Machine Learning and Programming in Python

Lecture for Master and PhD students

Chair of Data Science in Economics

Ruhr University Bochum

Summer semester 2024

Lecture 7

Model evaluation for Regression

- 1. Accuracy of $\hat{\beta}$s :
  - Hypothesis tests:
    H_0: no relation between X and Y, $\beta_1 = 0$
    H_1: there is some relation between X and Y, $\beta_1 \neq 0$
  - Standard error of $\hat{\beta}_1$:
    $SE(\hat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$
    How much does $\hat{\beta}_1$ deviate from β_1?
  - t-value:
    $t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)}$, significant if $|t| > 1.645$
    How many standard deviations is $\hat{\beta}_1$ away from zero?
  - p-value :
    If small p-value, then it will be unlikely that the relation between X and Y is just random, H_0 will be rejected
    Significant if p-value < 0.1

- 2. Accuracy of the Model :
  - $RSE = \sqrt{\frac{1}{n-2} RSS} = \sqrt{\frac{1}{n-2} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$
    Lack of fit of model
  - $R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$
    with $TSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$
    gives the share of variability in Y which can be explained by X
    RSS= leftover unexplained variance after regression
    TSS=variability in Y before regression

Note: Can compute the errors from regression tasks, based on the y and $\hat{y}$

How to assess the accuracy of the **classifier** $\hat{f}$?

- Training error rate: Proportion of mistakes when we apply $\hat{f}$ to the training data

  $\frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$

- Test error rate: Proportion of mistakes in the test data

  $\frac{1}{m} \sum_{i=1}^{m} |y_i - \hat{y}_i|$

- Two types of classification errors:
  - False positives: predicted $\hat{y} = 1$ when true $y = 0$. (Type I error)
  - False negatives: predicted $\hat{y} = 0$ when true $y = 1$. (Type II error)
- E.g., in predicting default:
  - False positives: (incorrectly) predicted default among customers who actually don't default.
  - False negatives: (incorrectly) predicted non-default for actual defaulters.

Confusion matrix

		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

Prediction accuracy metrics

- accuracy = $\frac{TP+TN}{N+P}$
- recall (sensitivity, or true positive rate) = $\frac{TP}{P}$
- specificity = $\frac{TN}{N}$
- false positive rate (1 - specificity) = $\frac{FP}{N}$
- precision = $\frac{TP}{P*}$

Accuracy

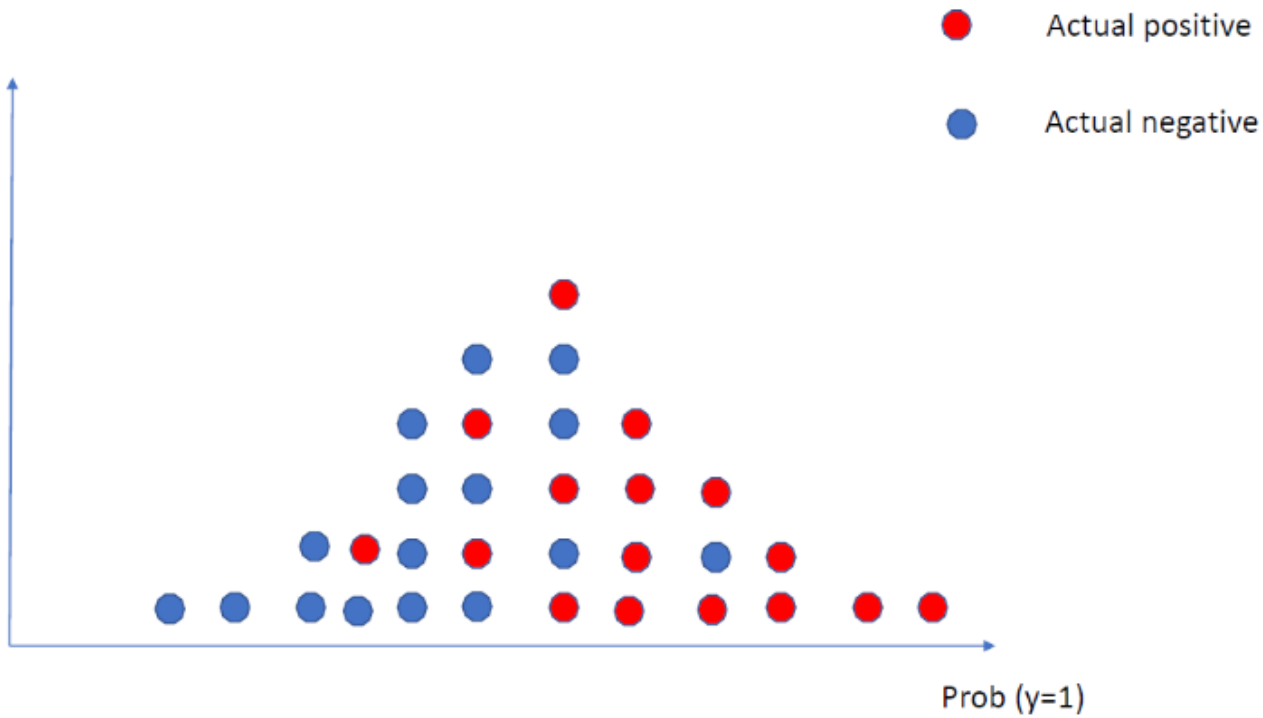
- A measure of overall prediction quality but is not always useful.
- If one class dominates, say 95% of the cases are 'negative'.
  - Is a classifier with 97% accuracy good?
  - What is the accuracy of a classifier that always predicts negative?
- Different costs/benefits of false positives vs. false negatives.
  - Consider a diagnostic test for cancer.
  - A false positive implies another (confirmatory) test.
  - A false negative could delay treatment and endanger the patient's life.

