Instructions

1. A supervised learning model has been built to predict whether someone is infected with a new strain of a virus. The probability of any one person having the virus is 5%. Using accuracy as a metric, what would be a good choice for a baseline accuracy score that the new model would want to outperform? 1 point

0.95

2. Given the following confusion matrix: 1 point

	Predicted Positive	Predicted Negative
Condition Positive	67	11
Condition Negative	3	8

Compute the accuracy to three decimal places.

0.843

3. Given the following confusion matrix: 1 point

	Predicted Positive	Predicted Negative
Condition Positive	102	56
Condition Negative	17	78

Compute the precision to three decimal places.

0.857

4. Given the following confusion matrix: 1 point

	Predicted Positive	Predicted Negative
Condition Positive	102	56
Condition Negative	17	78

Compute the recall to three decimal places.

0.646

5. Using the fitted model `m` create a precision-recall curve to answer the following question: 1 point

For the fitted model `m`, approximately what precision can we expect for a recall of 0.8?

(Use y_test and X_test to compute the precision-recall curve. If you wish to view a plot, you can use `plt.show()`)

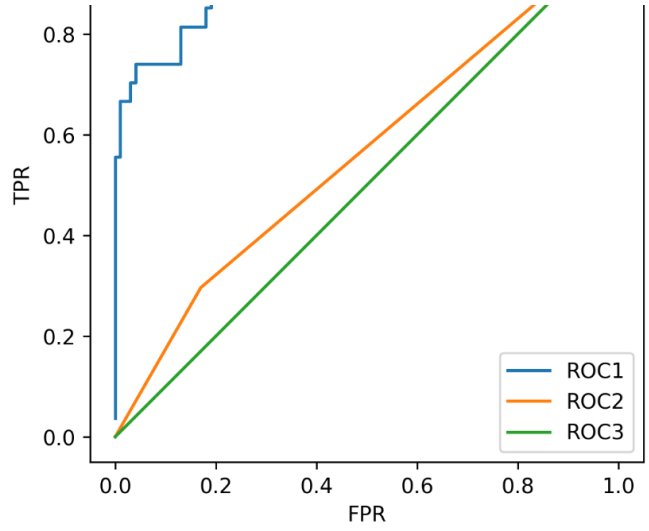
```
# Run this in your notebook
y_scores = m.decision_function(X_test)
from sklearn.metrics import precision_recall_curve
import numpy as np

precision, recall, thresholds = precision_recall_curve(y_test, y_scores)
# Find closest recall to 0.8
closest_zero = np.argmin(np.abs(recall - 0.8))
print(precision[closest_zero])
```

Run
Reset

0.55

6. Given the following models and AUC scores, match each model to its corresponding ROC curve. 1 point

- Model 1 test set AUC score: 0.91
- Model 2 test set AUC score: 0.50
- Model 3 test set AUC score: 0.56



- Model 1: Roc 1
- Model 2: Roc 2
- Model 3: Roc 3

- Model 1: Roc 1
- Model 2: Roc 3
- Model 3: Roc 2

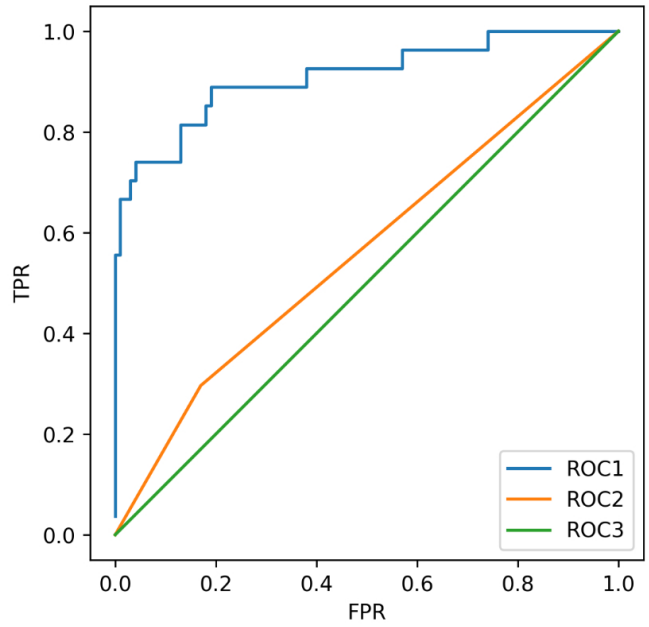
- Model 1: Roc 2
- Model 2: Roc 3
- Model 3: Roc 1

- Model 1: Roc 3
- Model 2: Roc 2
- Model 3: Roc 1

Not enough information is given.

