MACHINE LEARNING FUNDAMENTALS

Amir Ghaderi, Ph.D., P.Eng.
February – March 2020
Tillyard Auditorium

Session 3 & 4: March 3rd & March 4th

1

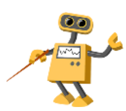
2

Supervised Classification: Binary Classification

- In binary classification, a collection of objects is given, and the task is to classify the objects into two groups based on their features

- Evaluating a classifier means measuring how accurately the predicted classes or labels match the true labels in the evaluation set. This is most often trickier than evaluation of a regression model

In binary classification, we usually call the smaller and more interesting of the two classes as positive and the larger/other class as negative



2

3

Classification: Skewed and unbalanced data set

Coronavirus

Sample	Age	Body Temperature	Coughing Frequency	Runny nose	Test Result
	Measurements				Detection Kit Results
1	36	39	High	No	+
2	50	37	NA	Yes	-
3	81	37.5	High	Yes	+
4	47	40	Moderate	No	+
5	47	37.6	Low	NA	-
…	…	…	…	…	…
10694	55	38.1	Low	No	-

+ 133
- 10561

You have been asked to give a model to replace kit detection method.
What is the accuracy of a model which predicts constant "negative" output?

3

Classification of Imbalanced Dataset: Beyond Accuracy - Recall and Precision

- The infection detection task is an **imbalanced (skewed) classification problem** (a very common problem in ML)
- Our intuition says we should focus on identifying the positive cases (statistically called **recall**)

What is the recall of a model which generates a constant "positive" output for coronavirus dataset? Is it an exact classifier?

This new model would suffer from low **precision.**

What is the recall and precision of a model which predicts 97 actual positive cases and 219 actual negative cases positive and the rest negative?

Recall=72% **Precision=30%**

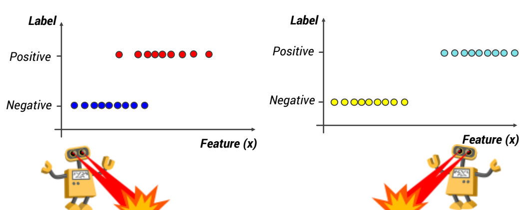
Recall refers to the percentage of total relevant results correctly classified by model

Trade-off

Precision expresses the percentage of the model's results which are relevant

4

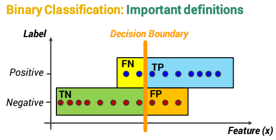
Binary Classification: Example



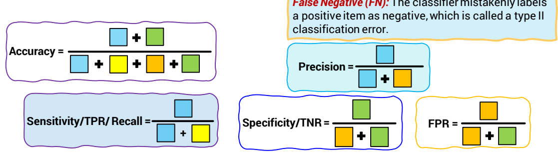
Any suggestion for a classification line for each example?

5

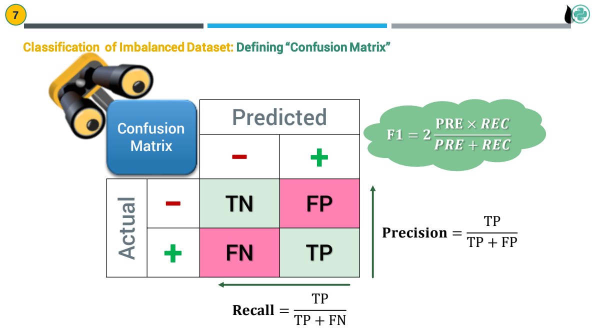
Binary Classification: Important definitions



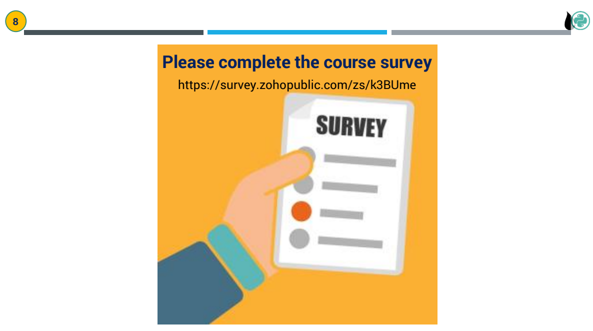
True Positive (TP): Here the classifier predicts or labels a positive item as positive which is a correct prediction.
True Negative (TN): Here the classifier correctly determines that a member of the negative class deserves a negative label which is a correct prediction again.
False Positive (FP): The classifier predicts a negative item as a positive by mistake, which is called a type I classification error.
False Negative (FN): The classifier mistakenly labels a positive item as negative, which is called a type II classification error.



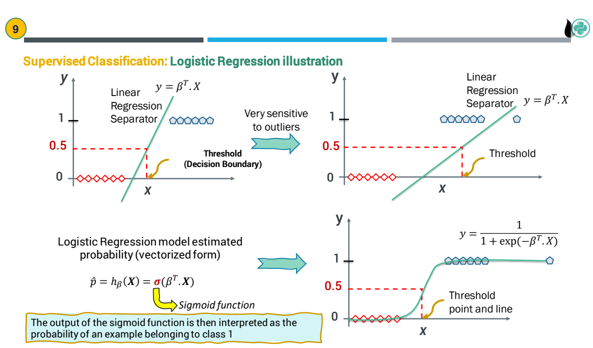
6


7
Classification of Imbalanced Dataset: Defining "Confusion Matrix"
Confusion Matrix
Predicted
Actual
TN
FP
FN
TP
$F1 = 2\frac{PRE \times REC}{PRE + REC}$
$\text{Precision} = \frac{TP}{TP + FP}$
$\text{Recall} = \frac{TP}{TP + FN}$

