Data Science Assignment-04:

Nuthan Reddy Vaddi Reddy

UTA ID:1002175133

---

1. Data from our lives:

Insurance company:

Consider a normal situation of a person looking for insurance. In this case, classification is crucial for insurance companies to determine risk and set right premium rates. To divide an individual into different risk groups, the system of categorization considers a number of conditions such as age, health history, and lifestyle. As an example, a person who leads a healthy lifestyle may be classified as a low-risk policyholder, giving them lower premiums. Individuals with existing medical conditions, on the other hand, might be at a higher risk. The categorization guarantees fair pricing and personalized coverage, allowing customers to make gained insurance selections based on their own unique needs and risk profiles.

Job market:

In the real-life scenario of students entering the job market, classification is an important part of the recruiting process. Classification models are frequently used by recruiters to evaluate and categorize job applicants based on their qualifications,

---

skills, and experiences. The process considers a number of features, including education, right job experience, and skills with particular skills, to classify applicants as entry-level, mid-level, or senior roles.

A student with an outstanding academic background, suitable internships, and specific skills, for example, may be considered a top-tier in the applicants, eligible for senior-level positions. Those with little work experience, on the other hand, may be marked as beginning applicants. The categorized look at improves the recruiting process for companies, allowing them to quickly identify the best applicants for certain tasks.

In addition, students benefit from this classification method because it explains their competitiveness in the job market and lets them adjust their job searches accordingly. Understanding your categorization supports graduates in understanding the job market more successfully and boosts their chances of getting relevant jobs in their chosen fields, whether it's writing focused resumes or preparing for particular interview scenarios. Overall, the use of categorization models in the recruiting process helps to provide more efficiently.

2 preprocessing:

2.1 Replace ['gas', 'diesel'] string values to [0, 1]:

Replacing the gas and diesel in fuel_type with 0 and 1 by using the replace function.

	fuel_type	wheel_base	length	width	heights	curb_weight	engine_size	bore	stroke	compression	horse_power	peak_
0	0	88.6	168.8	64.1	48.8	2548	130	3.47	2.68	9.0	111.0	5(
1	0	88.6	168.8	64.1	48.8	2548	130	3.47	2.68	9.0	111.0	5(
2	0	94.5	171.2	65.5	52.4	2823	152	2.68	3.47	9.0	154.0	5(
3	0	99.8	176.6	66.2	54.3	2337	109	3.19	3.40	10.0	102.0	55
4	0	99.4	176.6	66.4	54.3	2824	136	3.19	3.40	8.0	115.0	55
...	...	...	...	...	...	...	...	...	...	...	...	...
190	0	109.1	188.8	68.9	55.5	2952	141	3.78	3.15	9.5	114.0	54
191	0	109.1	188.8	68.8	55.5	3049	141	3.78	3.15	8.7	160.0	53
192	0	109.1	188.8	68.9	55.5	3012	173	3.58	2.87	8.8	134.0	55
193	1	109.1	188.8	68.9	55.5	3217	145	3.01	3.40	23.0	106.0	48
194	0	109.1	188.8	68.9	55.5	3062	141	3.78	3.15	9.5	114.0	54

195 rows × 15 columns

2.2 Define your X and y: your dependent variable is fuel_type, the rest of the variables are your independent variables

Placing the dependent variable as fuel_type feature and independent variable as remaining features as independent variables.

```
X = df.drop(['fuel_type'], axis=1)
y = df['fuel_type']
X.head()
```

	wheel_base	length	width	heights	curb_weight	engine_size	bore	stroke	compression	horse_power	peak_rpm	city_m
0	88.6	168.8	64.1	48.8	2548	130	3.47	2.68	9.0	111.0	5000.0	
1	88.6	168.8	64.1	48.8	2548	130	3.47	2.68	9.0	111.0	5000.0	
2	94.5	171.2	65.5	52.4	2823	152	2.68	3.47	9.0	154.0	5000.0	
3	99.8	176.6	66.2	54.3	2337	109	3.19	3.40	10.0	102.0	5500.0	
4	99.4	176.6	66.4	54.3	2824	136	3.19	3.40	8.0	115.0	5500.0	

