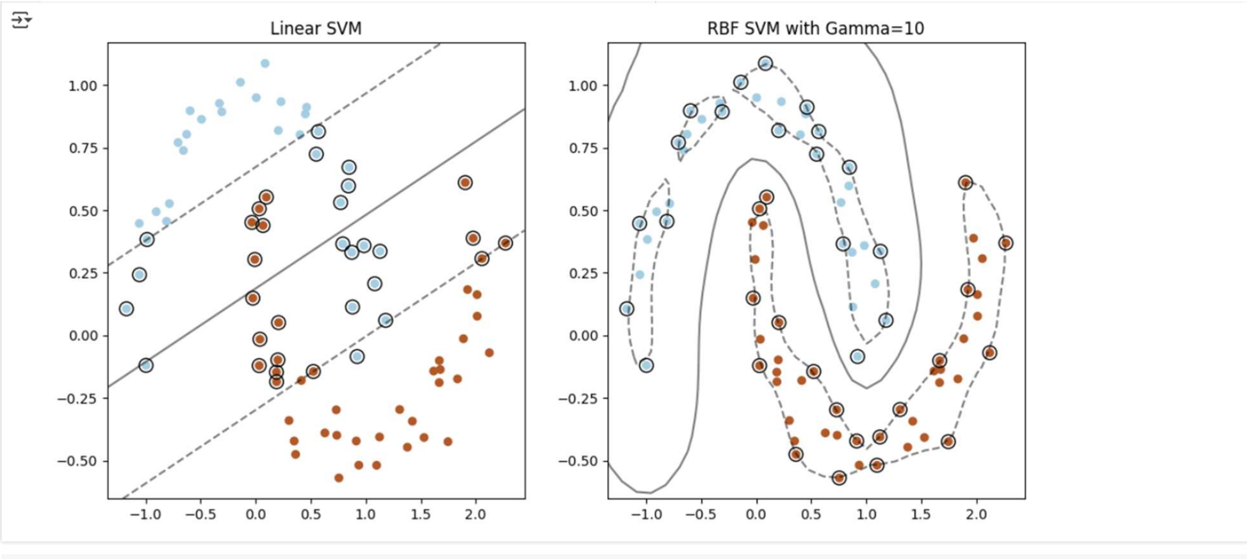
```
Model Coefficients:
Intercept: [-4.58180384]
Coefficients for features: [[1.40380241 1.42851107]]
Accuracy: 0.85
ROC AUC: 0.8452380952380952
```



Task 2: Understanding and Prevent Overfitting in the context of SVM

Write code to fit a Support Vector Machine model using (1) linear kernel and (2) RBF kernel. For the RBF kernel, use grid search to find the best gamma parameter using k-fold cross-validation.

Github link-
https://github.com/AryanJ09/IS733_Class/blob/main/01272025_CW/CW-03-10-2025/task2_lab6.ipynb



Submission: 1(a) writeup in a doc; 1(b) and (2) Python Notetook uploaded to GitHub and submit a link to Blackboard; link to chatGPT log