W207 – Applied Machine Learning

Cornelia Paulik, PhD

School of Information
UC Berkeley

Logistic Regression – multiclass

Announcements

- Final project:

  - you can start working on it (I approved your proposals);

  - if you proposed multiple projects: go to the project doc and delete those you won't be working on;

  - make sure I am added as a contributor to your GitHub repo.

Announcements

- Midterm:

  - see bCourses or my website on how to prepare for it;

  - multiple choice or open-ended questions;

  - don’t wait until the last minute to prepare for the exam.

Announcements

- Assignment 5:

  - start early

  - it’s an open-ended assignment: chocolate dataset;

  - at the minimum follow the questions in the assignment;

  - if you want to do more than that, go for it!

Announcements

- Assignments (general thoughts):

  - the hyperparameter tuning exercises are pretty much open-ended;

  - if you want to try new tuning methods, go for it!

Announcements

- Women in MIDS

  - Slack channel: women-in-mids;

  - This Friday (June 7) @ 1 pm PT: #IAmRemarkable workshop

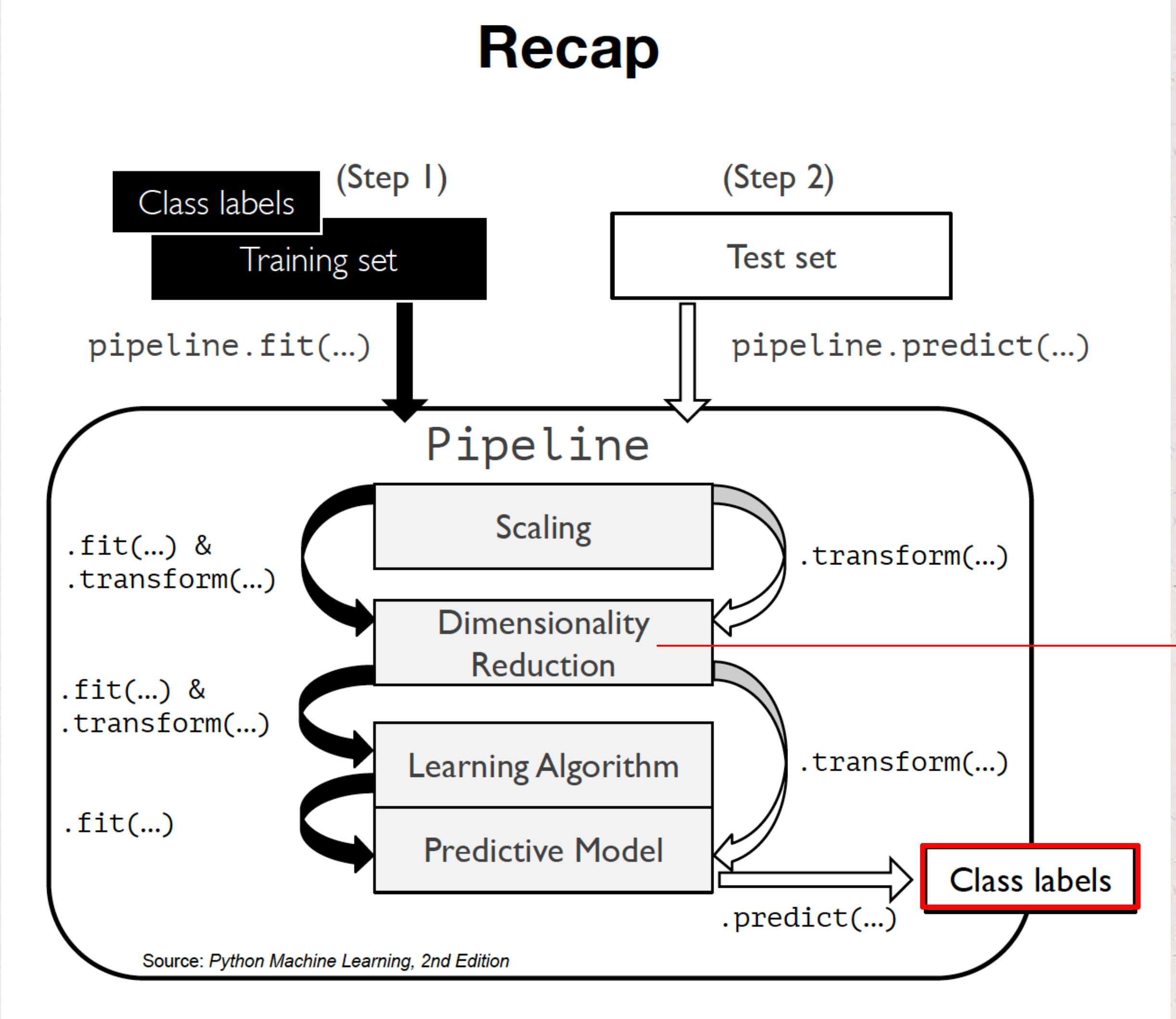
Last week

- Binary Logistic Regression and Gradient Descent
- Predict a binary outcome variable using the wine dataset.
- Breakout room exercise: diabetes dataset.