2.3 Split your data into a training and testing set. Use test_size=0.3, random_state=746 !:

Splitting the data into a training and testing set with test_size=0.3 and random_state=746 by using X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=746)

```
X_train.head()
```

	wheel_base	length	width	heights	curb_weight	engine_size	bore	stroke	comprassion	horse_power	peak_rpm	city_
60	110.0	190.9	70.3	56.5	3515	183	3.58	3.64	21.5	123.0	4350.0	
191	109.1	188.8	68.8	55.5	3049	141	3.78	3.15	8.7	160.0	5300.0	
79	96.3	172.4	65.4	51.6	2405	122	3.35	3.46	8.5	88.0	5000.0	
129	93.7	157.9	63.6	53.7	2120	108	3.62	2.64	8.7	73.0	4400.0	
110	108.0	186.7	68.3	56.0	3130	134	3.61	3.21	7.0	142.0	5600.0	

3. Classification:

3.1 Use Logistic regression to classify your data. Print/report your confusion matrix, classification report and AUC:

Confusion Matrix :

The confusion matrix shows that the model correctly predicted all instances of both classes. There were 50 true positives and 9 true negatives, indicating a perfect classification.

True Positive (TP): 50 instances were correctly predicted as class 0.

False Negative (FN): 0 instances that actually belong to class 1 were incorrectly predicted as class 0.

True Negative (TN): 9 instances were correctly predicted as class 1.

False Positive (FP): 0 instances that actually belong to class 0 were incorrectly predicted as class 1.

The confusion matrix provides full information on the predictions made by the model:There are 50 cases of class 0 and 9 cases of class 1 represented by the values along the main diagonal, which show the correctly defined examples. All of the values off the diagonal are zeros, indicating there are no incorrect classifications in the predictions.

Classification Report:

The classification report offers a comprehensive summary of the model's performance, including precision, recall, F1-score, and support for each class.

precision, recall, and F1-score are all perfect (1.00) for both classes, indicating an excellent performance in terms of classification.

F1=2*Precision×Recall %Precision+Recall

precision out of total predictions how many are correct predictions, predicted 100% were correctly predicted as class 1 and class 0.

Recall means out of total 0's how many are correctly predicted 100% were correctly predicted in class 0 and class 1.

With an overall accuracy of 100%, both classes have excellent precision, recall, and F1-score. This indicates that the logistic regression model did an excellent job of distinguishing between the two classes.

Accuracy: 1.00 The ratio of correctly predicted instances to the total instances. Macro Average: Precision: 1.00 (average precision across classes) Recall: 1.00 (average recall across classes) F1-Score: 1.00 (average F1-score across classes) Weighted Average: Precision: 1.00 (weighted by the number of instances in each class) Recall: 1.00 (weighted by the number of instances in each class) F1-Score: 1.00 (weighted by the number of instances in each class)

AUC:

AUC provides a measure of how properly the model can distinguish between the two classes. The AUC score in this case is 1.0, which indicates excellent class discrimination between 2 classes.

With the given dataset, the Logistic Regression model performed very well, achieving perfect precision, recall, and F1-score for both classes. The model's robustness is further supported by the AUC score of 1.0. The results suggest the