Precision versus Recall

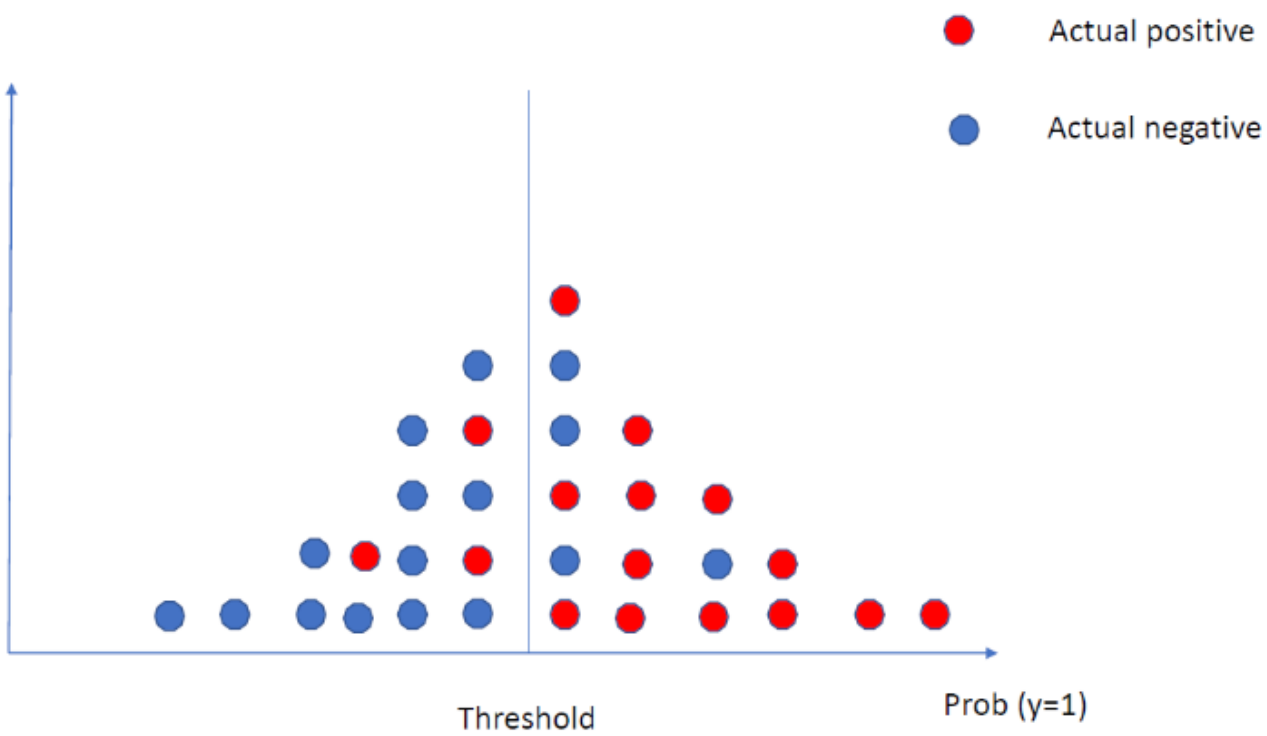
- Recall ($\frac{TP}{P}$) is the fraction of *actual* positive cases correctly classified.
- Precision ($\frac{TP}{P_*}$) is the fraction of *predicted* positive cases correctly classified.
- Diagnosis of a dangerous disease? Recall more important than precision?
- Netflix recommendations? Precision more important than recall?
- However, multiple metrics important to evaluate a classifier:
  - Can trivially get 100% recall by classifying every case as positive; perfect recall but very imprecise.
  - Can achieve high precision but poor recall by using a strict threshold.
- Explore such trade-offs using precision-recall curve or ROC curve.