Today’s learning objectives

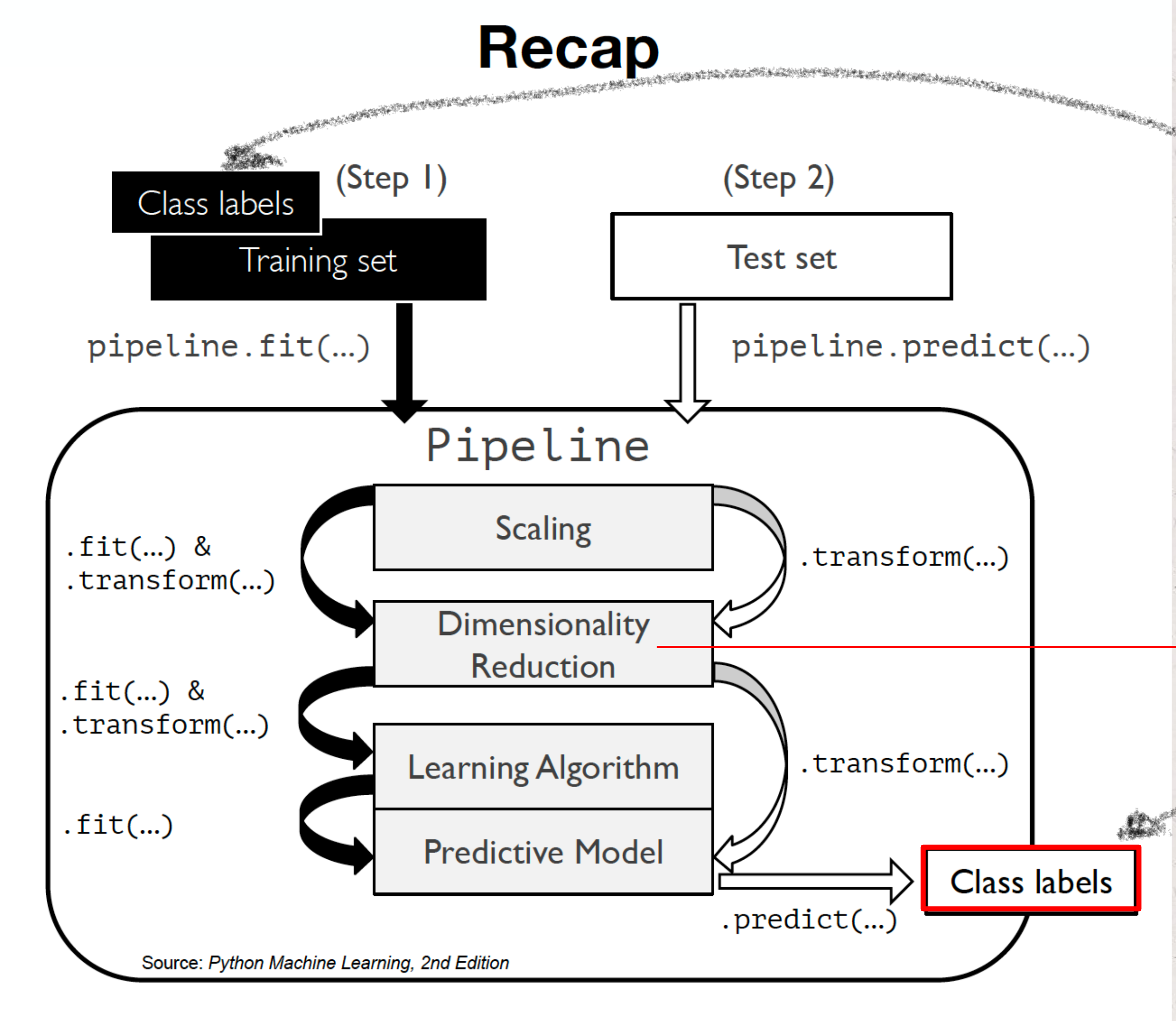
- Multi-class Logistic Regression (extend the wine dataset to 3 classes)
- Evaluation metrics for classification tasks
- Dealing with class imbalance
- Breakout room exercise: wine dataset, compute ROC AUC

Evaluation metrics



Will do this in week 08

Evaluation metrics



How do they compare?