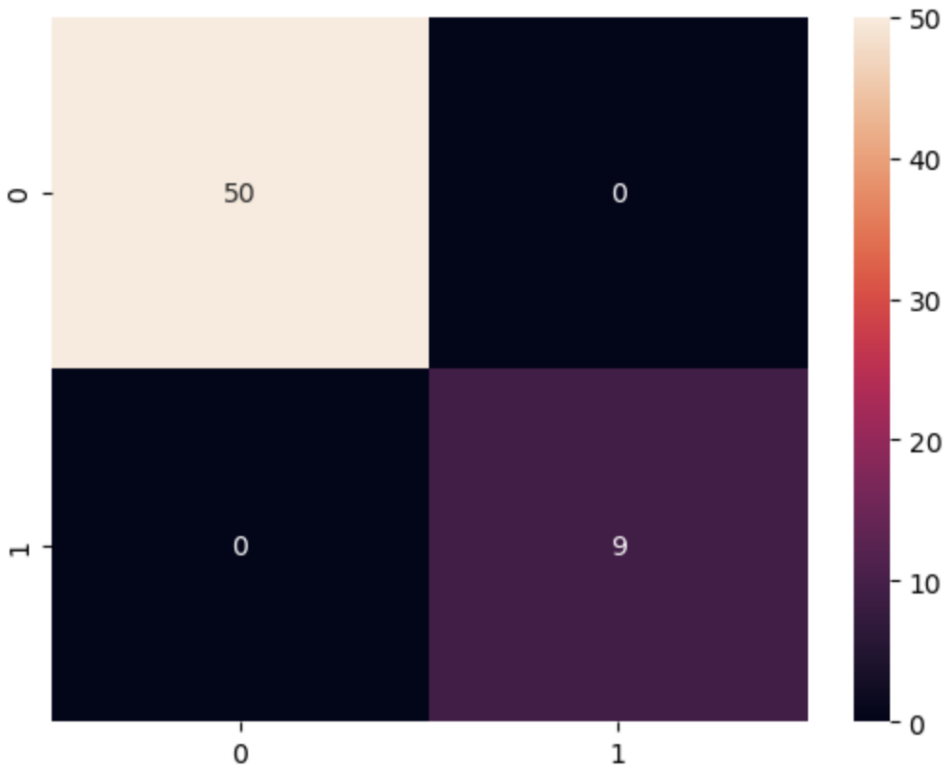
Logistic Regression model is an excellent match for this classification task.

```
Confusion Matrix:
 [[50  0]
 [ 0  9]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        50
           1       1.00      1.00      1.00         9

    accuracy                           1.00        59
   macro avg       1.00      1.00      1.00        59
weighted avg       1.00      1.00      1.00        59

AUC Score: 1.0
```



3.2Use Naive Bayes to classify your data. Print/report your confusion matrix, classification report and AUC:

Confusion matrix:

The confusion matrix shows that the model correctly predicted all instances of both classes. There were 50 true positives and 9 true negatives, indicating a perfect classification.

True Positive (TP): 50 instances were correctly predicted as class 0.

False Negative (FN): 0 instances that actually belong to class 1 were incorrectly predicted as class 0.

True Negative (TN): 9 instances were correctly predicted as class 1.

False Positive (FP): 0 instances that actually belong to class 0 were incorrectly predicted as class 1.

The confusion matrix provides full information on the predictions made by the model:There are 50 cases of class 0 and 9 cases of class 1 represented by the values along the main diagonal, which show the correctly defined examples. All of the values off the diagonal are zeros, indicating there are no incorrect classifications in the predictions.

Classification matrix:

precision, recall, and F1-score are all perfect (1.00) for both classes, indicating an excellent performance in terms of classification. The support column shows the number of actual occurrences of each class.

Recall means out of total 0's how many are correctly predicted 100% were correctly predicted in class 0 and class 1.

F1=2*Precision×Recall %Precision+Recall

precision out of total predictions how many are correct predictions, predicted 100% were correctly predicted as class 1 and class 0.

The precision, recall, and F1-score for both classes are perfect, with an overall accuracy of 100%. This indicates an excellent performance of the Naive Bayes model in distinguishing between the two classes.

Accuracy: 1.00 The ratio of correctly predicted instances to the total instances. Macro Average: Precision: 1.00 (average precision across classes) Recall: 1.00 (average recall across classes) F1-Score: 1.00 (average F1-score across classes) Weighted Average: Precision: 1.00 (weighted by the number of instances in each class) Recall: 1.00 (weighted by the number of instances in each class) F1-Score: 1.00 (weighted by the number of instances in each class)

AUC:

AUC provides a measure of how properly the model can distinguish between the two classes. The AUC score in this case is 1.0, which indicates excellent class discrimination between 2 classes.

In conclusion, the Naive Bayes classification model performed exceptionally well on the given dataset, achieving perfect precision, recall, F1-score, AUC score, and overall accuracy.

