Making Predictions

PREDICTING CUSTOMER CHURN IN PYTHON

Mark Peterson
Senior Data Scientist, Alliance Data

(Supervised) Machine Learning Primer

- Goal: Predict whether or not a customer will churn
- Target Variable: `'Churn'`
- Supervised Machine Learning
- Learn from historical (training) data to make new predictions

scikit
learn

Model Selection

- Which model to use?
- ... it depends!
- In this course: Experiment with several models
- To learn about their inner workings: Check out other DataCamp courses

Model Selection

- Logistic regression: Good baseline
    - Offers simplicity and interpretability
    - Cannot capture more complex relationships
- Random forests
- Support vector machines

Training your Model

```python
from sklearn.svm import SVC

svc = SVC()

svc.fit(telco[features], telco['target'])
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

Making a Prediction

```
prediction = svc.predict(new_customer)


print(prediction)
```

```
[0]
```

Let's practice!

PREDICTING CUSTOMER CHURN IN PYTHON

Evaluating Model Performance

PREDICTING CUSTOMER CHURN IN PYTHON

Mark Peterson
Senior Data Scientist, Alliance Data

Accuracy

- One possible metric: Accuracy
  - Total Number of Correct Predictions / Total Number of Data Points
- What data to use?
  - Training data not representative of new data

Training and Test Sets

- Fit your classifier to the training set
- Make predictions using the test set

Training and Test Sets using scikit-learn

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(telco['data'], telco['target'],
                                   test_size=0.2, random_state = 42)
from sklearn.svm import SVC

svc = SVC()

svc.fit(X_train, y_train)

svc.predict(X_test)
```

Computing Accuracy

```
svc.score(X_test, y_test)
```

```
0.857
```

- 85.7% accuracy: Quite good for a first try!

Improving your model

- Overfitting: Model fits the training data too closely
- Underfitting: Does not capture trends in the training data
- Need to find the right balance between overfitting and underfitting

Let's practice!

PREDICTING CUSTOMER CHURN IN PYTHON

Model Metrics

PREDICTING CUSTOMER CHURN IN PYTHON

Mark Peterson
Senior Data Scientist, Alliance Data

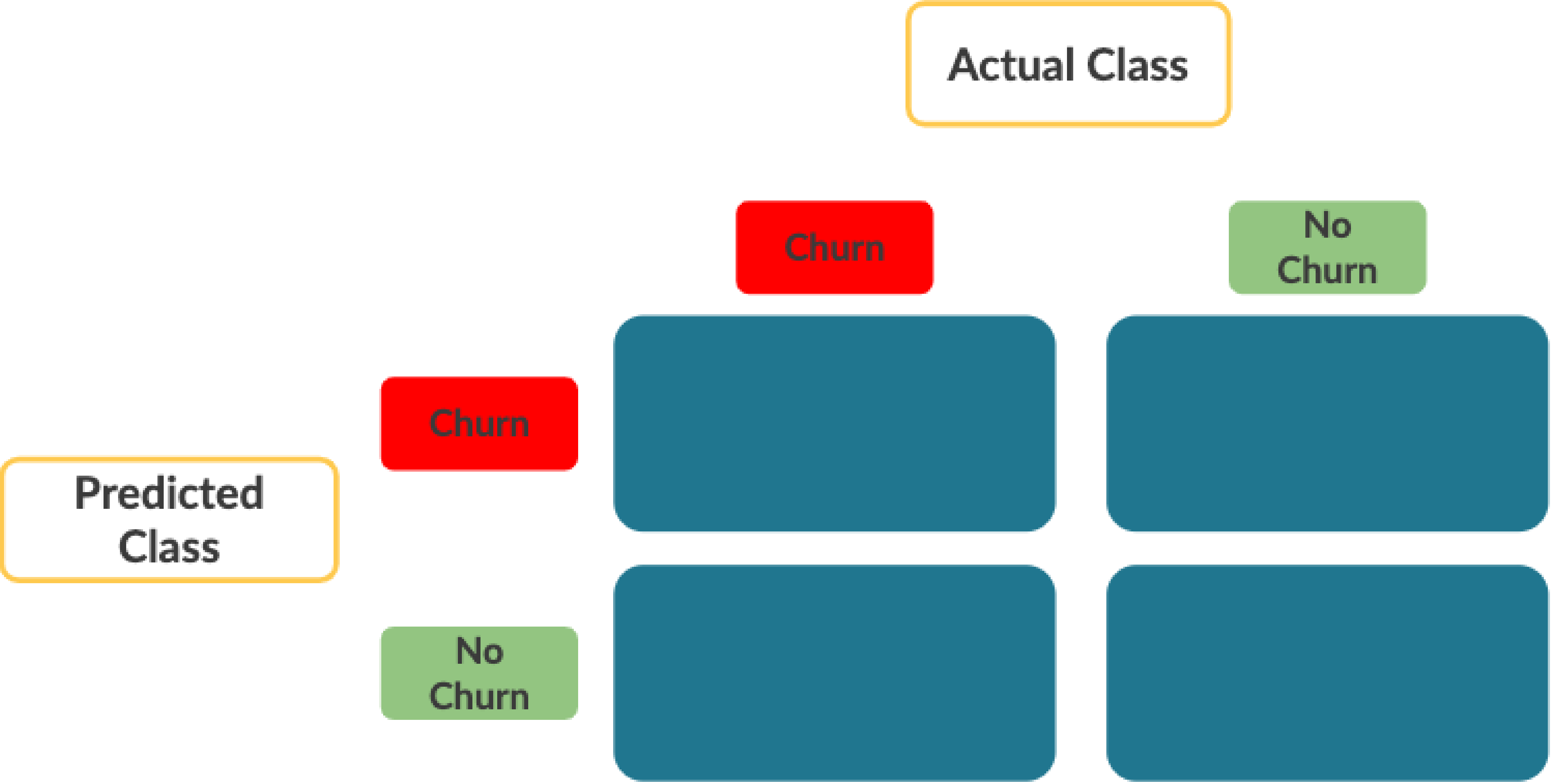
Imbalanced classes

```
telco['Churn'].value_counts()
```

```
no      2850
yes      483
Name: Churn, dtype: int64
```

