Module 3 Quiz

TOTAL POINTS 14

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 1%. 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

0.99

2. Given the following confusion matrix:

1 point

	Predicted Positive	Predicted Negative
Condition Positive	96	4
Condition Negative	8	19

0.905

3. Given the following confusion matrix:

1 point

	Predicted Positive	Predicted Negative
Condition Positive	96	4
Condition Negative	8	19

Compute the precision to three decimal places.

0.923

4. Given the following confusion matrix:

1 point

	Predicted Positive	Predicted Negative
Condition Positive	96	4
Condition Negative	8	19

Compute the recall to three decimal places.

0.960

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

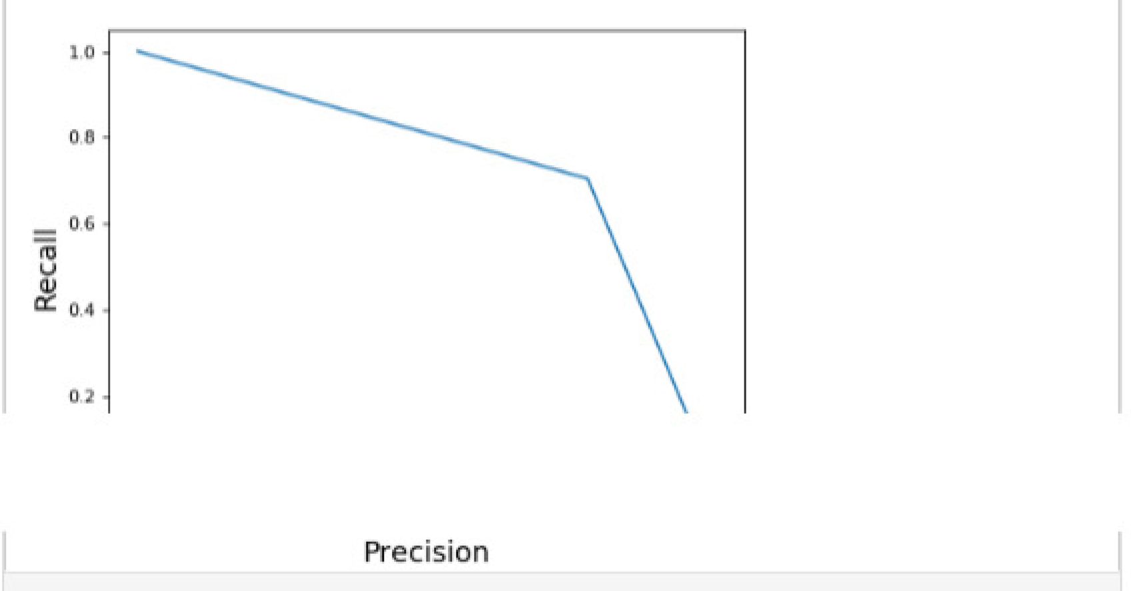
1 point

plt.show()

```
from sklearn.metrics import precision_recall_curve
preds = m.predict(X_test)
precision, recall, thresholds = precision_recall_curve(y_test, preds)
plt.plot(precision,recall)
plt.xlabel('Precision', fontsize=16)
plt.ylabel('Recall', fontsize=16)
plt.show()

#print(m)
```