- Depending on the domain, define a threshold
- e.g. $\overline{p}$ to predict $\hat{y}|x$
- e.g. $\overline{p} = 0.5 \Rightarrow \hat{y}|x = 1$ if $\hat{p}(y = 1|x) > 0.5$
- But a bank may choose the default probability threshold conservatively, say $\hat{p}(y = 1|x) > 0.2$

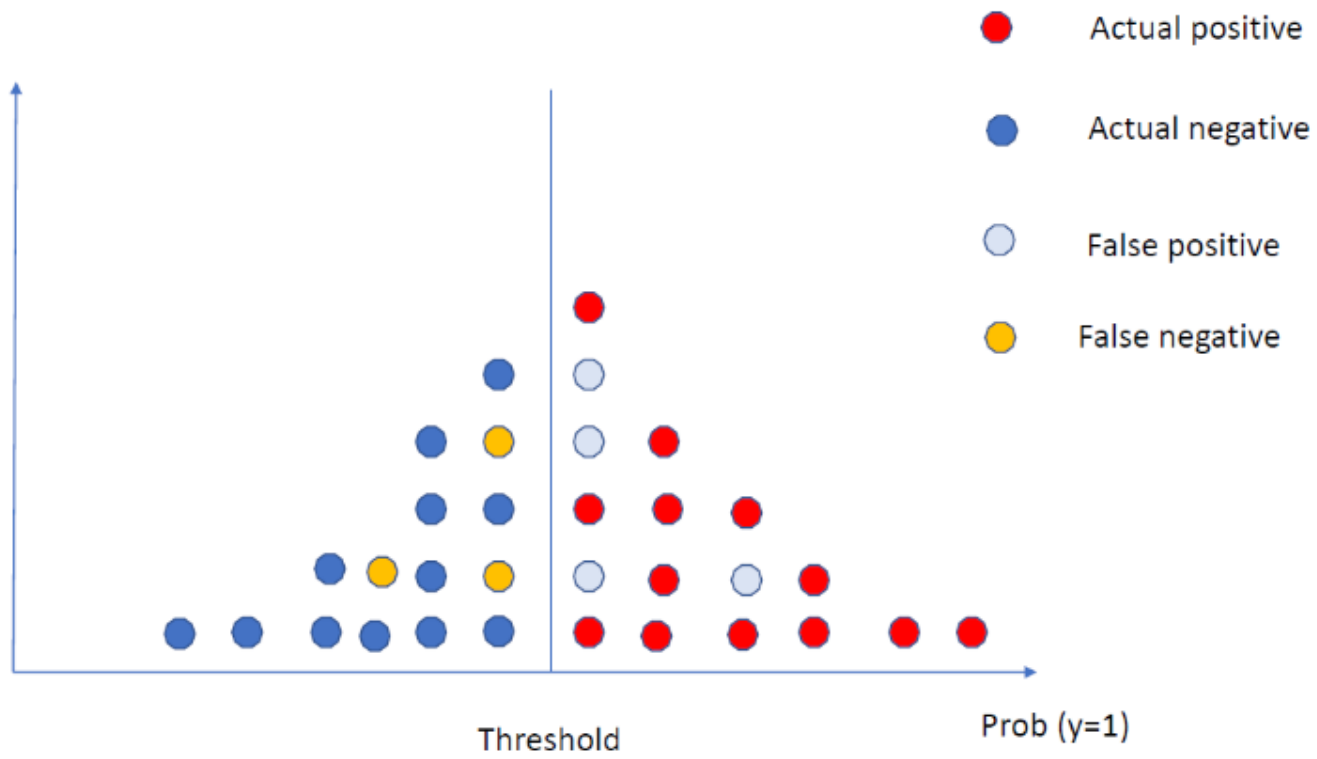
ROC Illustration



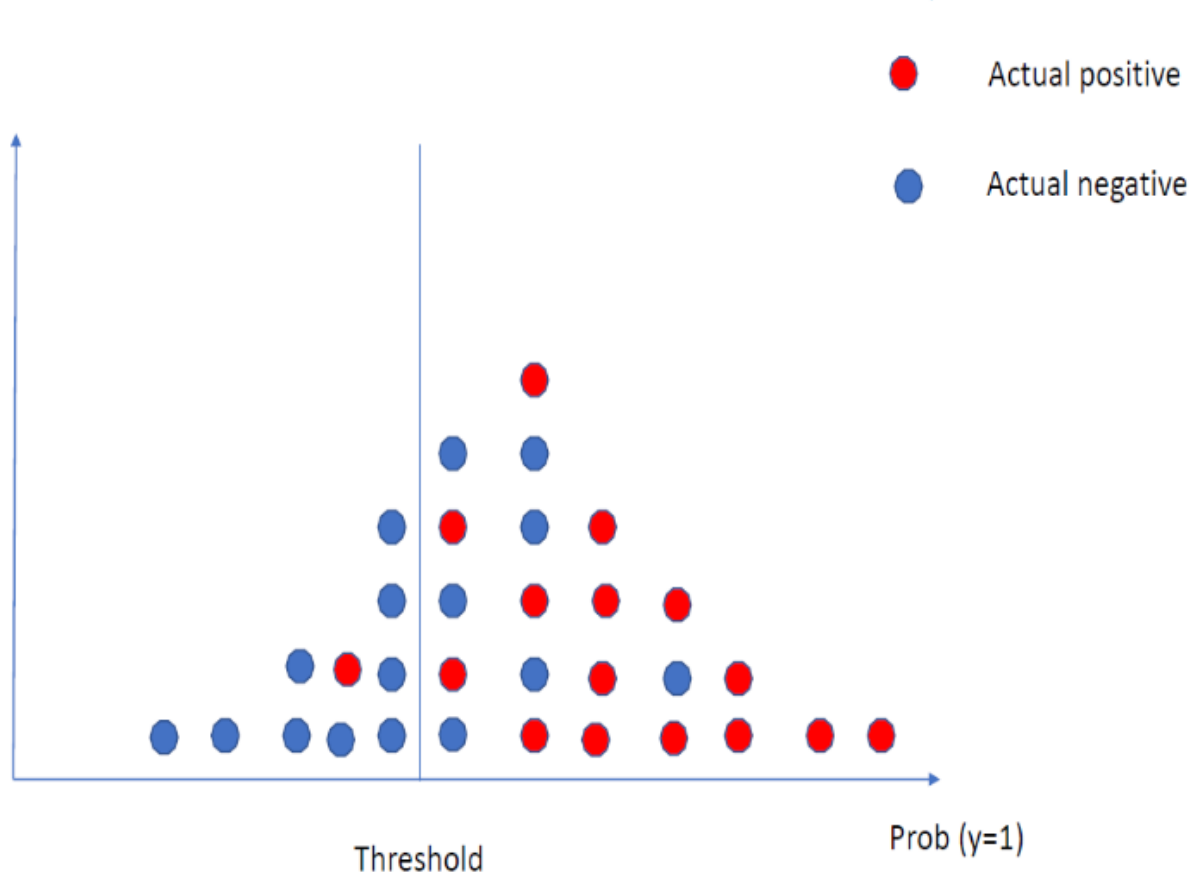
ROC Illustration



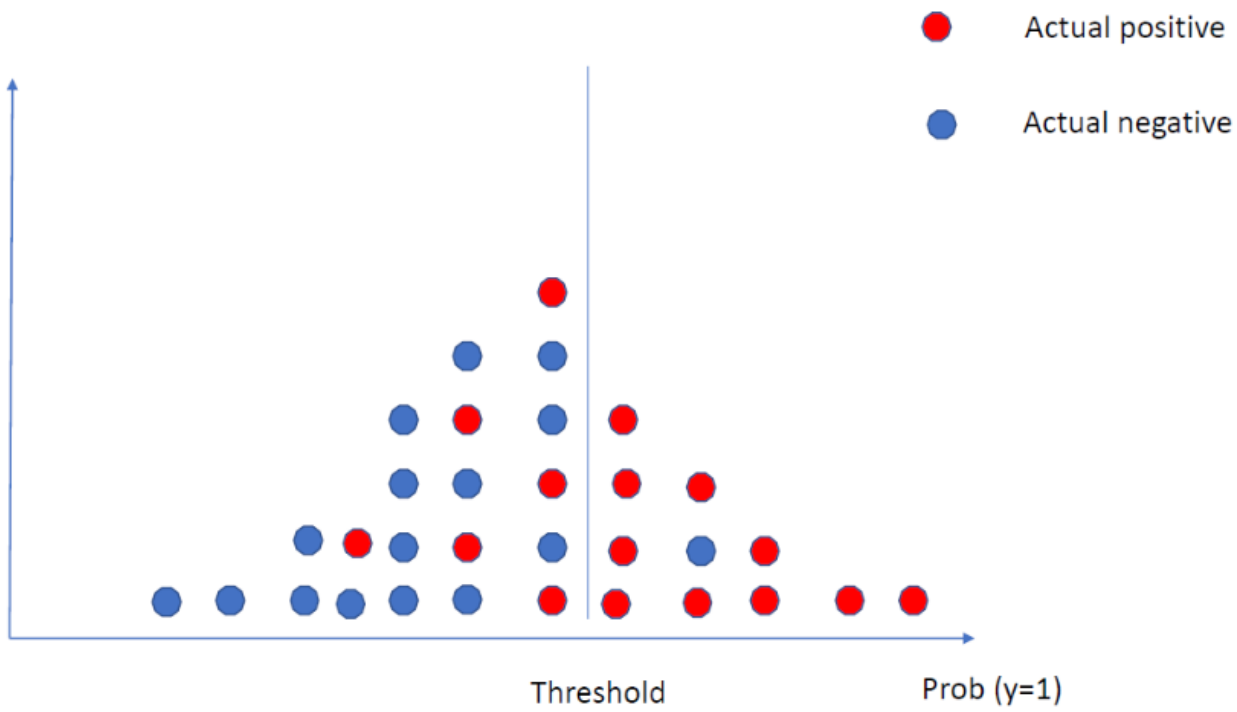
ROC Illustration



Smaller threshold reduces false negatives, but reduces also true negatives

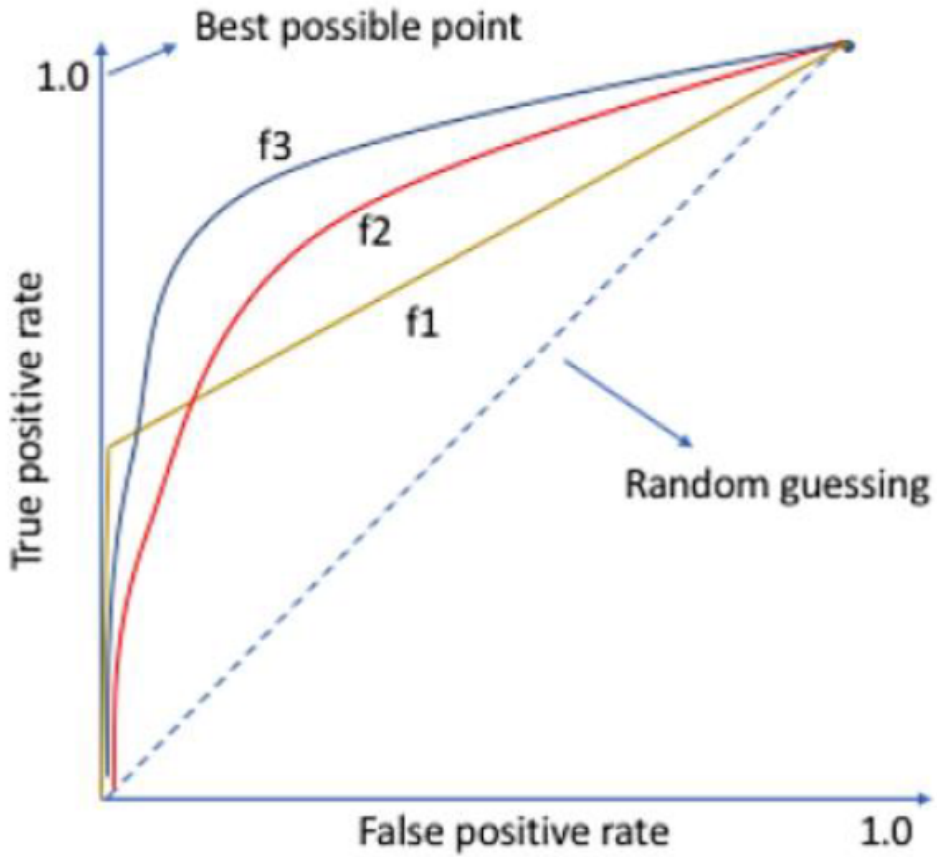


Higher threshold reduces false positives, but reduces also true positives



ROC = Receiver Operating Curve

Plots the true positive rate (Recall) against the false positive rate



- Prefer High TPR and low FPR.
- Perfect classifier has TPR=1 and FPR=0.
- After estimating $\hat{f}$, can plot the ROC by varying the threshold probability ($\bar{p}$) for classification.
  - A very high $\bar{p}$ prevents false positives but also misses many true positives.
