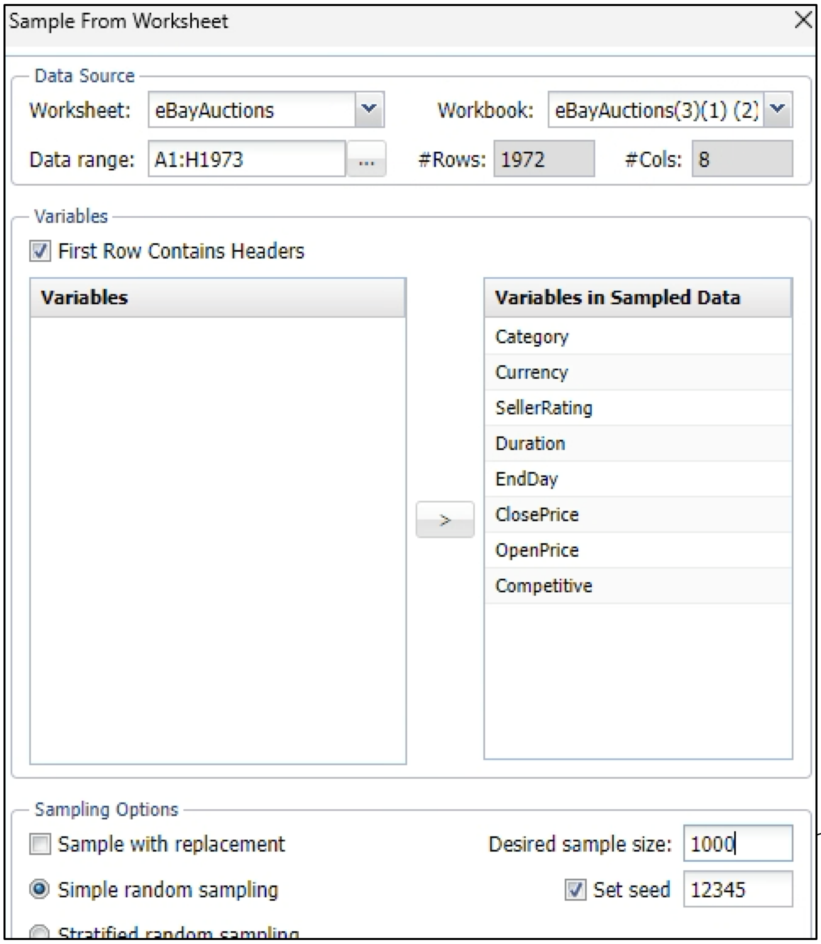
Question 1: Start by getting a sample of 1000 records (Please set seed to 12345). What is the record number for the last selected record in the sample? (1.5 points)



Sample From Worksheet

Data Source
Worksheet: eBayAuctions
Workbook: eBayAuctions(3)(1) (2)
Data range: A1:H1973
#Rows: 1972
#Cols: 8

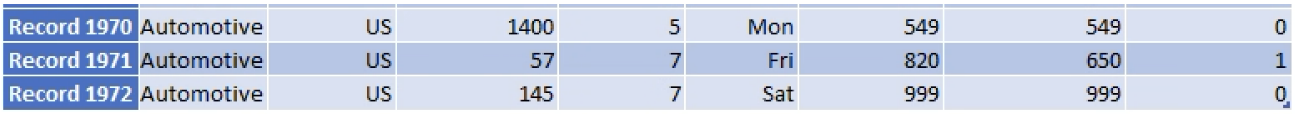
Variables
☑ First Row Contains Headers

Variables	Variables in Sampled Data
	Category
	Currency
	SellerRating
	Duration
	EndDay
	ClosePrice
	OpenPrice
	Competitive

Sampling Options
□ Sample with replacement
Desired sample size: 1000
● Simple random sampling
☑ Set seed 12345
○ Stratified random sampling

Set sample to 1000 and seed is 12345 as per question.