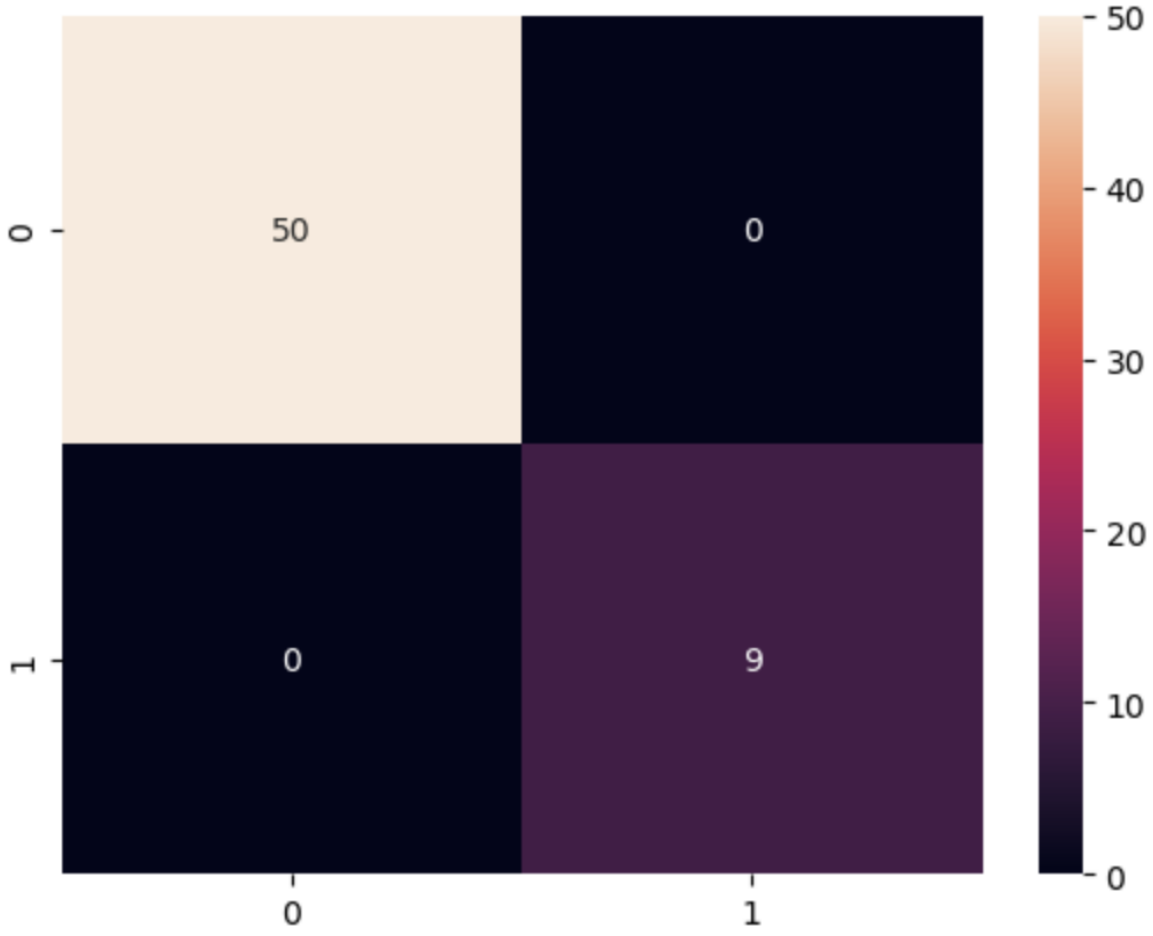
With the given dataset, the Naive Bayes model performed very well, achieving perfect precision, recall, and F1-score for both classes. The model's robustness is further supported by the AUC score of 1.0. The results suggest the Naive Bayes model is an excellent match for this classification task.

```
Confusion Matrix:
 [[50  0]
 [ 0  9]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        50
           1       1.00      1.00      1.00         9

    accuracy                           1.00        59
   macro avg       1.00      1.00      1.00        59
weighted avg       1.00      1.00      1.00        59

AUC Score:
 1.0
Accuracy of the model :
 1.0
```



3.3 Use KNN to classify your data. First find the optimal k and then run your classification. Print/report your confusion matrix, classification report and AUC:

Identifying binary outcomes, specifically the classes 0 and 1, is the model's goal. Grid search and cross-validation were used to determine 2 as the perfect amount of k.