- Accuracy not a very useful metric

Actual Class	
Churn	No Churn



Actual Class

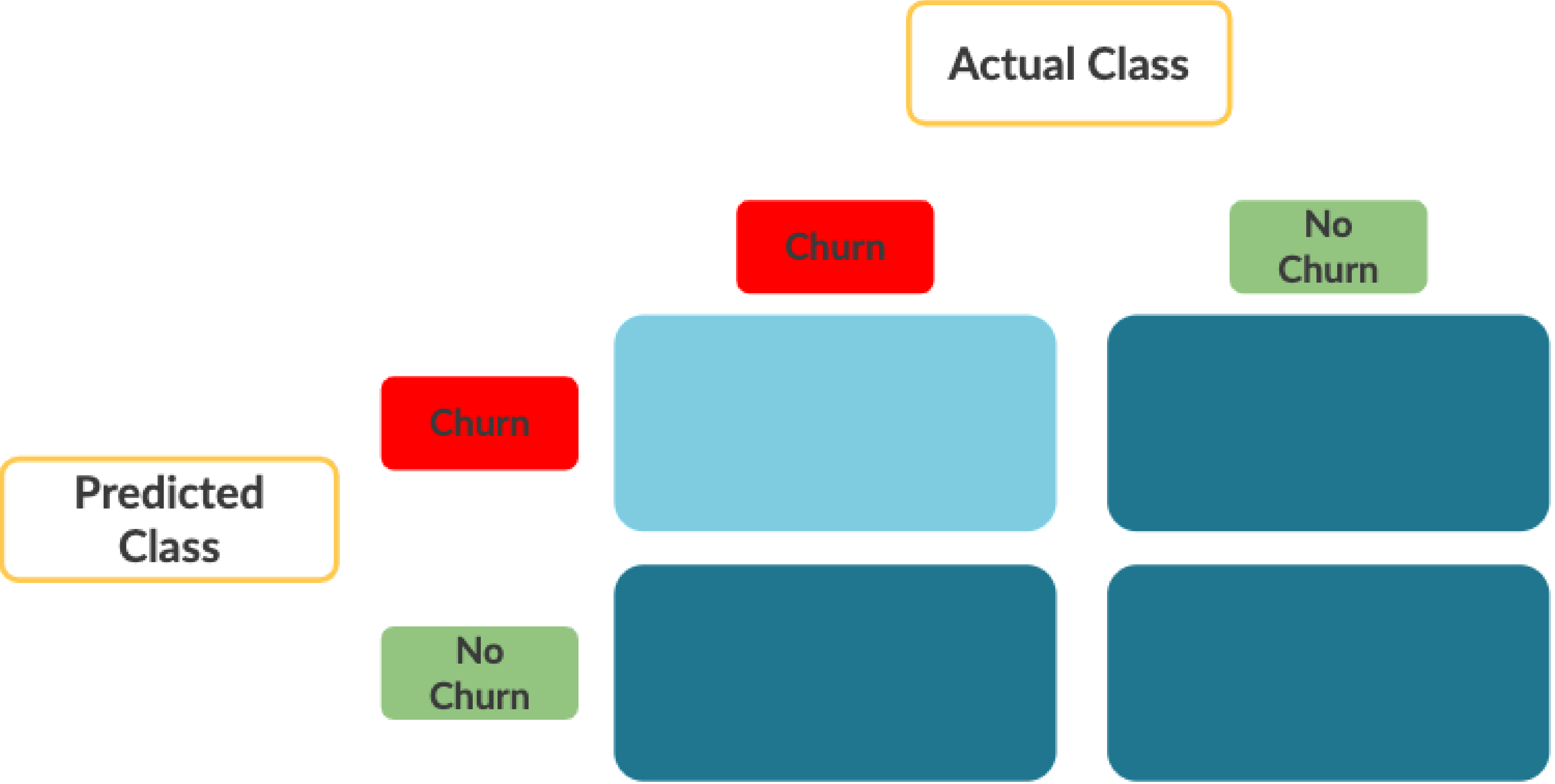
Churn

No Churn

Predicted Class

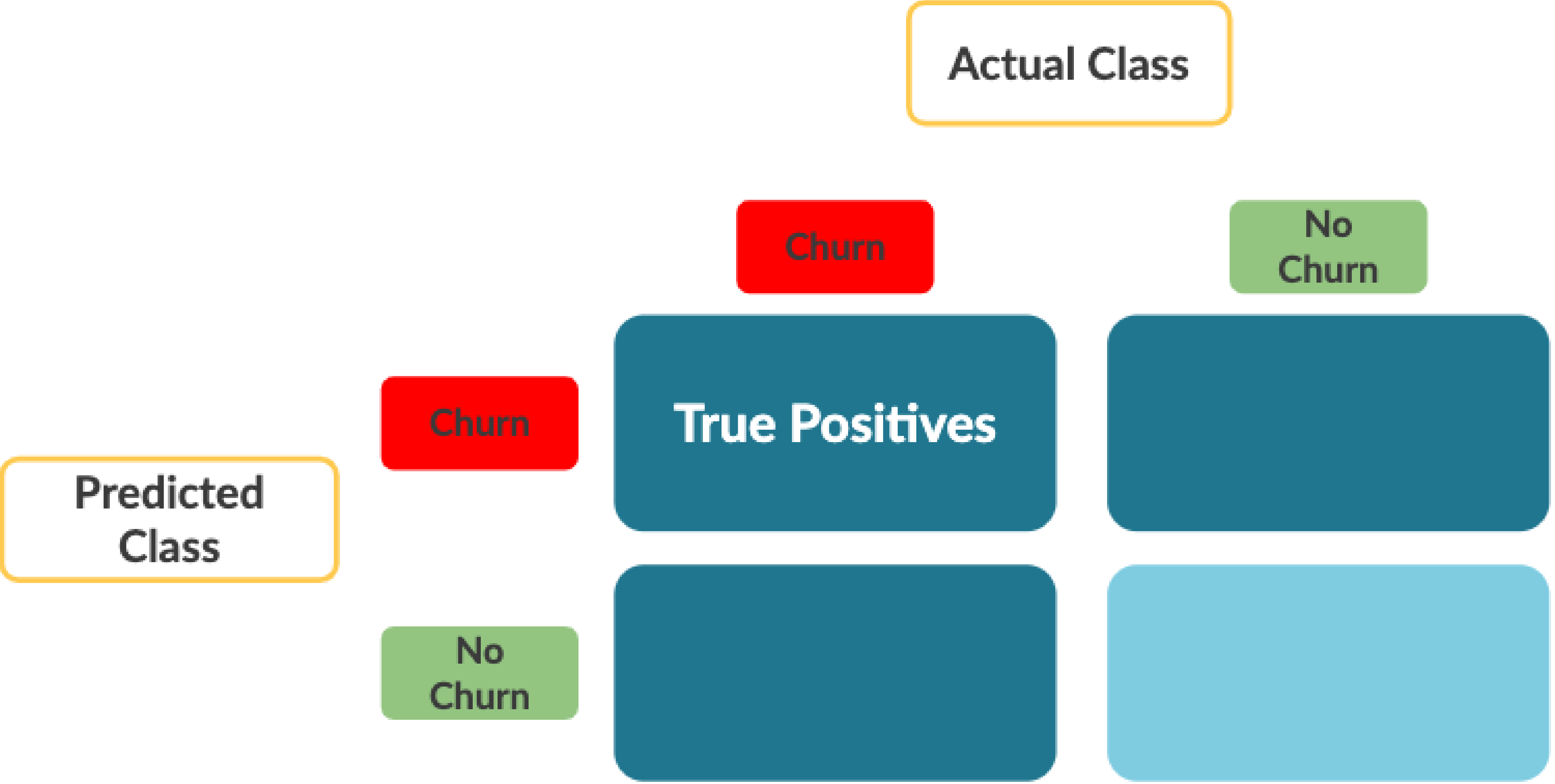
Churn

No Churn



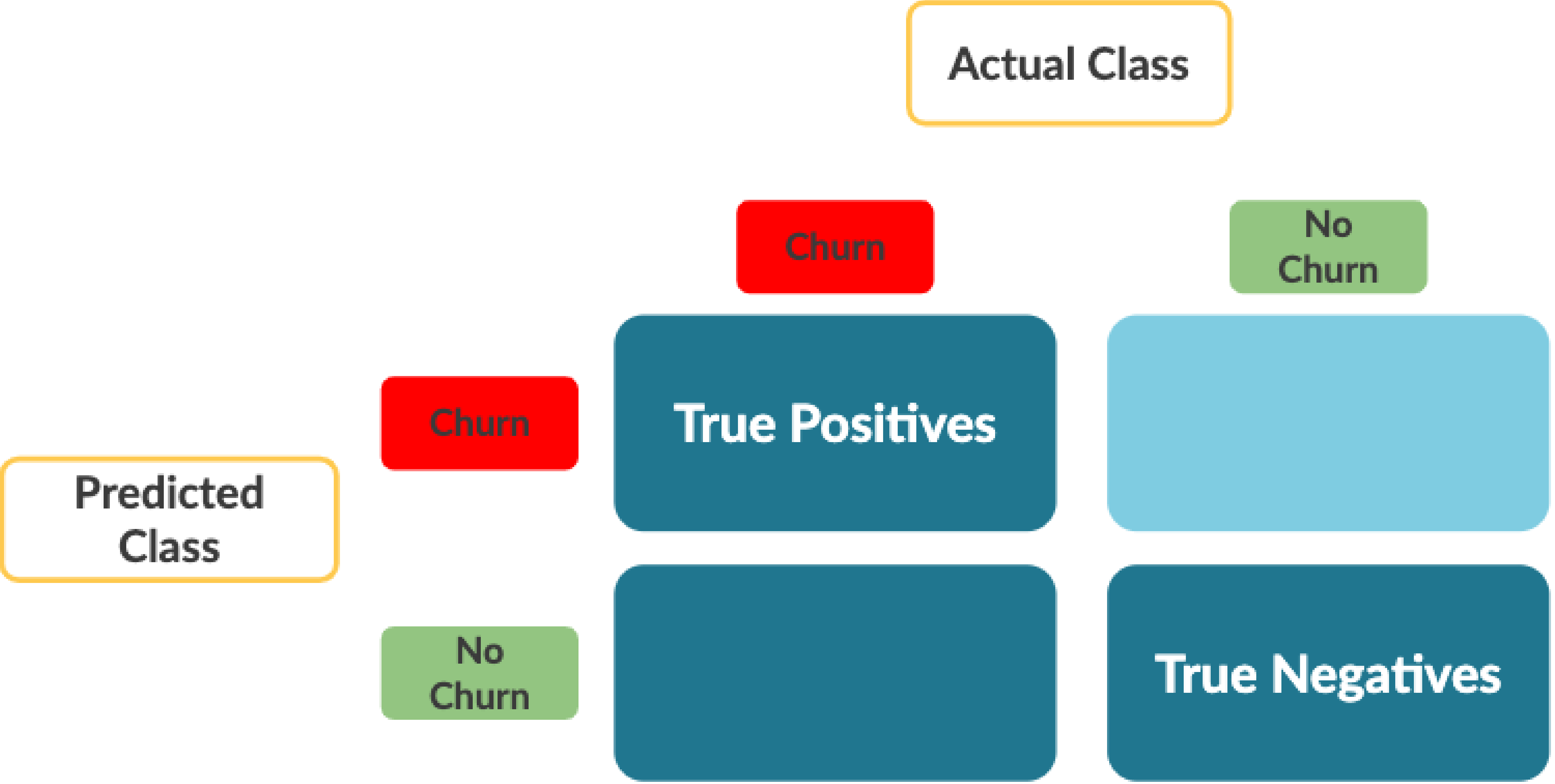
		Actual Class	
		Churn	No Churn
Predicted Class	Churn		
	No Churn		

