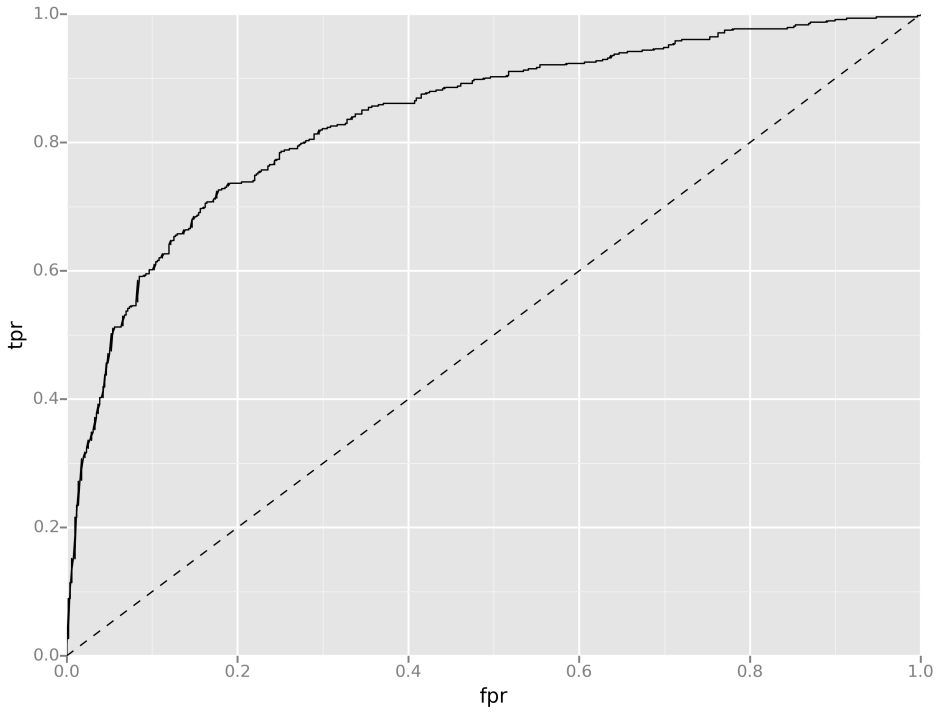
Calculating an ROC Curve in Python

- scikit-learn makes it easy to calculate ROC Curves.
- Firstly, we need a classification model to evaluate.
- For this example, we will make a synthetic dataset and then build a logistic regression model using scikit-learn.

```
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression

# Make a Simulated Data Set
X, y = make_classification(n_samples=10000,
         n_features=10, n_classes=2,
         n_informative=5)

# Split it into Testing and Training Sets
Xtrain = X[:9000]
Xtest = X[9000:]
ytrain = y[:9000]
ytest = y[9000:]

# Train the Model
clf = LogisticRegression()
clf.fit(Xtrain, ytrain)


# Apply model to Testing Data
preds = clf.predict_proba(Xtest)[:,1]
```

```
from sklearn import metrics
import pandas as pd
from ggplot import *


fpr, tpr, _ = metrics.roc_curve(ytest, preds)

## Create as DataFrame

df = pd.DataFrame(dict(fpr=fpr, tpr=tpr))

## ggplote takes pandas Data frames as arguments

ggplot(df, aes(x='fpr', y='tpr')) +
    geom_line() + geom_abline(linetype='dashed')
```



Finally to calculate the AUC:

```
auc = metrics.auc(fpr,tpr)

# Add this to the plot
#  - see the "ggtitle" option

ggplot(df, aes(x='fpr', ymin=0, ymax='tpr')) +
    geom_area(alpha=0.2) +
    geom_line(aes(y='tpr')) +
    ggtitle("ROC Curve w/ AUC=%s" % str(auc))
```

The AUC is 0.900.