Confusion matrix:

The confusion matrix shows that the model correctly predicted all instances of both classes. There were 50 true positive , 8 False positive and 1 true negative, indicating a good classification.

True Positive (TP): 50 instances were correctly predicted as class 0.

False Negative (FN): 8 instances that actually belong to class 1 were incorrectly predicted as class 0.

True Negative (TN): 1 instance was correctly predicted as class 1.

False Positive (FP): 0 instances that actually belong to class 0 were incorrectly predicted as class 1.

The confusion matrix provides full information on the predictions made by the model:There are 50 cases of true negatives ,8 False negative and 1 true positive, The actual value is 1 but the predicted value is 0 in 8 cases.

Classification Report:

Precision is 0.86, recall is 1 and F1-score is 0.93 for class 0, indicating a good in terms of classification, Recall means out of total 0's how many are correctly predicted 100% were correctly predicted as class 0.

The precision is 0.86 it means out of total 0 predictions how many are correct out of 58 0's 50 predictions are correct. The Precision is true positive divided by true positive+false positive.

F1=2*Precision×Recall %Precision+Recall

Precision is 1, recall is 0.11 and F1-score is 0.20 for class 1 , The recall is low in the class 1, The recall is 0.11 it means out of total 1's how many are correctly predicted in the total 9 ones only one predicted correctly, The recall is true positive divided by true positive+false negative, precision out of total predictions how many are correct predictions, predicted 100% were correctly predicted as class 1.

Accuracy: 0.86 The ratio of correctly predicted instances to the total instances. Macro Average: Precision: 0.93 (average precision across classes) Recall: 0.56 (average recall across classes) F1-Score: 0.56 (average F1-score across classes) Weighted Average: Precision: 0.88 (weighted by the number of instances in each class) Recall: 0.86 (weighted by the number of instances in each class) F1-Score: 0.82 (weighted by the number of instances in each class)

AUC:

The area under the ROC curve (AUC) provides a measure of the model's ability to distinguish between the two classes. In this case, the AUC score is 0.7155, indicating a moderate ability to discriminate between classes.