		Actual Class	
		Churn	No Churn
Predicted Class	Churn	True Positives	
	No Churn		

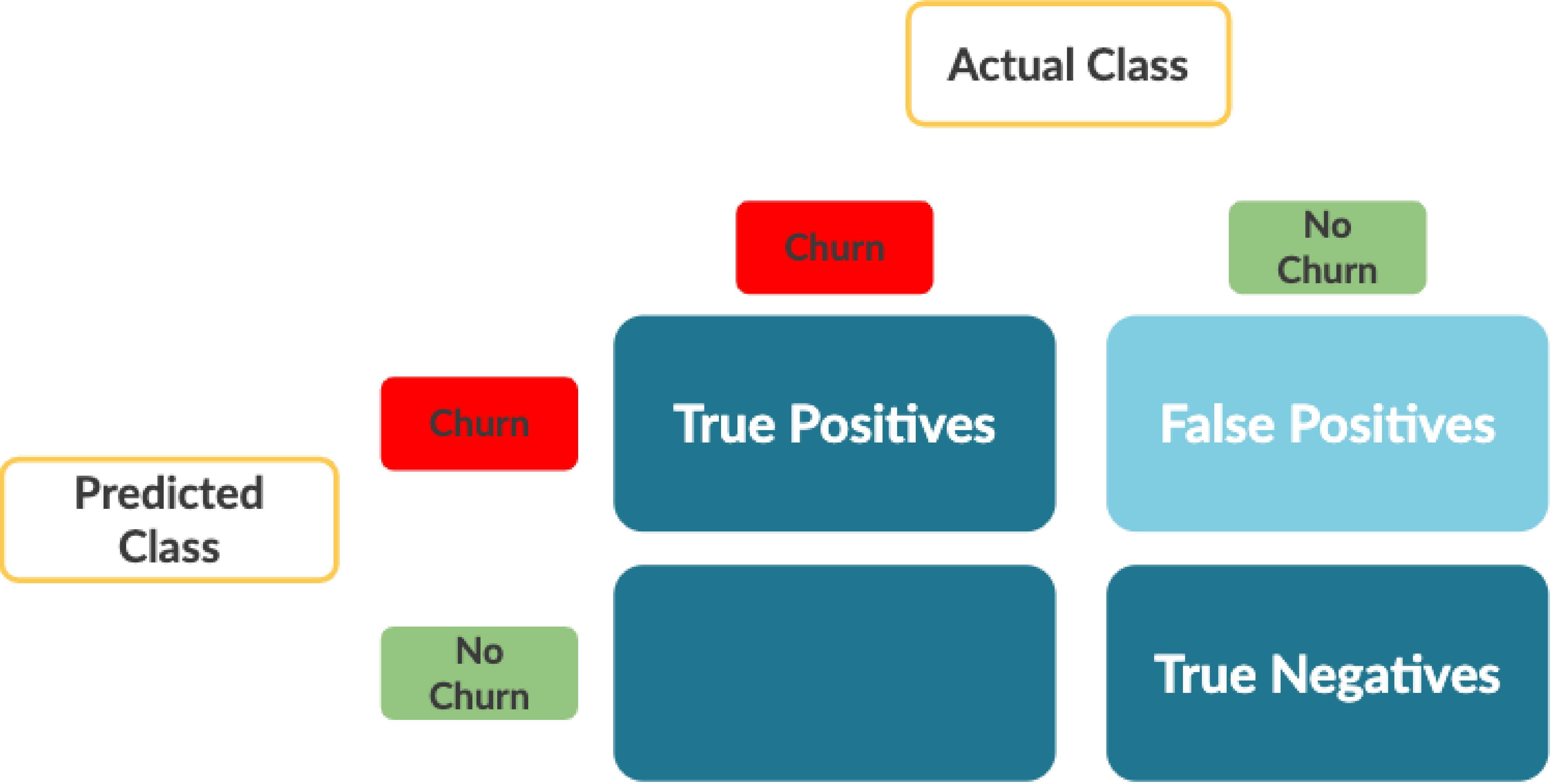


		Actual Class	
		Churn	No Churn
Predicted Class	Churn	True Positives	
	No Churn		

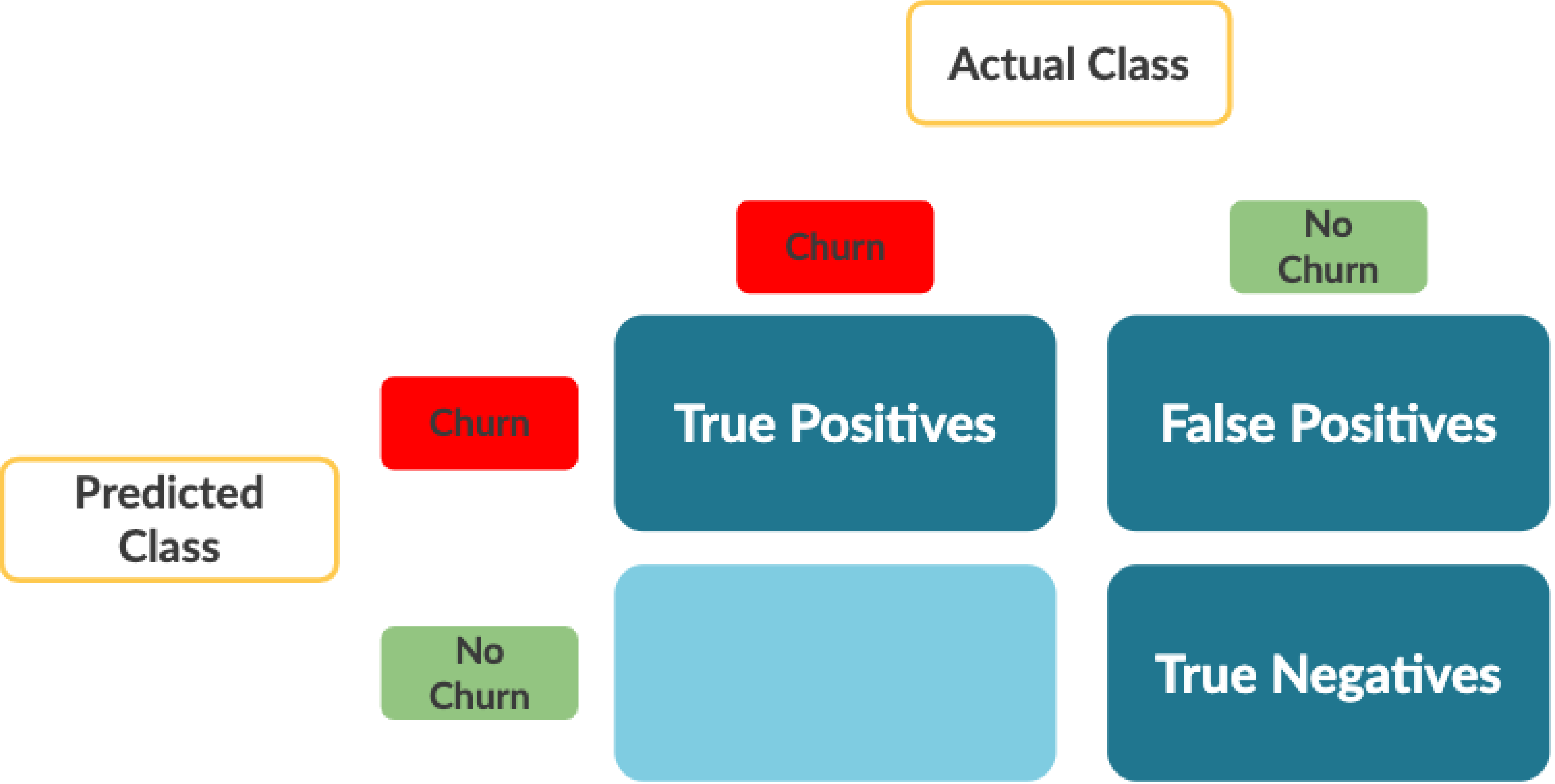
		Actual Class	
		Churn	No Churn
Predicted Class	Churn	True Positives	
	No Churn		True Negatives