  - Lowering $\bar{p}$ increases true positives but also increases false positive.
- Random guessing will trivially set TPR = FPR. Why?
- *Area Under Curve*; AUC is area under the ROC; a measure of predictive accuracy. High AUC is better.

Data Project

- Titanic data set

- Fit Logistic Regression, Linear Discriminant regression, Quadratic Discriminant Regression to the data
- Display the predicted y's
- Show model evaluation metrics

Predicting Titanic survivors

- On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew.
- Some groups of people were more likely to survive than others, such as women, children, and the upper-class.
- Kaggle has an ML challenge since 2012 (over 10,000 teams so far) for the best prediction of who survived.
- Build a Logistic Regression model as an example.

Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

- First few rows of training data:
  - **Sibsp** Num. of siblings and spouse aboard.
  - **Parch** Num. of parents and children aboard.
  - **Pclass** 1: Upper, 2: Middle, 3: Lower.
  - **embarked**: Port of embarkation, C (Cherbourg), Q (Queenstown), S (Southampton).

- Check 'survived' (y) is binary.
- Check for missing values (use *.isnull().sum()* method on training dataframe for a quick count).
  - Drop variables that have many missing values.
  - Age has some missing values: drop or impute the missing?
- Generate dummies for categorical variables e.g. pclass, sex, port of emarkation
- Check correlations (cannot use perfectly correlated variables).
- Finalise X variables and generate train and test data split.

- Screenshot of code from Jupyter Notebook:

```
In [64]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = .25, random_state=25)

In [74]: Y_train.shape

Out[74]: (535,)

In [75]: Y_test.shape

Out[75]: (179,)

In [65]: logit=LogisticRegression()

In [66]: logit.fit(X_train, Y_train)

Out[66]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                            intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                            penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                            verbose=0, warm_start=False)
```

- Sckit-learn **metrics** offers many performance measures: accuracy, confusion matrix, precision, recall etc.

```
Y_pred=logit.predict(X_test)
from sklearn import metrics
confusion_matrix = metrics.confusion_matrix(Y_test, Y_pred)
print(confusion_matrix)
```

```
[[86 17]
 [21 55]]
```

```
print(metrics.classification_report(Y_test, Y_pred))
```

```
             precision    recall  f1-score   support

          0       0.80      0.83      0.82       103
          1       0.76      0.72      0.74        76

avg / total       0.79      0.79      0.79       179
```

```
print(metrics.accuracy_score(Y_test, Y_pred))
```

```
0.787709497207
```

- Sckit-learn **metrics** also allows us to plot the ROC.
- You have to provide the actual test y, and the predicted probabilities.
- Output is TPR, FPR, and classification thresholds.

Now let us plot the ROC. First, we need the predicted probabilities in the test sample.

```
prob=logit.predict_proba(X_test)
```

What does `prob' have?