Record number for the last selected record in the sample: Record 1972

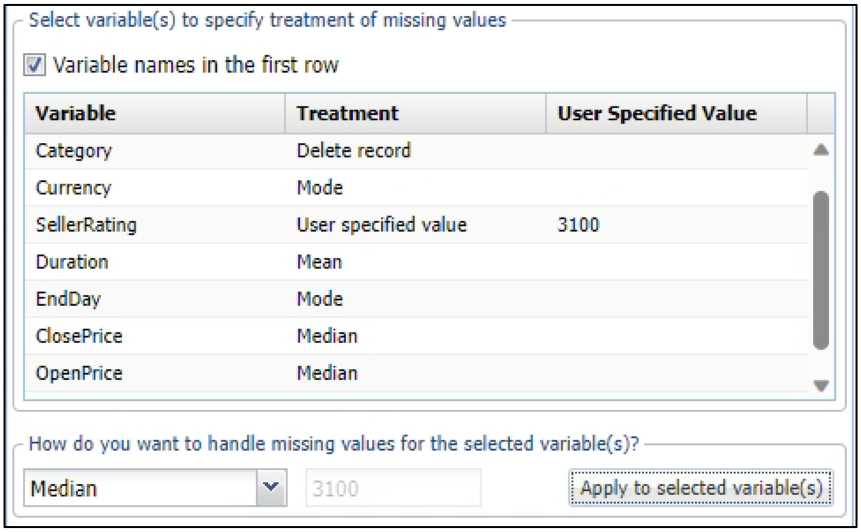


Record 1970	Automotive	US	1400	5	Mon	549	549	0
Record 1971	Automotive	US	57	7	Fri	820	650	1
Record 1972	Automotive	US	145	7	Sat	999	999	0

Question 2: Check to see whether there are any missing values in the dataset. Follow this table to treat the missing values:

Variable	**Treatment**
Category	Delete record
Currency	Use mode
SellerRating	Use the value 3100
Duration	Use mean value
EndDay	Use mode
ClosePrice	Use median
OpenPrice	Use median
Competitive	Use median

1. Click on **Data Mining** and then under Data Analysis click on *Transform* and then select **Missing Data Handling.**
2. For Category click on Delete and then press Apply to Selected variable (s).
3. We do this to the variables based on the above instructions and the result is as follows:



Select variable(s) to specify treatment of missing values

☑ Variable names in the first row

Variable	Treatment	User Specified Value
Category	Delete record	
Currency	Mode	
SellerRating	User specified value	3100
Duration	Mean	
EndDay	Mode	
ClosePrice	Median	
OpenPrice	Median	

How do you want to handle missing values for the selected variable(s)?

Median 3100

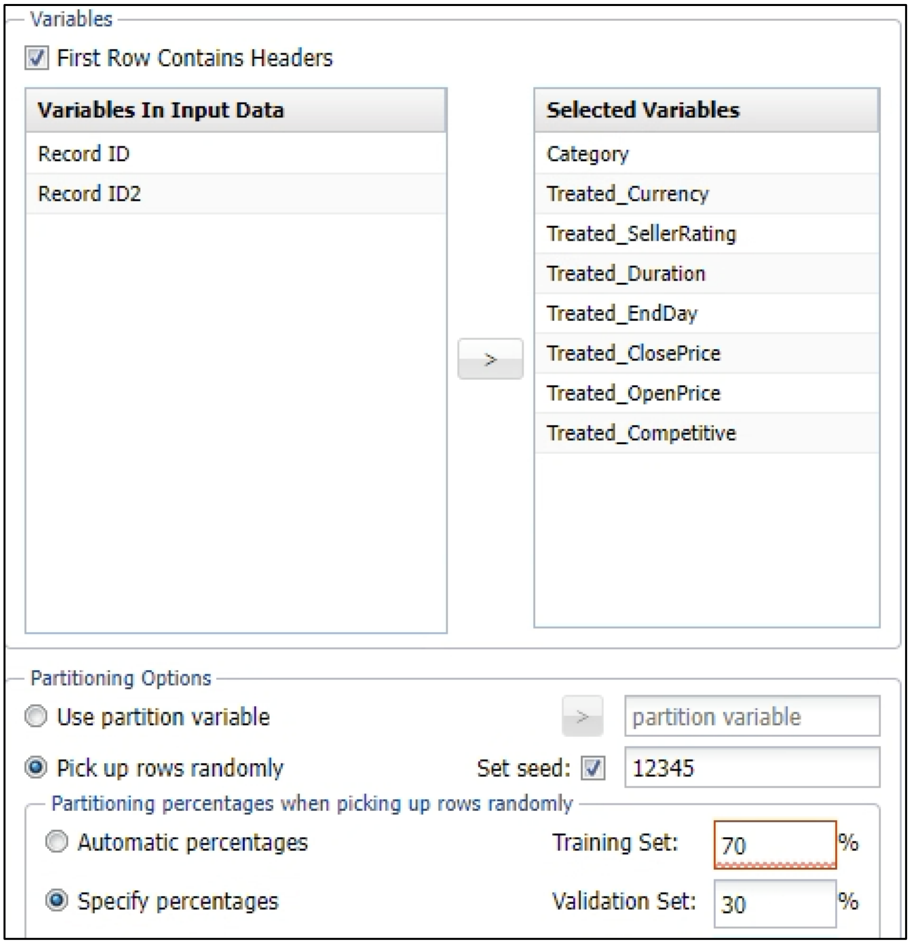
4. Click OK and a new tab open called computation.