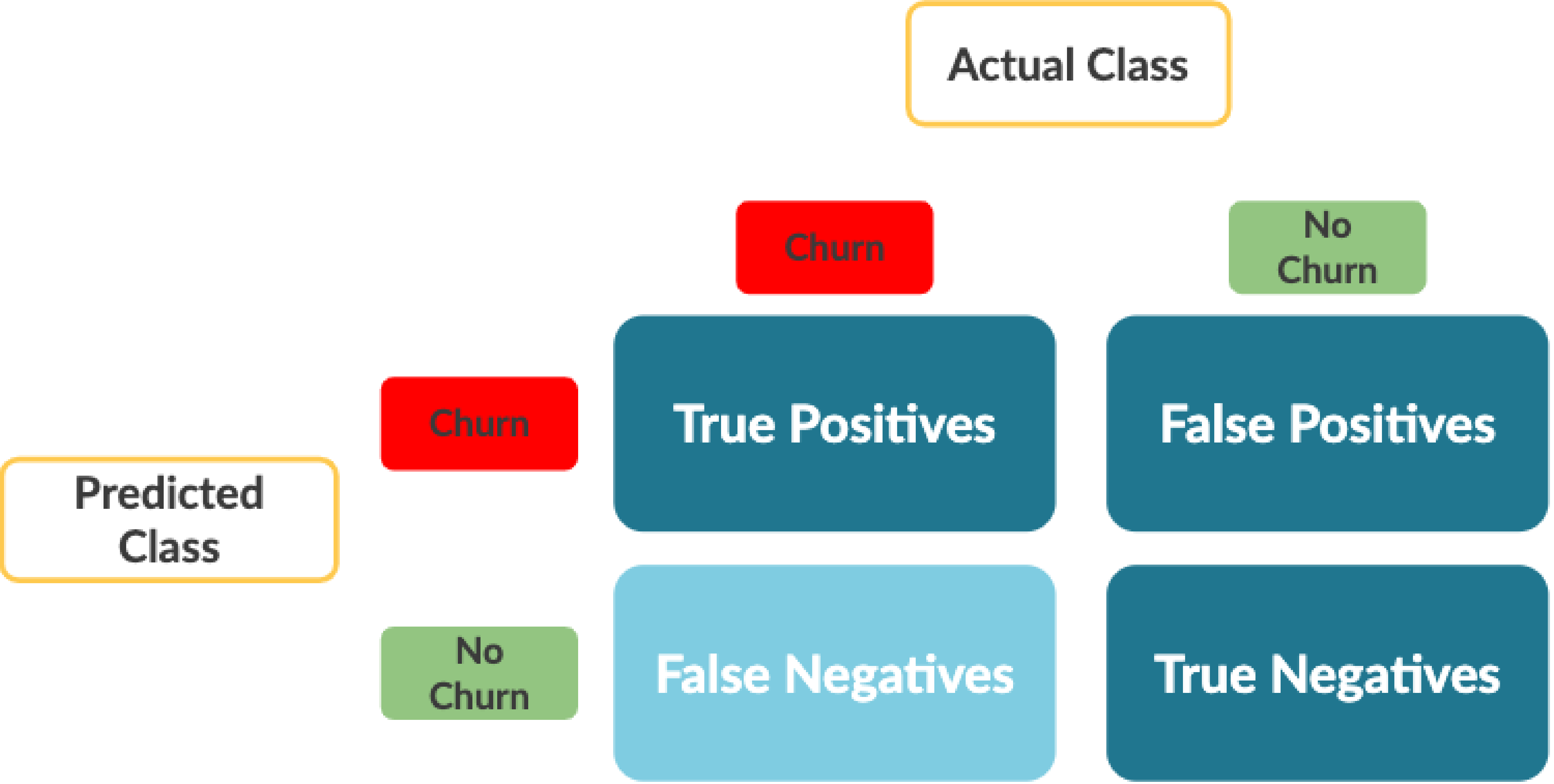
		Actual Class	
		Churn	No Churn
Predicted Class	Churn	True Positives	
	No Churn		True Negatives



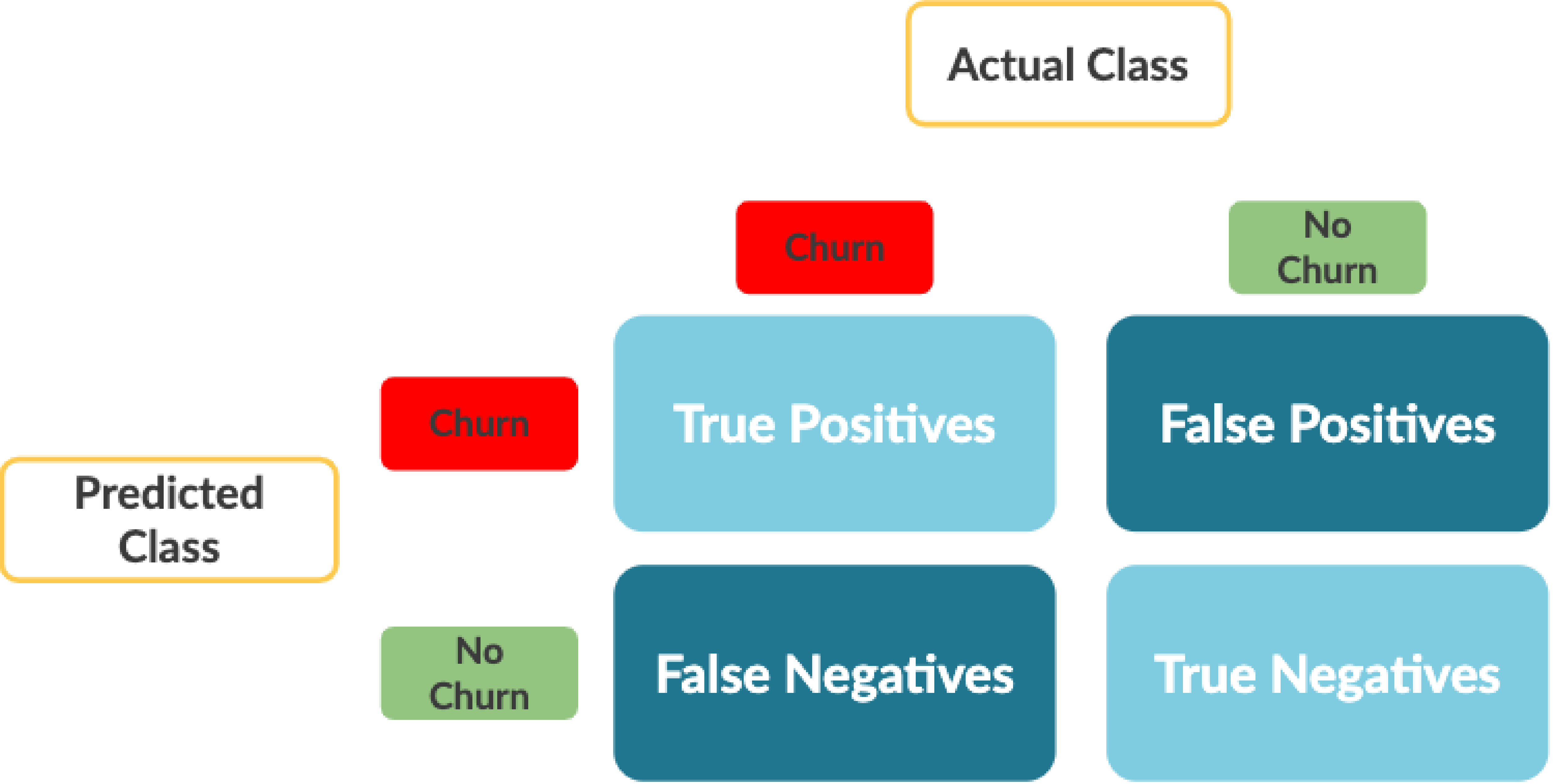
		Actual Class	
		Churn	No Churn
Predicted Class	Churn	True Positives	False Positives
	No Churn		True Negatives