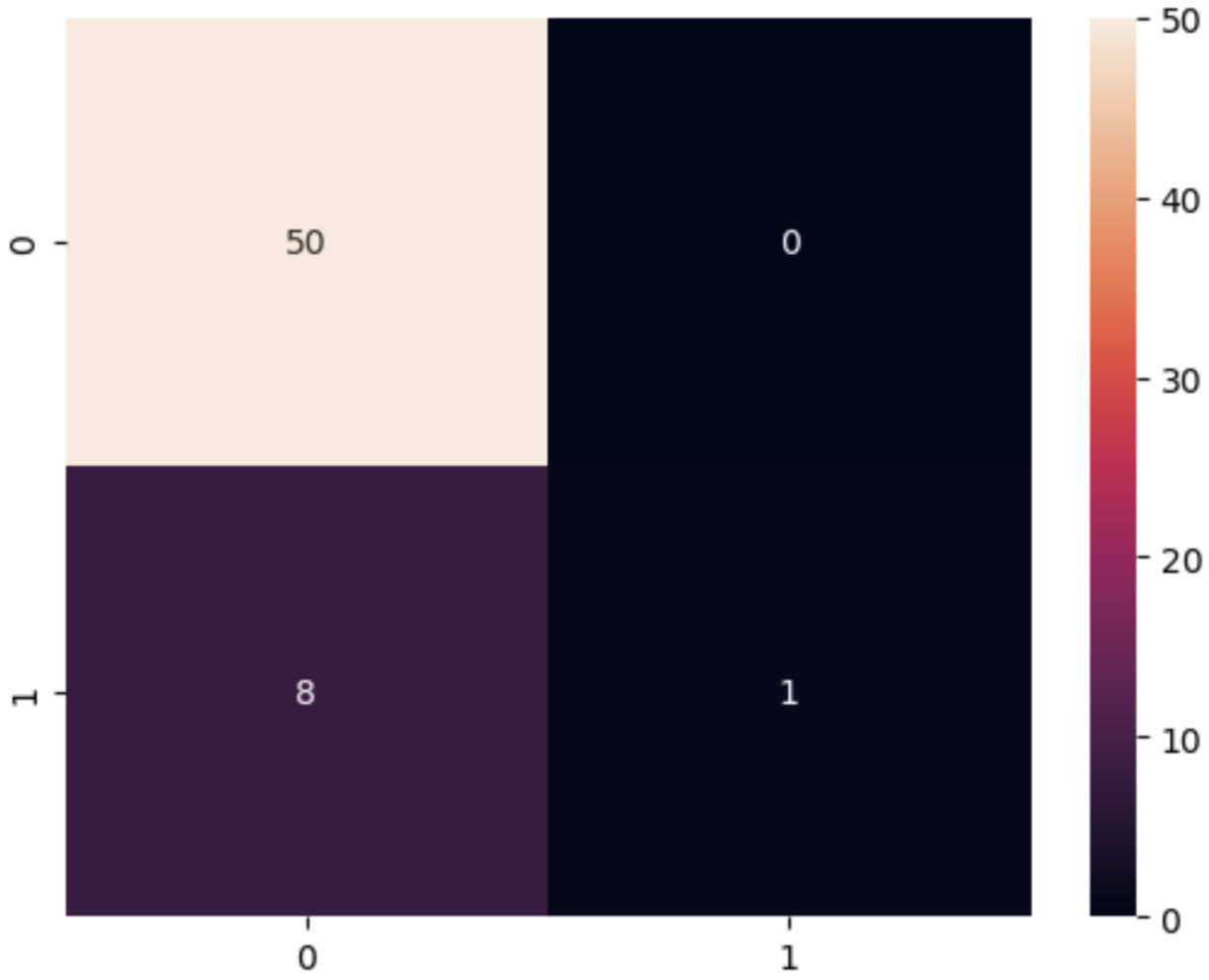
Based on the provided dataset, the K-Nearest Neighbours model shows good performance, with an optimal value of k=2. The model performs poorly on class 1, as seen by the lower recall and F1-score, even if it shows excellent precision and recall for class 0. A moderate amount of class discrimination is given based on the AUC score and overall accuracy is 86%.

```
Optimal k: 2
Confusion Matrix:
 [[50  0]
 [ 8  1]]
Classification Report:
              precision    recall  f1-score   support

           0       0.86      1.00      0.93        50
           1       1.00      0.11      0.20         9

    accuracy                           0.86        59
   macro avg       0.93      0.56      0.56        59
weighted avg       0.88      0.86      0.82        59

AUC Score: 0.7155555555555555
```



3.4 Choose one: SVM or Random Forest to classify your data. Print/report your confusion matrix, classification report and AUC:

Confusion Matrix:

The confusion matrix shows that the model correctly predicted all instances of both classes. There were 50 true positives and 9 false negatives, indicating a good classification.

True Positive (TP): 50 instances were correctly predicted as class 0.

False Negative (FN): 9 instances that actually belong to class 1 were incorrectly predicted as class 0.

True Negative (TN): 0 instances were correctly predicted as class 1.

False Positive (FP): 0 instances that actually belong to class 0 were incorrectly predicted as class 1.

Classification Report:

Precision is 0.85, recall is 1 and F1-score is 0.92 for 0 class indicating a good in terms of classification, Recall means out of total 0's how many are correctly predicted 100% were correctly predicted as class 0.

The precision is 0.85 it means out of total 0 predictions how many are correct out of 59 0's 50 predictions are correct. The Precision is true positive divided by true positive+false positive.

F1=2*Precision×Recall %Precision+Recall

Precision is 0, recall is 0 and F1-score is 0 for 1 class , The recall is zero in the class 1, The recall is 0 it means out of total 1's how many are correctly predicted in the total 9 ones zero predicted correctly, The recall is true positive divided by true