How many records are treated for each of the variables? (2 points)

Imputer Parameters									
Variable	**Record ID**	**Category**	**Currency**	**SellerRating**	**Duration**	**End Day**	**Close Price**	**Open Price**	**Competitive**
Reduction Type	**NONE**	**DELETE RECORD**	**MODE**	**VALUE (3100)**	**MEAN**	**MODE**	**MEDIAN**	**MEDIAN**	**MEDIAN**
# Records Treated	**0**	**0**	**11**	**5**	**8**	**17**	**12**	**12**	**9**

Total Variables Treated: 11 + 5 + 8 + 17 + 12 + 12 + 9 = 74

Question 3: Partition the treated sample; assign 70% for training and 30% for validation part. Do not include Record ID and Record ID2 in your partitioning. How many rows are in the Validation part? (1.5 points)



Variables

☑ First Row Contains Headers

Variables In Input Data

Record ID

Record ID2

Selected Variables

Category

Treated_Currency

Treated_SellerRating

Treated_Duration

Treated_EndDay

Treated_ClosePrice

Treated_OpenPrice

Treated_Competitive

Partitioning Options

○ Use partition variable — partition variable

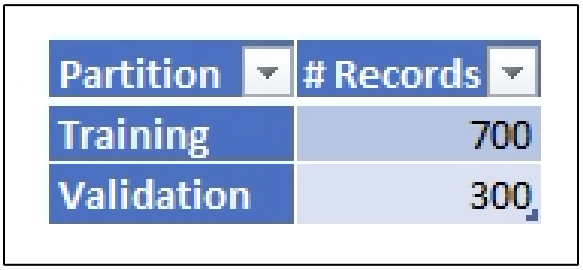
● Pick up rows randomly — Set seed: ☑ 12345

Partitioning percentages when picking up rows randomly

○ Automatic percentages — Training Set: 70 %

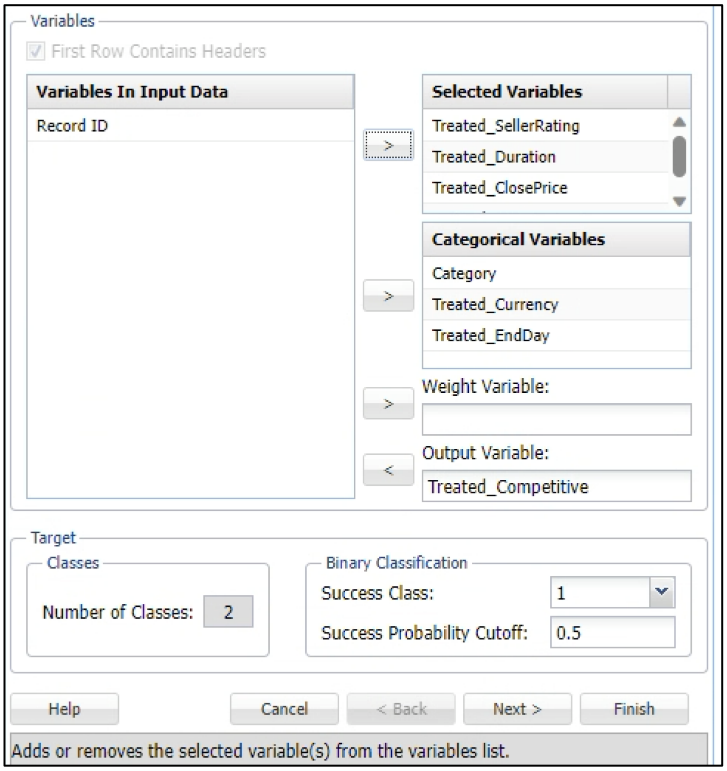
● Specify percentages — Validation Set: 30 %