		Actual Class	
		Churn	No Churn
Predicted Class	Churn	True Positives	False Positives
	No Churn		True Negatives



		Actual Class	
		Churn	No Churn
Predicted Class	Churn	True Positives	False Positives
	No Churn	False Negatives	True Negatives



		Actual Class	
		Churn	No Churn
Predicted Class	Churn	True Positives	False Positives
	No Churn	False Negatives	True Negatives

Precision

Metric	Formula
Precision	True Positives / (True Positives + False Positives)

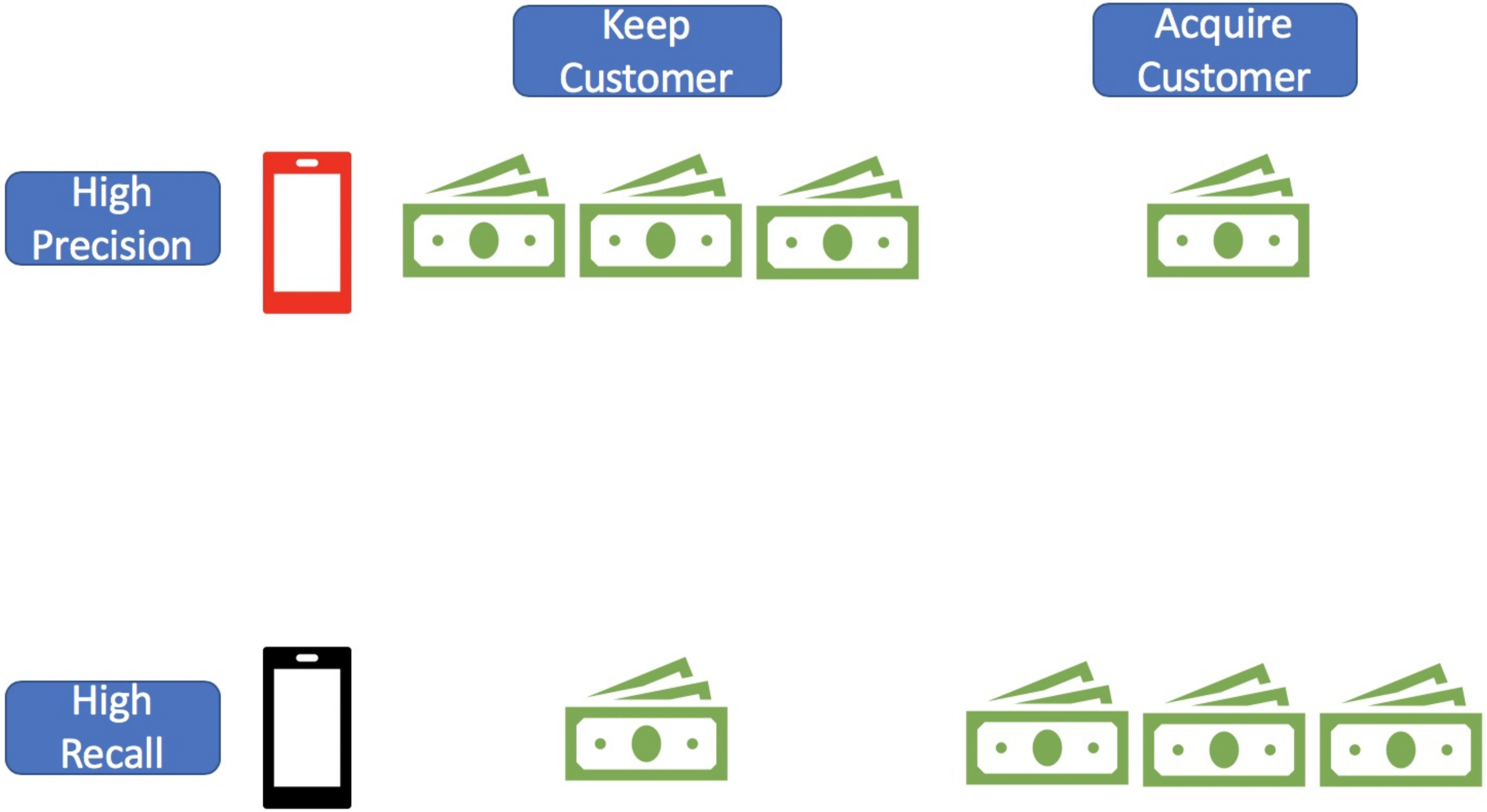
- A model with high precision indicates:
  - Few false positives ("false alarms")
  - Not many non-churners were classified as churners

Recall

Metric	Formula
Recall/Sensitivity	True Positives / (True Positives + False Negatives)

- A model with high recall indicates that it correctly classified most churners

Precision vs. Recall



Confusion Matrix in scikit-learn

```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
```

Let's practice!