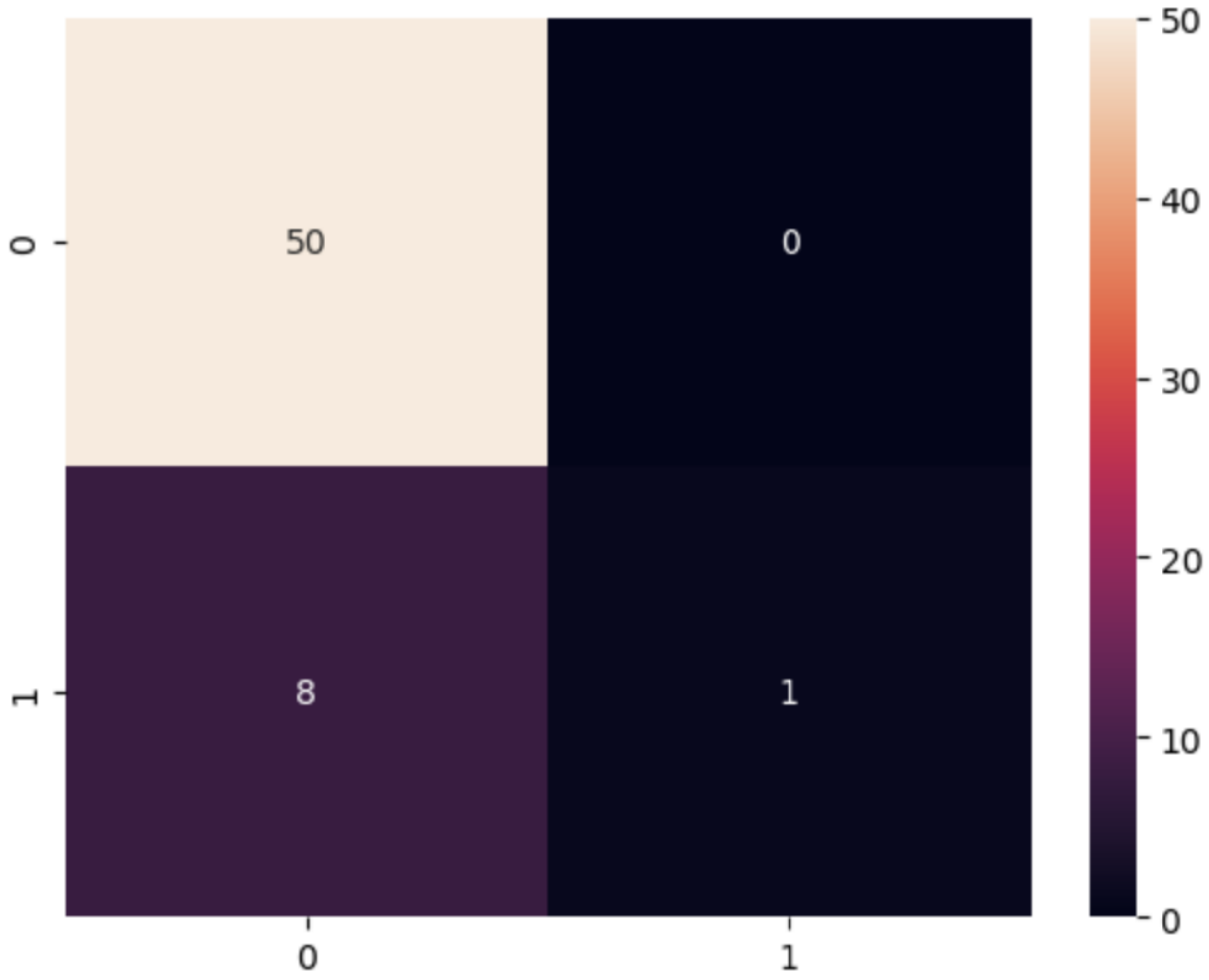
positive+false negative, precision is out of total predictions how many are correct predictions, out of 9 predictions no one is correct.

Accuracy: 0.85 The ratio of correctly predicted instances to the total instances. Macro Average: Precision: 0.42 (average precision across classes) Recall: 0.50 (average recall across classes) F1-Score: 0.46 (average F1-score across classes) Weighted Average: Precision: 0.72 (weighted by the number of instances in each class) Recall: 0.85 (weighted by the number of instances in each class) F1-Score: 0.78 (weighted by the number of instances in each class) .

Auc:

The area under the ROC curve (AUC) provides a measure of the model's ability to distinguish between the two classes. In this case, the AUC score is 0.85, indicating a moderate ability to discriminate between classes.