Once you specify the percentages 70 (training) and 30 (validation) which add up to 100%, click on OK and then you will see the partition summary. We have 300 records in validation set out of the 1000 and 700 in the training set.



Partition	# Records
Training	700
Validation	300

Question 4: Create a logistic Regression model to predict Competitive variable based on all other variables except Record ID. Set cut-off probability 50%. Create Detail and Summary Reports for Scoring Training and Validation data.


Variables

☑ First Row Contains Headers

Variables In Input Data

Record ID

Selected Variables

Treated_SellerRating
Treated_Duration
Treated_ClosePrice

Categorical Variables

Category
Treated_Currency
Treated_EndDay

Weight Variable:

Output Variable: Treated_Competitive

Target

Classes

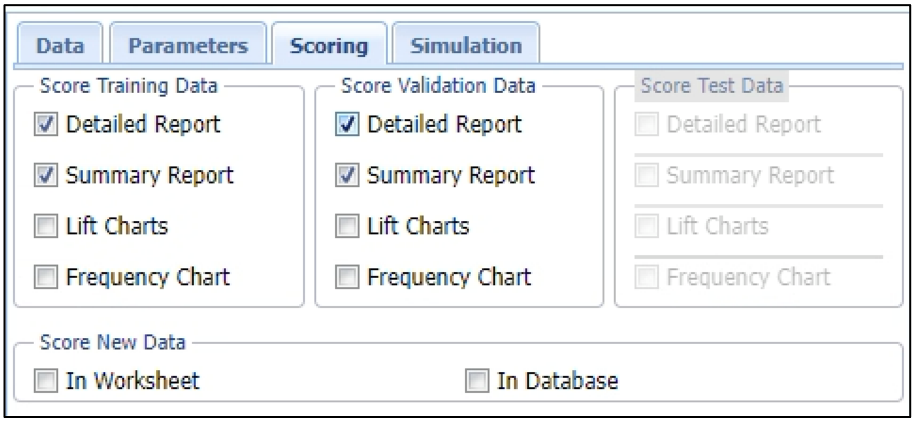
Number of Classes: 2

Binary Classification

Success Class: 1

Success Probability Cutoff: 0.5

Adds or removes the selected variable(s) from the variables list.