PREDICTING CUSTOMER CHURN IN PYTHON

Other model metrics

PREDICTING CUSTOMER CHURN IN PYTHON

Mark Peterson
Senior Data Scientist, Alliance Data

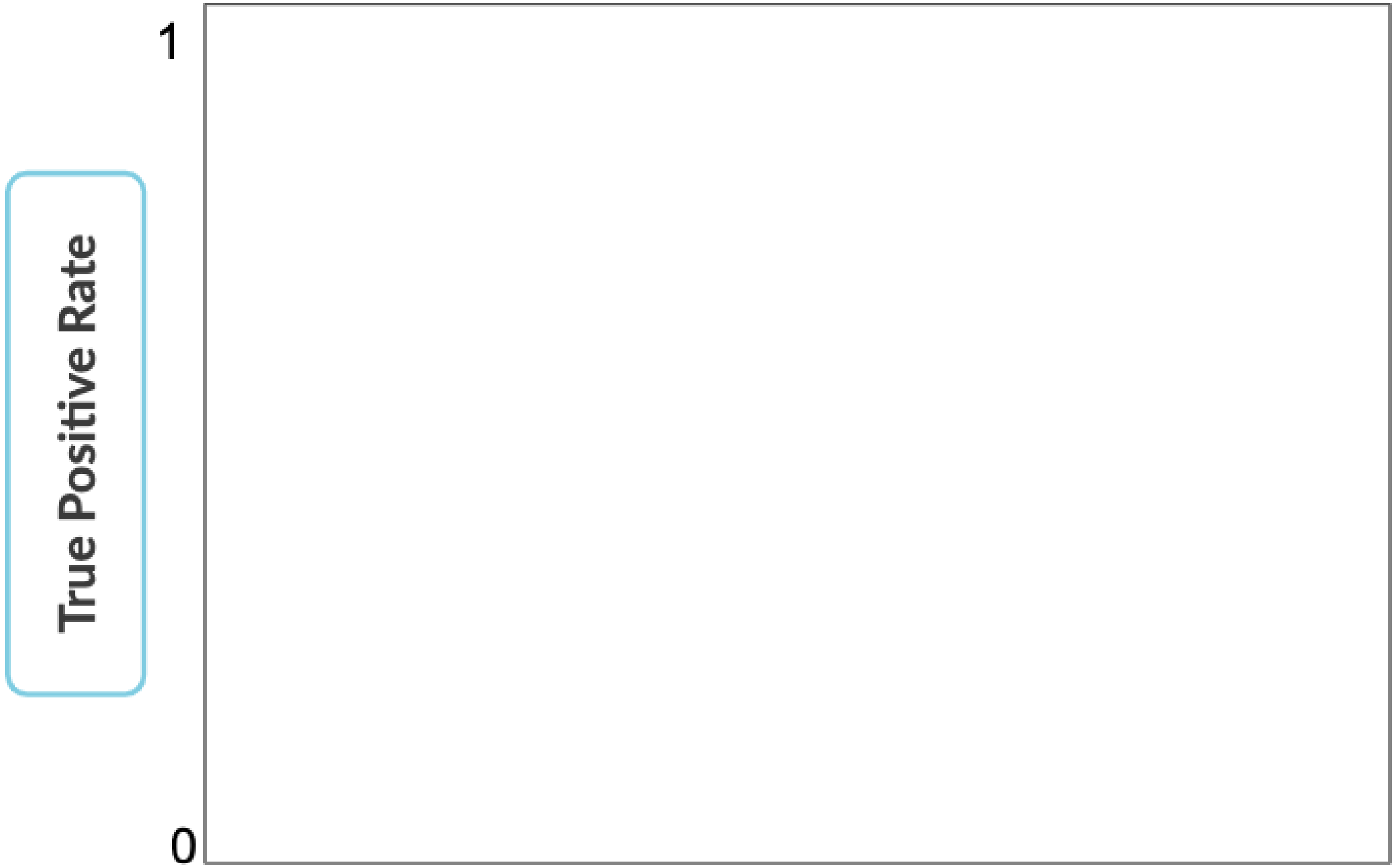
Probability thresholds

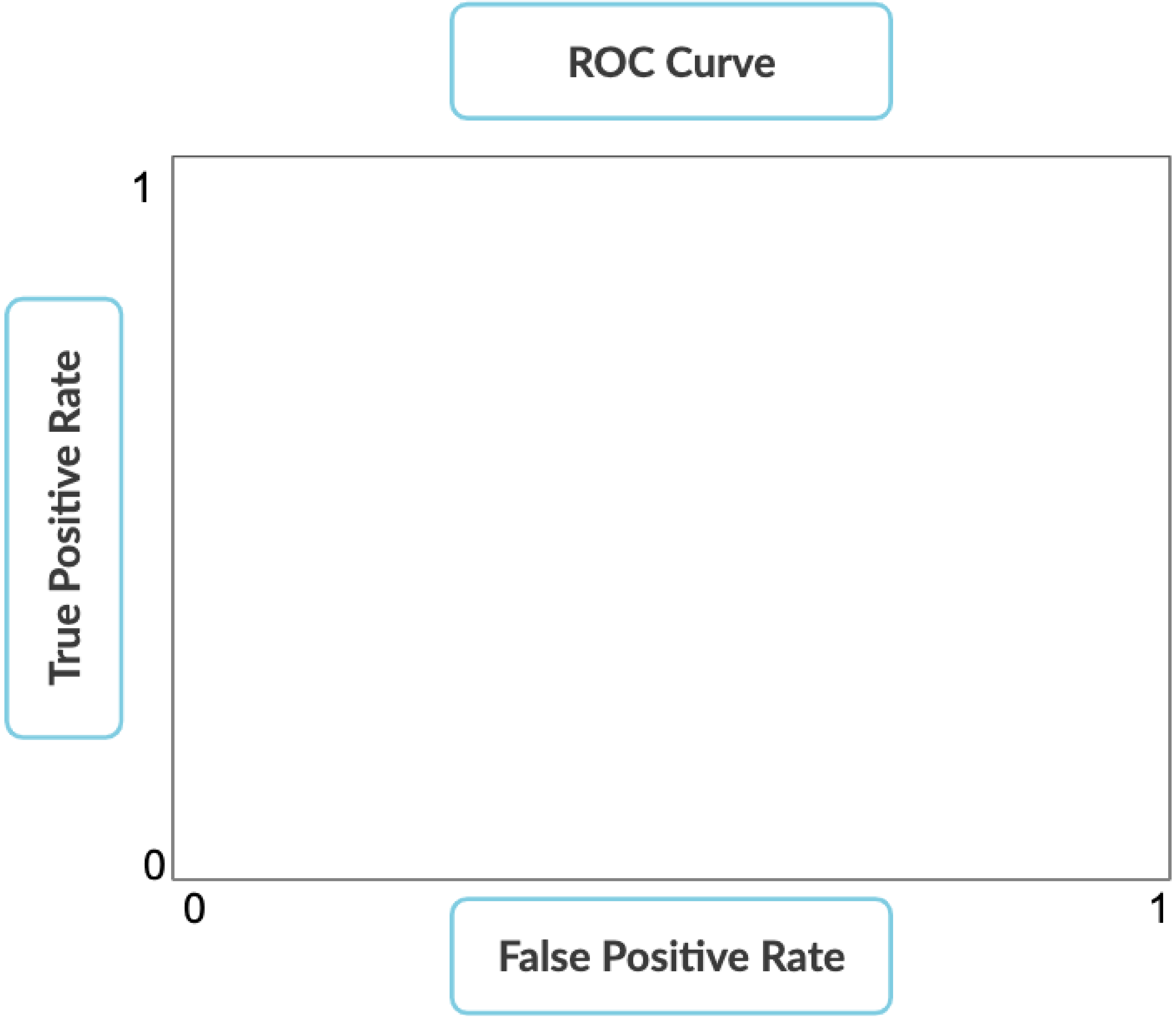
- Every prediction your classifier makes has an associated probability
- Default probability threshold in scikit-learn: 50%

    - What if we vary this threshold?