The model performs well in predicting instances of Class 0, as indicated by high precision, recall, and F1-score. However, for Class 1, the model struggles, as evidenced by the low precision, recall, and F1-score. The overall accuracy is 85%, but the model's performance is imbalanced due to the challenges in predicting Class 1 instances


50
0
8
1
0
1
0
1
50
40
30
20
10
0

3.5 Compare your results and comment on your findings. Which one(s) did the best job? What could have been the problem with the ones that did not work? Etc.

1. Logistic Regression:

Performance: Perfect precision, recall, and F1-Score for both classes.

AUC Score:1.0

Accuracy:1.0

Conclusion: Logistic Regression performed exceptionally well, achieving perfect classification on the provided dataset.

2.Naive Bayes:

Performance:Perfect precision, recall, and F1-Score for both classes.

AUC Score:1.0

Accuracy:1.0

Conclusion:Naive Bayes also performed exceptionally well, achieving perfect classification on the provided dataset.

3.K-Nearest Neighbors (KNN):

Performance: Good accuracy in class 0, but struggled with predicting instances of Class 1, as reflected in low recall and F1-Score for Class 1.

AUC Score: 0.715

Conclusion: KNN performed reasonably well, but its struggle with imbalanced classes suggests the need for further optimization or exploration of alternative algorithms.

4.Support Vector Machines (SVM):

Performance:Good accuracy for Class 0, but poor performance for Class 1 (low precision, recall, and F1-Score).

AUC Score: 0.855

Conclusion: SVM achieved decent overall accuracy, especially for Class 0. However, its inability to correctly predict instances of Class 1 indicates a challenge in handling imbalanced classes.

Comparison and Findings:

Logistic Regression and Naive Bayes outperformed the other models, achieving perfect classification on the given dataset.

KNN showed reasonable accuracy but struggled with imbalanced classes, resulting in lower performance metrics for Class 1.

SVM achieved good accuracy for Class 0 but struggled significantly with Class 1, indicating challenges in handling imbalanced classes.

Possible Issues:

Class 1 cases are challenging for both KNN and SVM to predict, which could be due to an imbalanced class distribution. The model could accord preference to the

majority class if one dominates the other, this would result in poor outcomes on the minority class.

The imbalanced nature of the dataset may have affected the performance of KNN and SVM, especially in predicting the class 1.

Further hyperparameter tuning might better KNN performance, especially when identifying the ideal value for k. The capacity of the model for generalization is greatly affected by the selection of k.

The choice of hyperparameters in KNN, such as the number of neighbors (k), could impact its performance.

Performance of SVM may be affected by the kernel it selects. Using non-linear kernels or trying with different kernels could increase the ability to determine complexity connections in the data.

4 Bonus question:

1.Confusion Matrix: The new KNN model's confusion matrix shows an improvement in predicting instances of Class 1 (True Positives increased from 1 to 6), 2.Classification Report: Precision for Class 1:Improved from 1.00 to 0.38, indicating a reduction in false positives. Recall for Class 1:Improved from 0.11 to 0.67, indicating a significant increase in the ability to identify true positives. F1-Score for Class 1:Improved from 0.20 to 0.48, indicating a better balance between precision and recall.

3.Overall Metrics: Accuracy:Decreased from 0.86 to 0.78, mainly due to the increase in false positives for Class 1. Macro Average: Precision: Improved slightly

from 0.93 to 0.65. Recall: Improved from 0.56 to 0.73. F1-Score:Improved from 0.56 to 0.67. Weighted Average: Precision:Improved slightly from 0.88 to 0.85. Recall:Decreased from 0.86 to 0.78. F1-Score:Improved from 0.82 to 0.80.

4.AUC Score: The AUC score improved from 0.715 to 0.841, indicating enhanced discriminative ability.

Conclusion: If compared to the old model, the new KNN model, which features an optimal k=2, performs more effectively in identifying positive cases (Class 1). The greater AUC score and F1-Score for Class 1 suggest the modifications have improved the balance between precision and recall. On the other hand, a decrease in overall accuracy indicates an agreement in wrongly classifying negative cases (Class 0). When evaluating the model's performance, it is essential that you take into account its unique aims and priorities.