Run
Reset

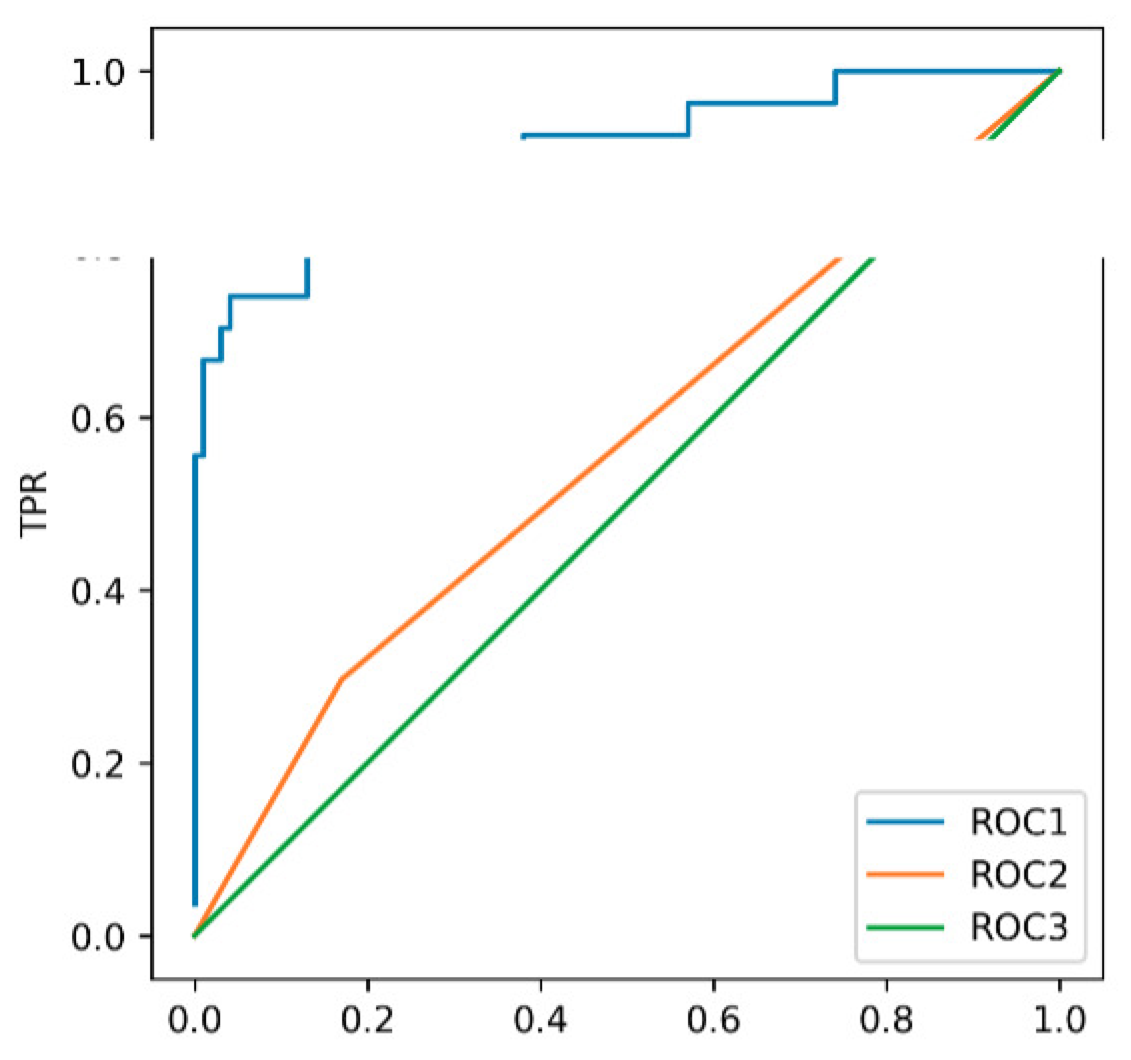


0.6

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

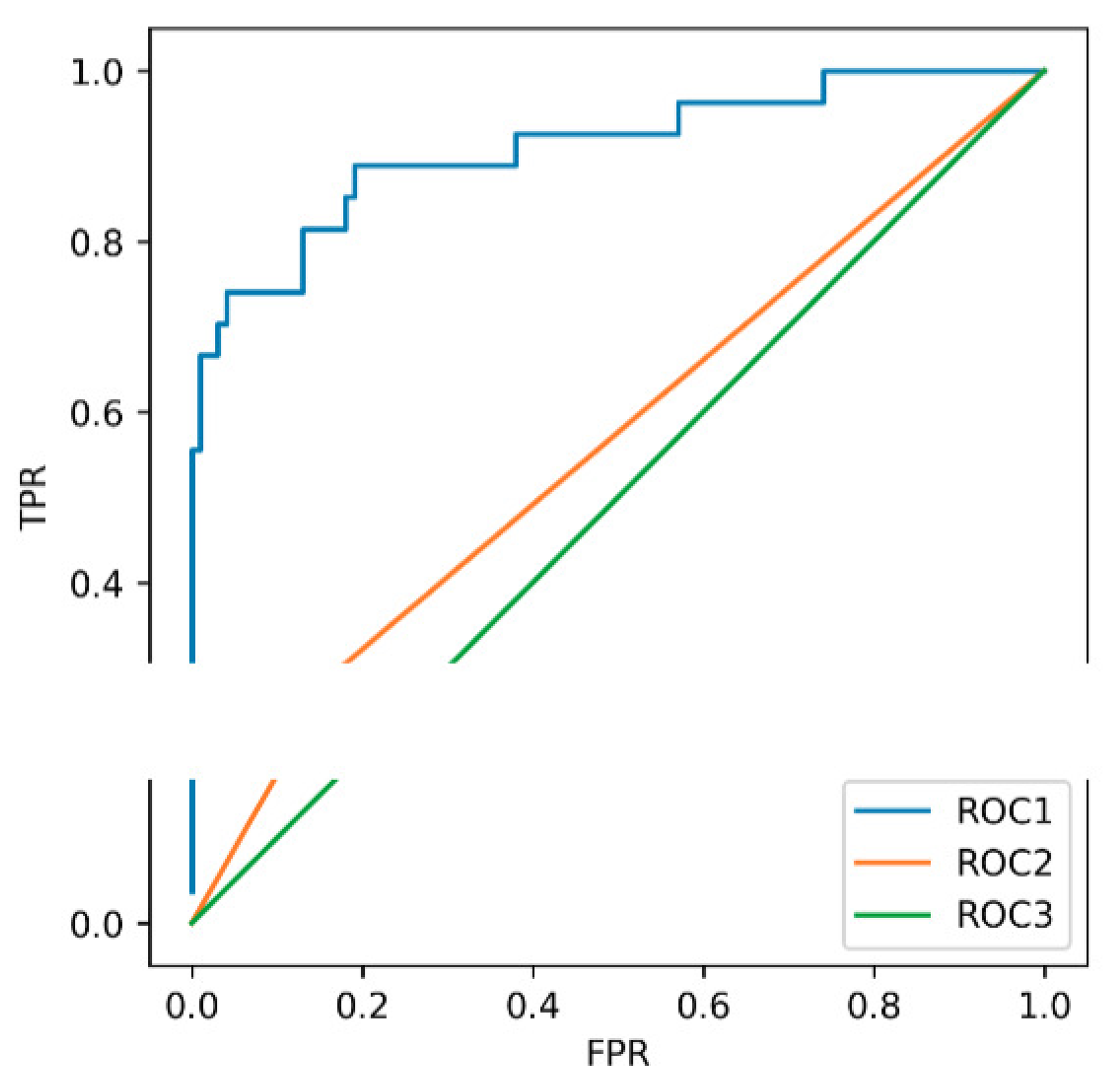
1 point

- 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.

- 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 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.)

```
pred = m.predict(X_test)
print(precision_score(y_test, pred, average = 'macro'))
```

Run
Reset

```
0.805008635579
0.805008635579
None
```

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

1 point

- [ ] The worst possible score is 0.0
- [ ] A model that always predicts the mean of y would get a negative score
- [x] The best possible score is 1.0
- [x] A model that always predicts the mean of y would get a score of 0.0

10. In a future society, a machine 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
- ○ AUC

11. Consider the machine 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. After looking at the model's precision scores, you find that the micro averaging is much smaller than the macro averaging score. Which of the following is most likely happening?

1 point

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.)

```
grid_values = {'C': [0.01, 0.1, 1, 10] , 'gamma': [0.01, 0.1, 1, 10] }
grid_recall = GridSearchCV(m, param_grid = grid_values, scoring='recall')
grid_recall.fit(X_train, y_train)
pred = grid_recall.best_estimator_.predict(X_test)
rec = recall_score(y_test, pred)
prec = precision_score(y_test, pred)
print(rec-prec)
```

Run
Reset

```
0.52
0.52
None
```

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.)

```
grid_values = {'C': [0.01, 0.1, 1, 10] , 'gamma': [0.01, 0.1, 1, 10] }
print(prec-rec)
```

Reset

```
0.15
0.15
None
```

0.15

- [x] I, **Ezequiel Kinigsberg**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

Learn more about Coursera's Honor Code

Save
Submit