Score Training Data

☑ Detailed Report
☑ Summary Report
□ Lift Charts
□ Frequency Chart

Score Validation Data

☑ Detailed Report
☑ Summary Report
□ Lift Charts
□ Frequency Chart

Score Test Data

□ Detailed Report
□ Summary Report
□ Lift Charts
□ Frequency Chart

Score New Data

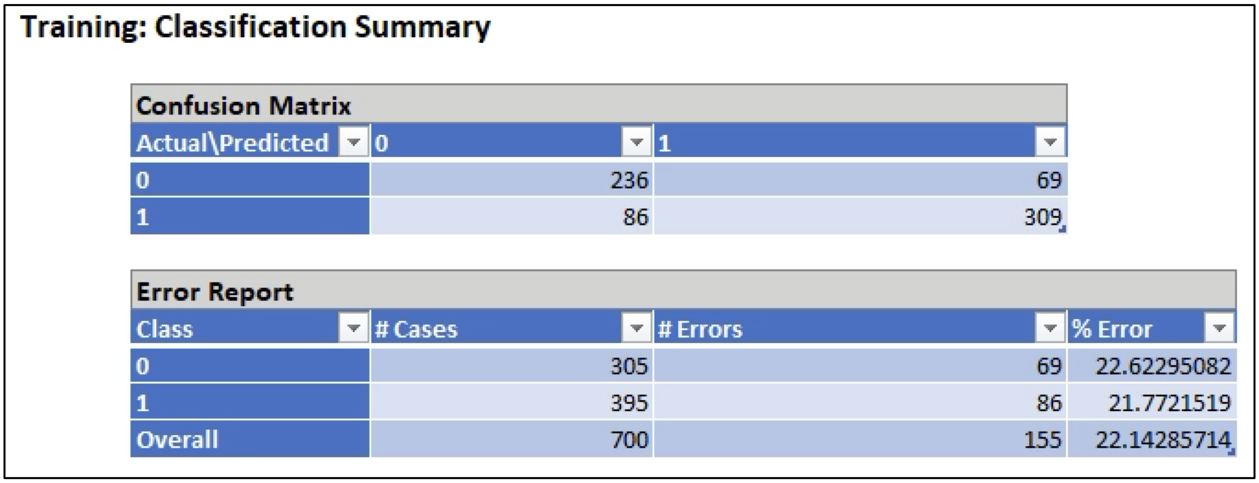
□ In Worksheet
□ In Database

Then click Finish (x2)

What is the percentage of error for the training partition? How many false positives and false negatives do we have? Has the model worked better for training or validation part and why? (1.5 points)

Percentage of error for the training partition:

- The percentage of error can be calculated using the following formula:
- Percentage Error = (Number of Errors / Total Number of Cases) * 100
  - Number of Errors = 155 (69 false positives + 86 false negatives)
  - Total Number of Cases = 700 (Sum of all cases in the confusion matrix)
- Percentage Error = (155 / 700) * 100 = 22.14285714%



Training: Classification Summary

Confusion Matrix

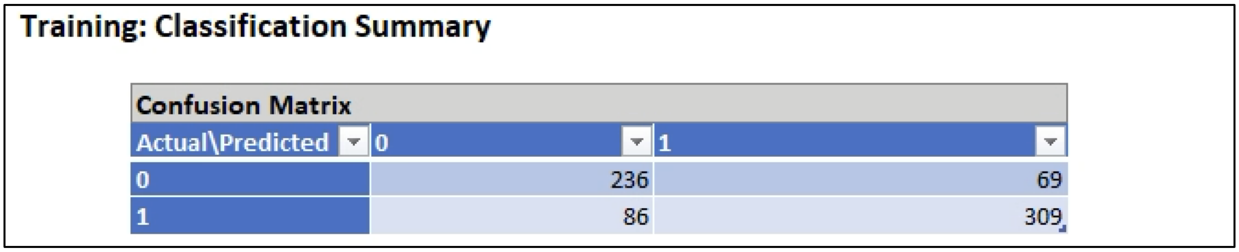
Actual\Predicted	0	1
0	236	69
1	86	309

Error Report

Class	# Cases	# Errors	% Error
0	305	69	22.62295082
1	395	86	21.7721519
Overall	700	155	22.14285714

False Positive and False Negatives:

The false positives are typically found in the row where the actual value is 0, and the predicted value is 1.



Training: Classification Summary

Confusion Matrix

Actual\Predicted	0	1
0	236	69
1	86	309

The false positives are represented by the value 69. These are instances where the actual class was 0 (negative). 69 cases were incorrectly classified as positive when they should have been classified as negative.

To determine whether the model has worked better for the training or validation part, we can compare performance metrics between the two datasets. In general, a model that performs better on the validation dataset is preferred because it indicates that the model is more likely to generalize well to new, unseen data.