7


8
Please complete the course survey
https://survey.zohopublic.com/zs/k3BUme
SURVEY

8


9
Supervised Classification: Logistic Regression illustration
Linear Regression Separator
$y = \beta^T.X$
Very sensitive to outliers
Threshold (Decision Boundary)
Threshold
Logistic Regression model estimated probability (vectorized form)
$\hat{p} = h_\beta(X) = \sigma(\beta^T.X)$
Sigmoid function
The output of the sigmoid function is then interpreted as the probability of an example belonging to class 1
$y = \frac{1}{1 + \exp(-\beta^T.X)}$
Threshold point and line

9

10

Supervised Classification: Logistic Regression

♋ Logistic Regression is commonly used to estimate the probability that an instance belongs to a particular class. It is one of the most widely used algorithms for classification in industry ♋

$$\hat{y} = \begin{cases} 0 \ if \ \hat{p} \geq 0.5 \\ 1 \ if \ \hat{p} < 0.5 \end{cases}$$

Binary classifier

Learning the weights: Cost Function

$$J(\beta) = \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\log(\hat{p}^{(i)}) - (1-y^{(i)})\log(1-\hat{p}^{(i)})\right)$$

bad news! It doesn't have a closed form solution

$$\frac{\partial}{\partial \beta_j}J(\beta) = \frac{1}{m}\sum_{i=1}^{m}\left(\sigma(\beta^T.x^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

Using GD with suitable learning rate gives the global minimum coefficients

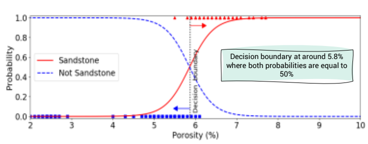
10

11

Logistic Regression: Implementation

Code 1

```
x_train, x_test, y_train, y_test = \
    train_test_split(x, y, test_size=0.1, random_state=42)
log_reg = LogisticRegression(solver='lbfgs') ❶
log_reg.fit(x_train, y_train)
x_prediction = np.linspace(2, 10, 1000).reshape(-1, 1)
y_probability = log_reg.predict_proba(x_prediction) ❷
decision_boundary = x_prediction[y_probability[:, 1] >= 0.5][0, 0]
```



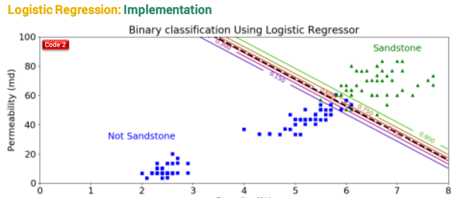
❶ `LogisticRegression` can use different solvers to tackle different problems. Although 'lbfgs' is good for most of the problem with moderate size, you may consider changing it for your problem.

❷ The `predic_prob()` method of is used to determine the likelihood of a datapoint belong to a certain class. To predict the class, you should use `predict()` method.

11

12

Logistic Regression: Implementation



The hyperparameter controlling the regularization is not alpha, but its inverse: C. The higher the value of C, the *less* the model is regularized.

l2 (Rigde) regularization is the default scheme in scikit-learn

Each parallel line represents the points where the model outputs a specific probability, from 15% (bottom left) to 90% (top right)

Drive the equation for the boundary line representing $\hat{p} = 50\%$ & 90% ?

12

13

Logistic Regression: Confusion matrix

		Predicted: Carbonate	Predicted: Sandstone
Actual	Carbonate	64	1
	Sandstone	13	21

Code 3

Model's Accuracy: 86%

Precision of Sandstone Classifier: 95.5%

Recall of Sandstone Classifier: 62%

f1-score of Sandstone Classifier: 75%

? What are the values for the "carbonate" classifier?

13

14

Softmax (Multinomial Logistic) Regression: How it works

The Logistic Regression model can be generalized to support multiple classes directly (not multi-output though)

when given an instance **x**, the Softmax Regression model first computes a score *sk*(**x**) for each class *k*, then estimates the probability of each class by applying the *softmax function* (also called the *normalized exponential*) to the scores.

Softmax score for class k

$$s_k(\mathbf{x}) = \beta_k^T . \mathbf{x}$$

Note that each class has its own dedicated parameter vector β_k^T

Softmax function

$$\hat{p} = \sigma(s(x))_k = \frac{\exp(s_k(x))}{\sum_{j=1}^{K} \exp(s_j(x))}$$

K is the number of classes

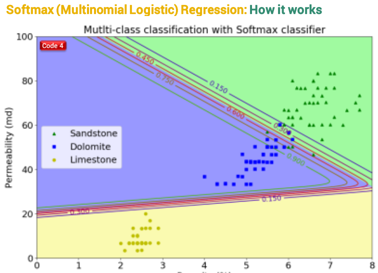
Softmax Regression classifier prediction

$$\hat{y} = \underset{k}{argmax}(\sigma(s(x))_k) = \underset{k}{argmax}(s_k(x)) = \underset{k}{argmax}(\beta_k^T . \mathbf{x})$$

14

15

Softmax (Multinomial Logistic) Regression: How it works



Notice that the decision boundaries between any two classes are linear

The point where all decision boundaries meet, all classes have an equal estimated probability of 33%

? Any recommendation to improve codes 2& 3?

15

16