7. Given the following models and accuracy scores, match each model to its corresponding ROC curve. 1 point

- Model 1 test set accuracy: 0.91
- Model 2 test set accuracy: 0.79
- Model 3 test set accuracy: 0.72



- Model 1: Roc 1
- Model 2: Roc 2
- Model 3: Roc 3

- Model 1: Roc 1
- Model 2: Roc 3
- Model 3: Roc 2

- Model 1: Roc 2
- Model 2: Roc 3
- Model 3: Roc 1

- Model 1: Roc 3
- Model 2: Roc 2
- Model 3: Roc 1

Not enough information is given.

8. Using the fitted model `m` what is the macro precision score? 1 point

(Use y_test and X_test to compute the precision score.)

```
from sklearn.metrics import precision_score
y_pred = m.predict(X_test)
print(precision_score(y_test, y_pred, average='micro'))
```

Run Reset

```
0.752
```

9. Which of the following is true of the R-Squared regression score metric? (Select all that apply) 1 point

- [ ] A model that always predicts the mean of y would get a score of 0.5
- [x] The score can sometimes be negative.
- [ ] A model that always predicts the mean of y would get a score of 0.0
- [x] The highest possible score is 1.0

10. In a future society, a video surveillance algorithm is used to predict a crime before it occurs. If you were responsible for tuning this machine, what evaluation metric would you want to maximize to ensure no innocent people (people not about to commit a crime) are imprisoned (where crime is the positive label)? 1 point

- Accuracy
- Precision
- Recall
- F1
- AUC

11. Consider the algorithm from the previous question. If you were responsible for tuning this machine, what evaluation metric would you want to maximize to ensure all criminals (people about to commit a crime) are imprisoned (where crime is the positive label)? 1 point

- Accuracy
- Precision
- Recall
- F1
- AUC

12. A classifier is trained on an imbalanced multiclass dataset, where there is one 'frequent' majority class that represents 80% of the labeled data, and ten rare classes that together represent the remaining 20% of 'infrequent' labels. After looking at the model's precision scores, you find that the micro-averaged precision is much larger than the *macro-averaged* precision score. Which of the following is most likely happening? 1 point

- The model is probably misclassifying the infrequent labels more than the frequent labels.
- The model is probably misclassifying the frequent labels more than the infrequent labels.

13. Using the already defined RBF SVC model `m`, run a grid search on the parameters C and gamma, for values [0.01, 0.1, 1, 10]. The grid search should find the model that best optimizes for recall. How much better is the recall of this model than the precision? (Compute recall - precision to 3 decimal places) 1 point

(Use y_test and X_test to compute precision and recall.)

```
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import precision_score, recall_score

grid_values = {'C': [0.01, 0.1, 1, 10], 'gamma': [0.01, 0.1, 1, 10]}
grid_clf_rec = GridSearchCV(m, param_grid=grid_values, scoring='recall')
grid_clf_rec.fit(X_train, y_train)
y_pred = grid_clf_rec.predict(X_test)

# Calculate recall - precision
print(recall_score(y_test, y_pred) - precision_score(y_test, y_pred))
```

Run

Reset

0.57

14. Using the already defined RBF SVC model `m`, run a grid search on the parameters C and gamma, for values [0.01, 0.1, 1, 10]. The grid search should find the model that best optimizes for precision. How much better is the precision of this model than the recall? (Compute precision - recall to 3 decimal places)

1 point

(Use y_test and X_test to compute precision and recall.)

```
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import precision_score, recall_score

grid_values = {'C': [0.01, 0.1, 1, 10], 'gamma': [0.01, 0.1, 1, 10]}
grid_clf_prec = GridSearchCV(m, param_grid=grid_values, scoring='precision')
grid_clf_prec.fit(X_train, y_train)
y_pred = grid_clf_prec.predict(X_test)

# Calculate precision - recall
print(precision_score(y_test, y_pred) - recall_score(y_test, y_pred))
```

Run

Reset

0.15