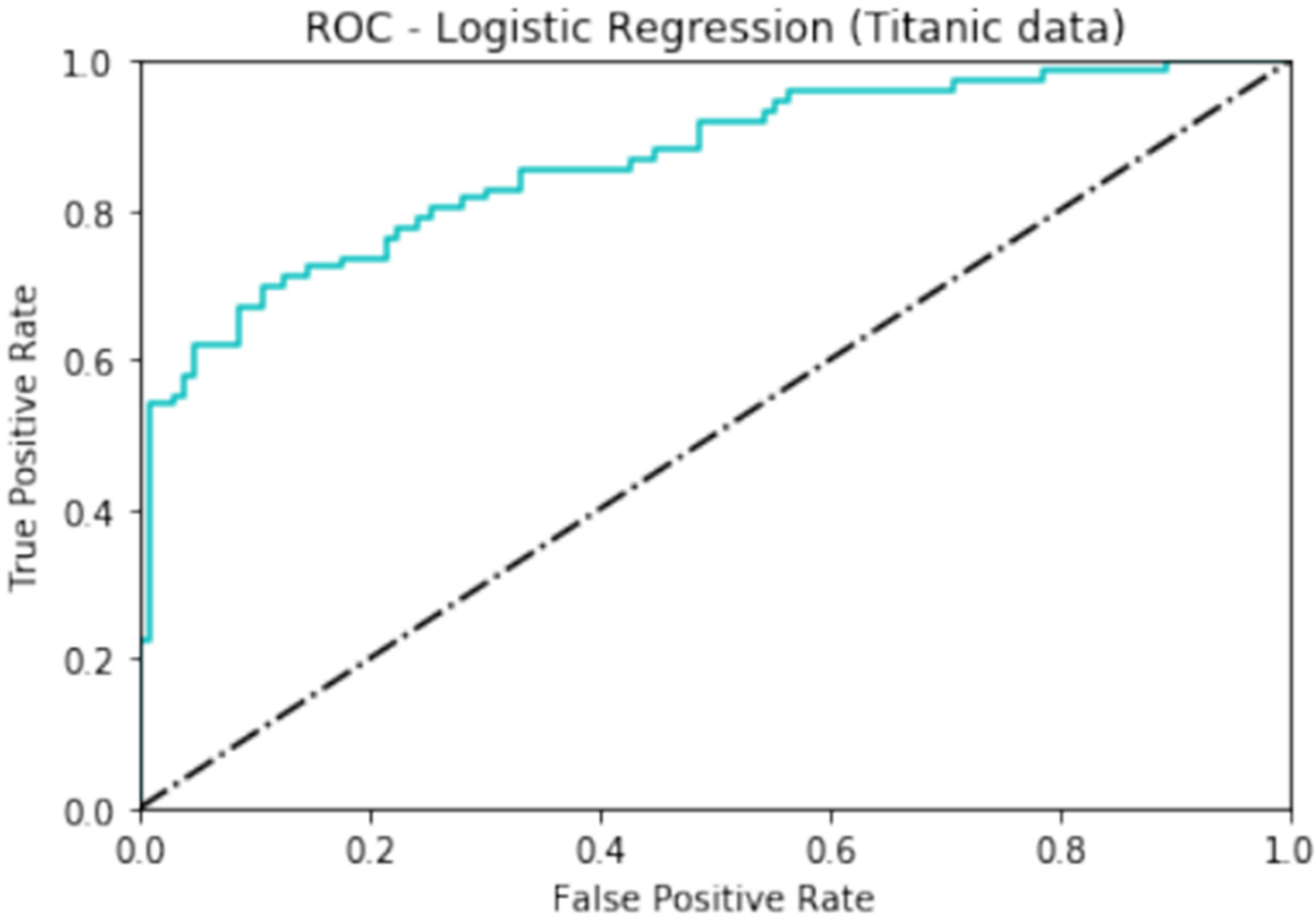
```
prob[0:4]
```

```
array([[ 0.36674199,  0.63325801],
       [ 0.45361404,  0.54638596],
       [ 0.09670912,  0.90329088],
       [ 0.251909  ,  0.748091  ]])
```

It is a Nx2 array where N is the number of test observations and the first column (column 0) is $p(y = 0)$ and second is $p(y = 1)$. Let us grab the second column.

```
pred=prob[:,1]
```

```
fpr, tpr, threshold = metrics.roc_curve(Y_test, pred)
```


ROC - Logistic Regression (Titanic data)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0