ROC Curve

ROC Curve

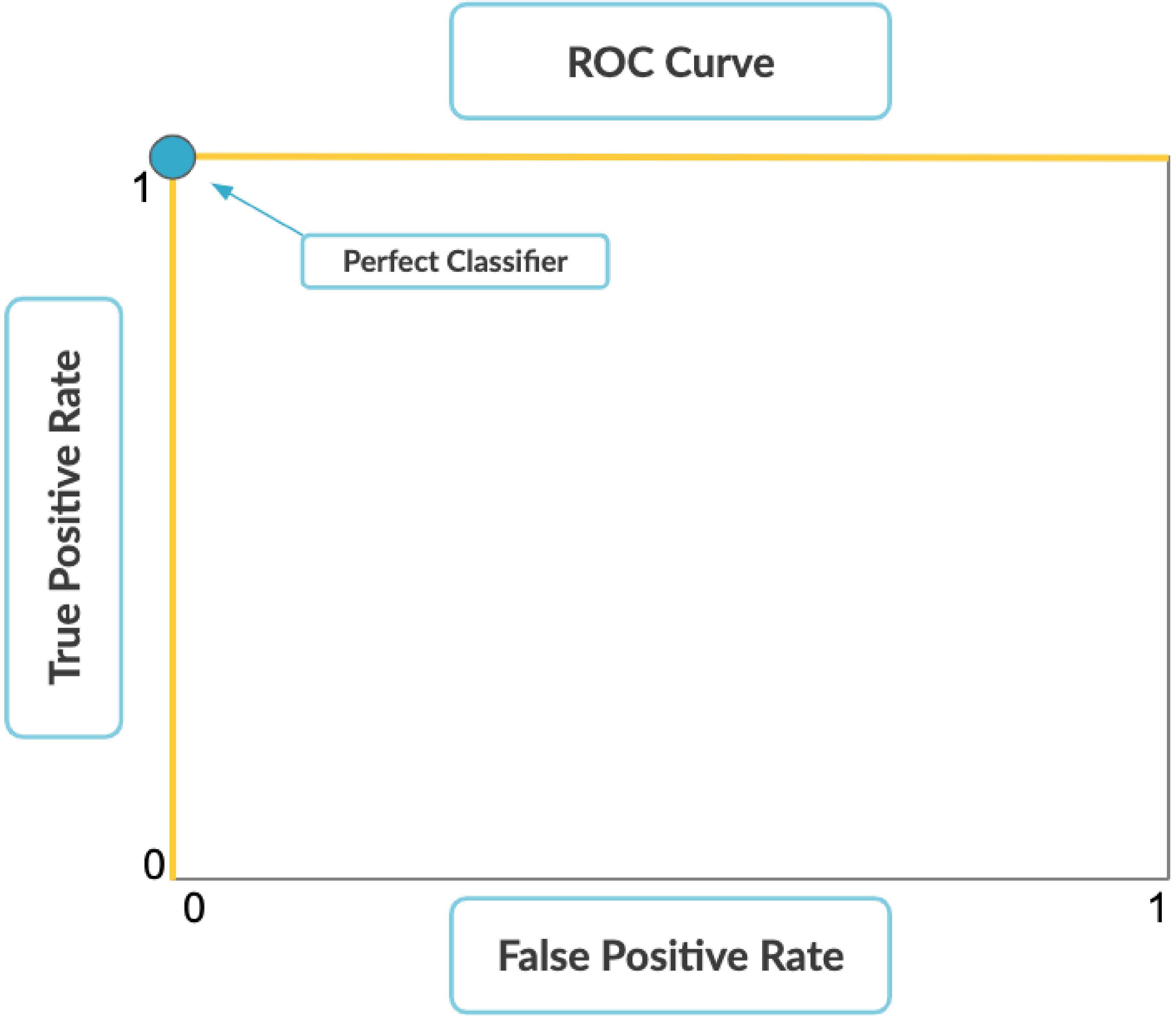


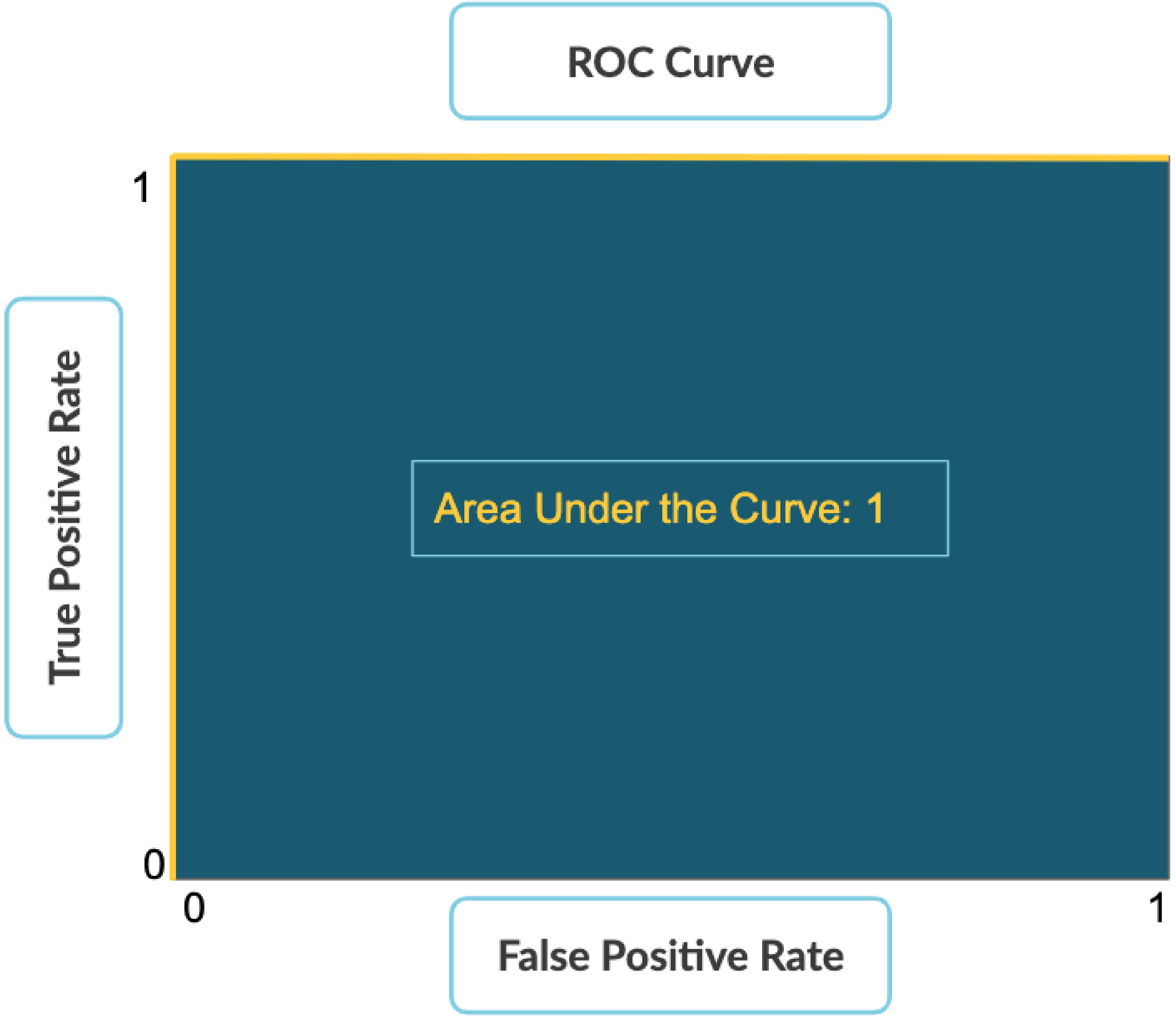


ROC Curve

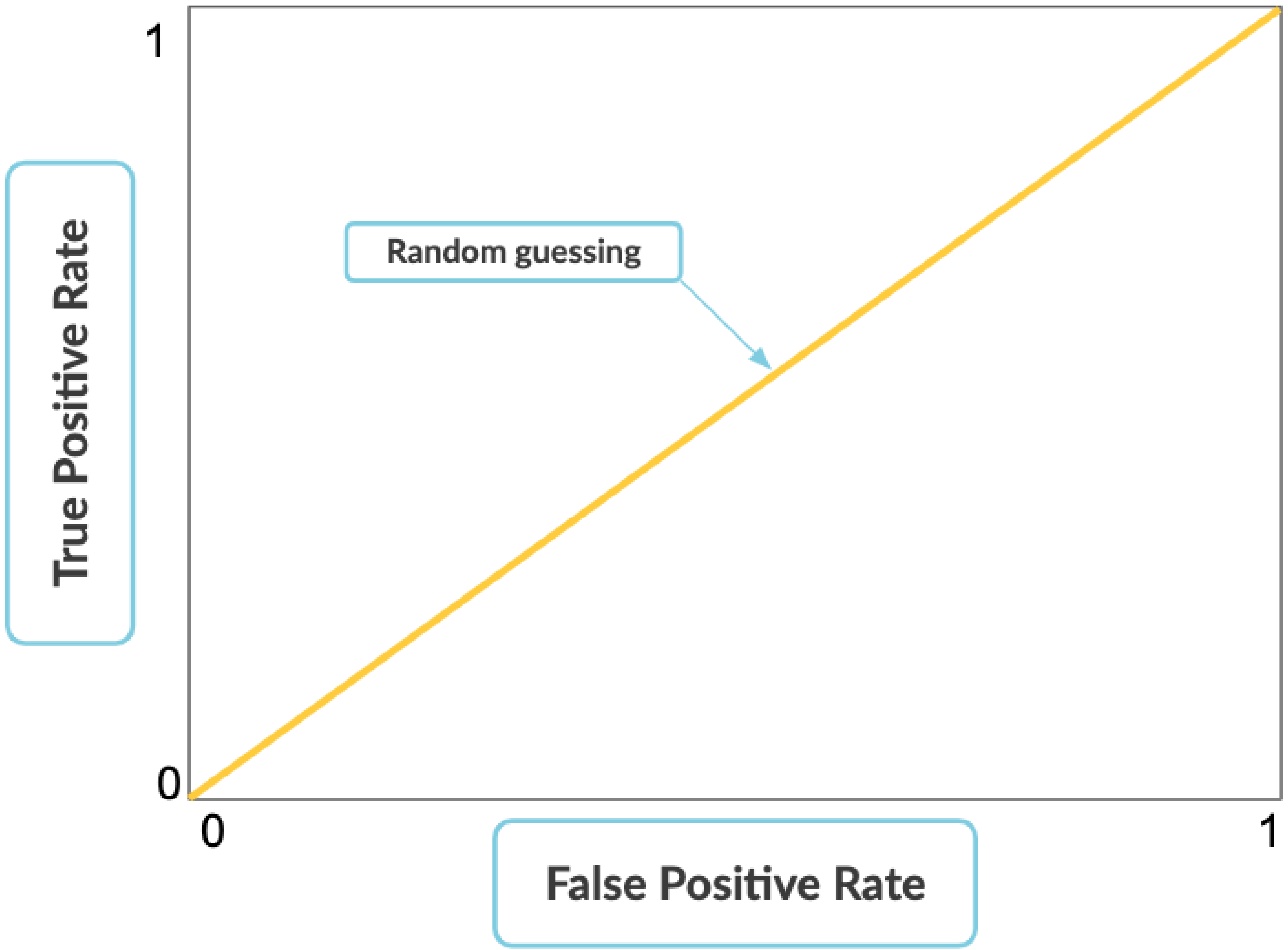
True Positive Rate

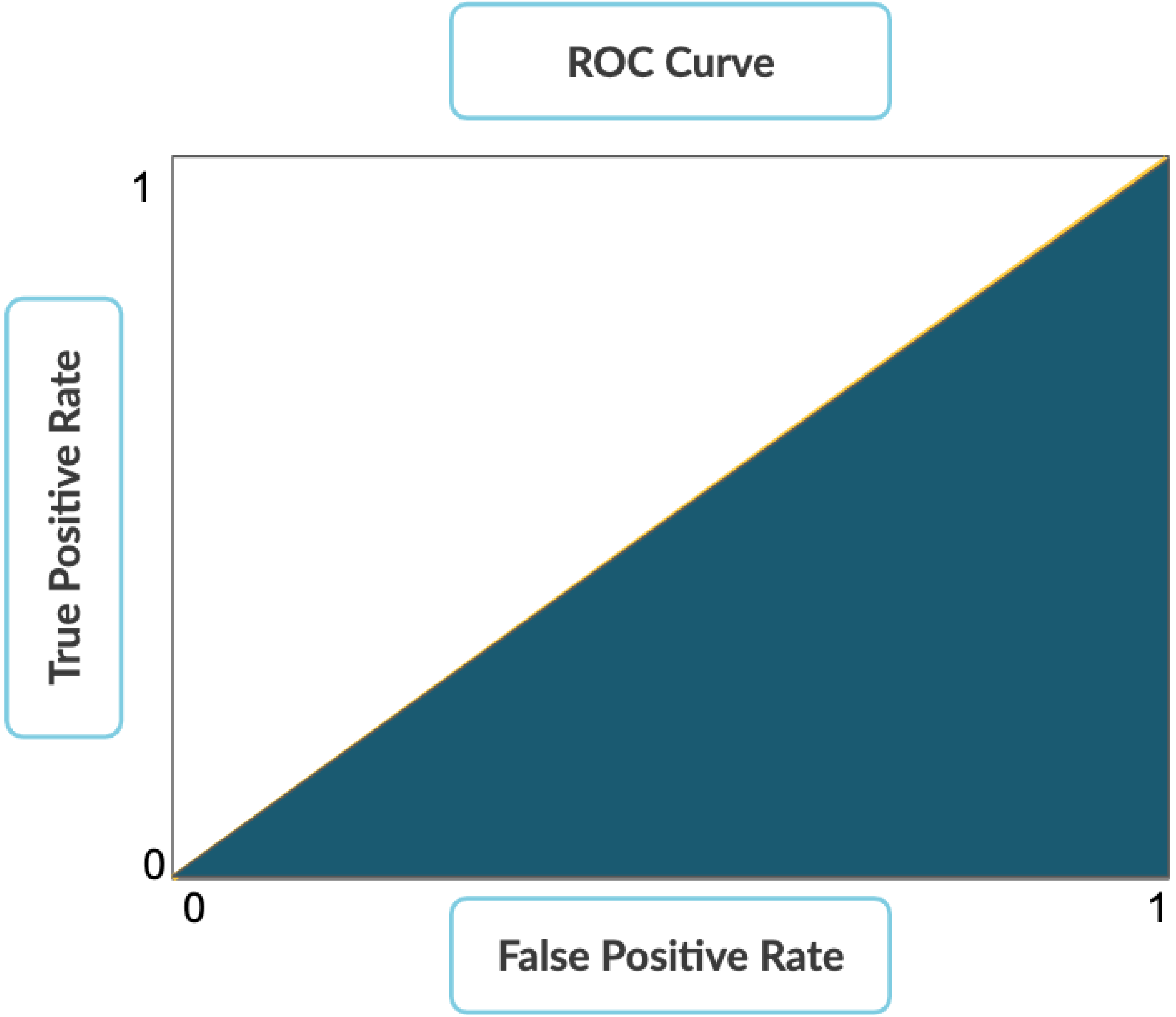
False Positive Rate

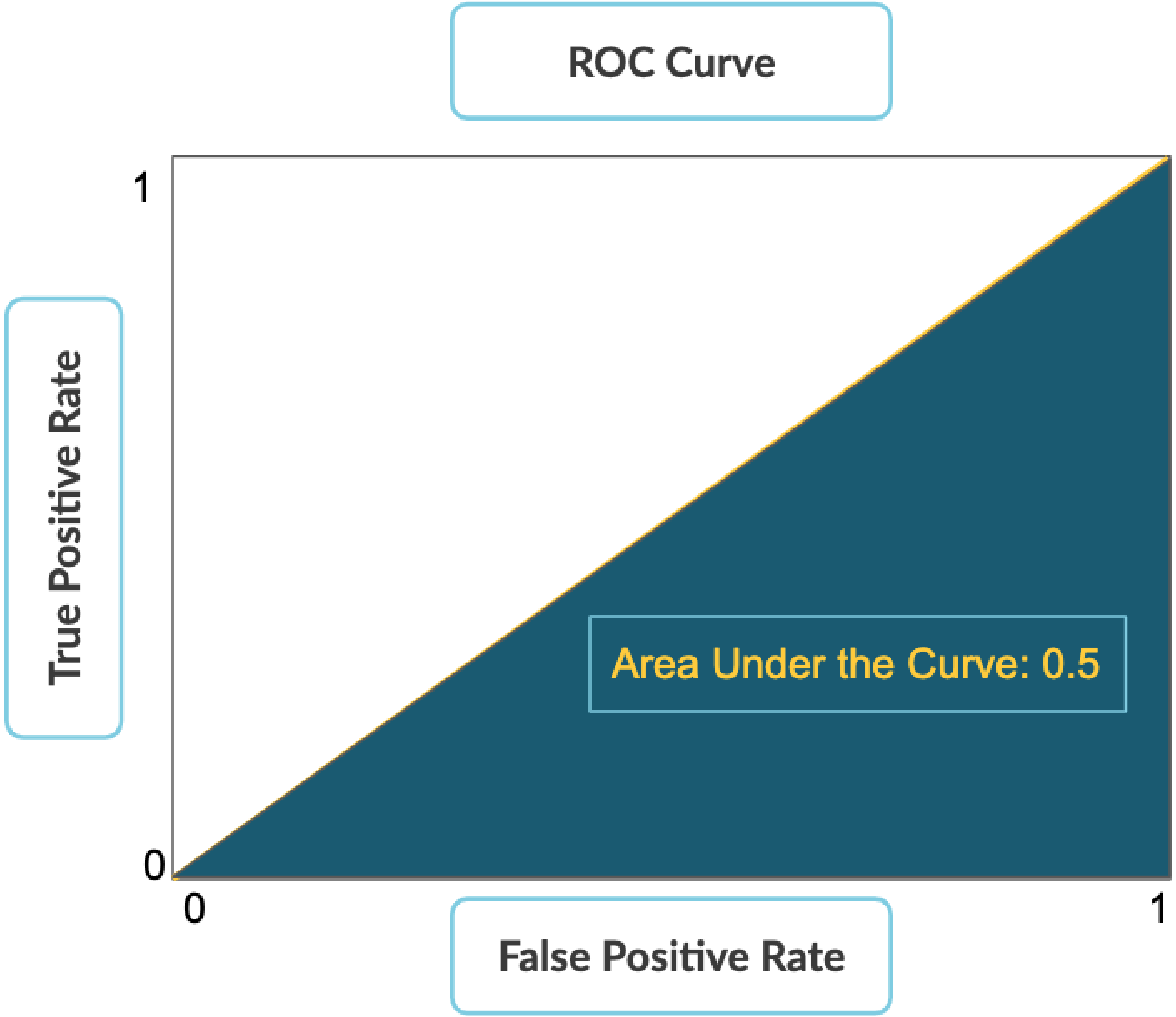

ROC Curve
Perfect Classifier
True Positive Rate
False Positive Rate
1
0
0
1


ROC Curve
1
0
True Positive Rate
Area Under the Curve: 1
0
1
False Positive Rate

ROC Curve


1
True Positive Rate
0
Random guessing
0
1
False Positive Rate


ROC Curve
1
True Positive Rate
0
0
1
False Positive Rate


ROC Curve
1
0
True Positive Rate
Area Under the Curve: 0.5
0
1
False Positive Rate

Generating probabilities in sklearn

```
logreg.predict_proba(X_test)[:,1]
```

```
array([[0.80188981, 0.19811019],
       [0.96484075, 0.03515925],
       [0.9182671 , 0.0817329 ],
       ...,
```

```
y_pred_prob = logreg.predict_proba(X_test)[:,1]
```

ROC curve in sklearn

```python
from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
```

```python
import matplotlib.pyplot as plt

plt.plot(fpr, tpr)

plt.xlabel("False Positive Rate")

plt.ylabel("True Positive Rate")

plt.plot([0, 1], [0, 1], "k--")

plt.show()
```

Area under the curve

```
from sklearn.metrics import roc_auc_score


auc = roc_auc_score(y_test, y_pred)
```

Let's practice!

PREDICTING CUSTOMER CHURN IN PYTHON