- Training:
  - Accuracy (%correct): 77.86%
  - Sensitivity (Recall): 78.23%
  - Specificity: 77.38%
  - Precision: 81.75%
  - F1 Score: 79.95%

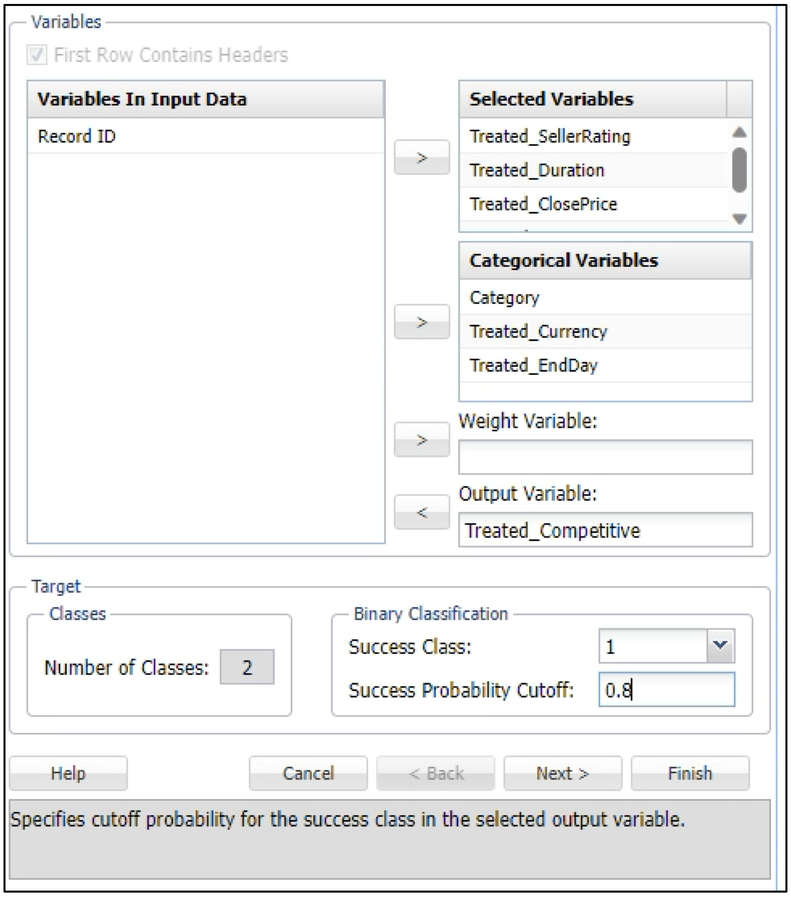
- Validation:
  - Accuracy (%correct): 76%
  - Sensitivity (Recall): 78.62%
  - Specificity: 73.05%
  - Precision: 76.69%
  - F1 Score: 77.64%

Based on these metrics, the model performs **slightly better on the training dataset** in terms of accuracy, specificity, and precision. However, the differences between the training and validation results are not substantial. The validation results are reasonably close to the training results.

It is important to note that the model's performance on the validation dataset is still relatively good, and the differences in performance metrics between training and validation are not significantly large. Therefore, the model seems to generalize reasonably well to the validation data, indicating that it may perform well on unseen data.

Question 5: Redo step 4 and this time, set cut-off probability equal to 80%. Which of the models of the current step and step 4 are better for prediction? (2 points)

Set the cut-off to 80%, the default option is always 0.5 (50%) so change this to 0.8 (80%) as per question.



Variables

☑ First Row Contains Headers

Variables In Input Data: Record ID

Selected Variables: Treated_SellerRating, Treated_Duration, Treated_ClosePrice

Categorical Variables: Category, Treated_Currency, Treated_EndDay

Weight Variable:

Output Variable: Treated_Competitive

Target

Classes

Number of Classes: 2

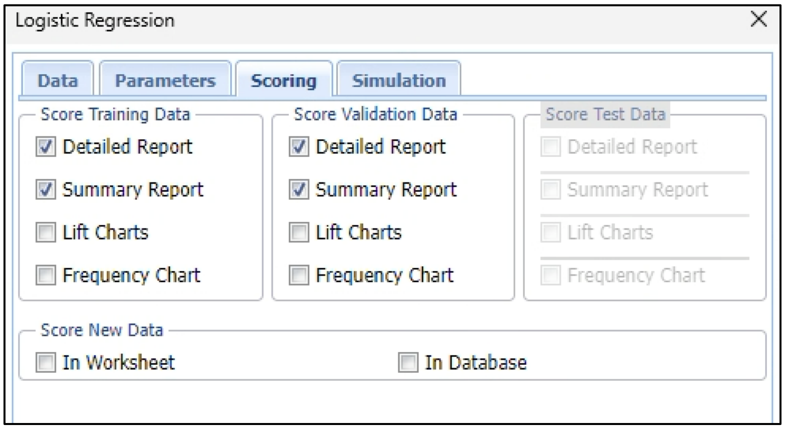
Binary Classification

Success Class: 1

Success Probability Cutoff: 0.8

Specifies cutoff probability for the success class in the selected output variable.

Generate the reports:



Logistic Regression

Score Training Data

☑ Detailed Report
☑ Summary Report
□ Lift Charts
□ Frequency Chart

Score Validation Data

☑ Detailed Report
☑ Summary Report
□ Lift Charts
□ Frequency Chart

Score Test Data

□ Detailed Report
□ Summary Report
□ Lift Charts
□ Frequency Chart

Score New Data

□ In Worksheet
□ In Database

Click Finish (x2)

Which is the better model for prediction?

Model with 80% Cutoff (0.8):

- Accuracy (%correct): 68% (Training), 73.33% (Validation)

- Sensitivity (Recall): 44.81% (Training), 52.83% (Validation)

- Specificity: 98.03% (Training), 96.45% (Validation)

- Precision: 96.72% (Training), 94.38% (Validation)

- F1 Score: 61.25% (Training), 67.74% (Validation)

Model with 50% Cutoff (0.5):

- Accuracy (%correct): 77.86% (Training), 76% (Validation)

- Sensitivity (Recall): 78.23% (Training), 78.62% (Validation)

- Specificity: 77.38% (Training), 73.05% (Validation)

- Precision: 81.75% (Training), 76.69% (Validation)

- F1 Score: 79.95% (Training), 77.64% (Validation)

Comparing the two models:

- Accuracy: The 50% cutoff model has higher accuracy on both the training and validation datasets.

- Sensitivity (Recall): The 50% cutoff model also has higher sensitivity on both datasets.

- Specificity: The 80% cutoff model has higher specificity, indicating fewer false positives.

- Precision: The 50% cutoff model has higher precision on both datasets.

- F1 Score: The 50% cutoff model has a higher F1 score on both datasets.

Based on these performance metrics, the model with the 50% cutoff (0.5) generally performs better in terms of accuracy, recall, precision, and F1 score on both the training and validation datasets. Therefore, the 50% cutoff model is better for prediction in this context.

The 80% cutoff model sacrifices recall for higher precision, which means it is more conservative in making positive predictions. The 50% cutoff model provides a better balance between precision and recall and is more suitable for your prediction task, especially if you want to capture more true positive cases.

Question 6: Based on the model that you created in step 5, for the first three records of validation partition, what is the prediction and what is the actual value of target? (1.5 points)

Based on the new model at 80% cut off (0.8):

1. Record 932:

   - Prediction: 1

   - Actual Value: 1

2. Record 873:

   - Prediction: 0

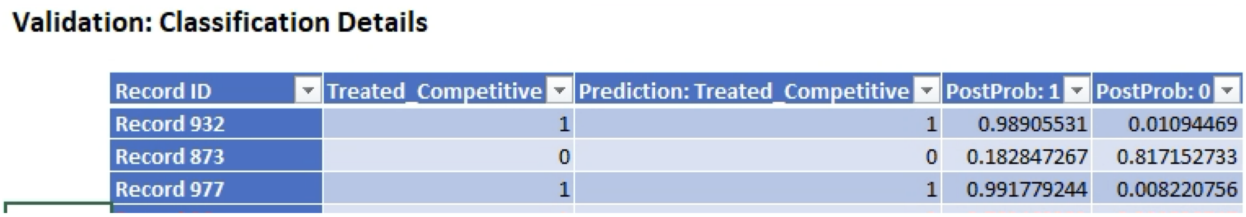
   - Actual Value: 0

3. Record 977:

   - Prediction: 1

   - Actual Value: 1

For these three records, the model's predictions match the actual values, as indicated below.



Validation: Classification Details

Record ID	Treated_Competitive	Prediction: Treated_Competitive	PostProb: 1	PostProb: 0
Record 932	1	1	0.98905531	0.01094469
Record 873	0	0	0.182847267	0.817152733
Record 977	1	1	0.991779244	0.008220756