Will do this in week 08

Evaluation metrics – confusion matrix

Predicted label

Actual label	+	−
+	**TP** True Positives	**FN** False Negatives Type II error
−	**FP** False Positives Type I error	**TN** True Negatives

Binary classification

Predicted label

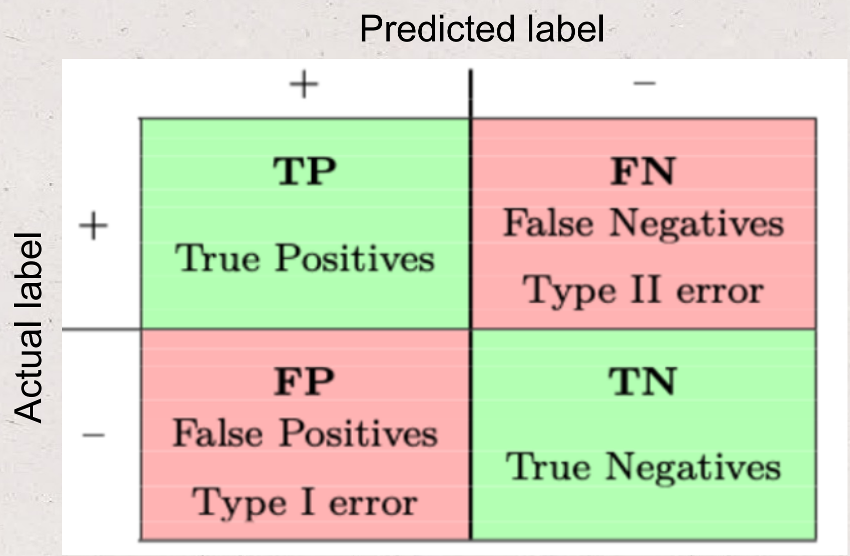
Actual label	Class 0	Class 1	Class 2
Class 0	T(0,0)		
Class 1		T(1,1)	
Class 2			T(2,2)

Multi-class classification

Confusion matrix: traditionally for binary class problems but can easily generalize it to multi-class settings

Source: [here]() and [here]()

Evaluation metrics – Accuracy



Binary classification

Metric	Formula	Interpretation
$\boxed{\text{Accuracy}}$	$\frac{TP + TN}{TP + TN + FP + FN}$	Overall performance of model
Precision	$\frac{TP}{TP + FP}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{TP}{TP + FN}$	Coverage of actual positive sample
Specificity	$\frac{TN}{TN + FP}$	Coverage of actual negative sample
F1 score	$\frac{2TP}{2TP + FP + FN}$	Hybrid metric useful for unbalanced classes

Not good if data is imbalanced

Source: here and here

Evaluation metrics – Precision vs. Recall

Predicted label

Actual label	+	−
+	**TP** True Positives	**FN** False Negatives Type II error
−	**FP** False Positives Type I error	**TN** True Negatives

Binary classification

Metric	Formula	Interpretation
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Overall performance of model
Precision	$\frac{TP}{TP + FP}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{TP}{TP + FN}$ Denominator: P (# of positives)	Coverage of actual positive sample
Specificity	$\frac{TN}{TN + FP}$ Denominator: N (# of negatives)	Coverage of actual negative sample
F1 score	$\frac{2TP}{2TP + FP + FN}$	Hybrid metric useful for unbalanced classes

<u>Precision:</u> if focus is spam classification (don't want to label emails as spam if not very confident)
<u>Recall:</u> if focus is to identify patients with cancer.

Source: here and here

Evaluation metrices – Sensitivity vs. Specificity

Predicted label

Actual label	+	−
+	**TP** True Positives	**FN** False Negatives Type II error
−	**FP** False Positives Type I error	**TN** True Negatives

Binary classification

Metric	Formula	Interpretation
Accuracy	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$	Overall performance of model
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$ Denominator: P (# of positives)	Coverage of actual positive sample
Specificity	$\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$ Denominator: N (# of negatives)	Coverage of actual negative sample
F1 score	$\frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$	Hybrid metric useful for unbalanced classes

<u>Sensitivity</u>: recovery rate of the Positives
<u>Specificity</u>: recovery rate of the Negatives

Source: here and here

Evaluation metrics – F1 score

Predicted label

Actual label	+	−
+	**TP** True Positives	**FN** False Negatives Type II error
−	**FP** False Positives Type I error	**TN** True Negatives

Binary classification

Metric	**Formula**	**Interpretation**
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Overall performance of model
Precision	$\frac{TP}{TP + FP}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{TP}{TP + FN}$ Denominator: P (# of positives)	Coverage of actual positive sample
Specificity	$\frac{TN}{TN + FP}$ Denominator: N (# of negatives)	Coverage of actual negative sample
$\boxed{\text{F1 score}}$	$\frac{2TP}{2TP + FP + FN}$	Hybrid metric useful for unbalanced classes

provides a balance (harmonic mean) between the precision and recall metrics

Source: here and here

Evaluation metrics – TPR vs. FPR

Predicted label

Actual label	+	−
+	**TP** True Positives	**FN** False Negatives Type II error
−	**FP** False Positives Type I error	**TN** True Negatives

Binary classification

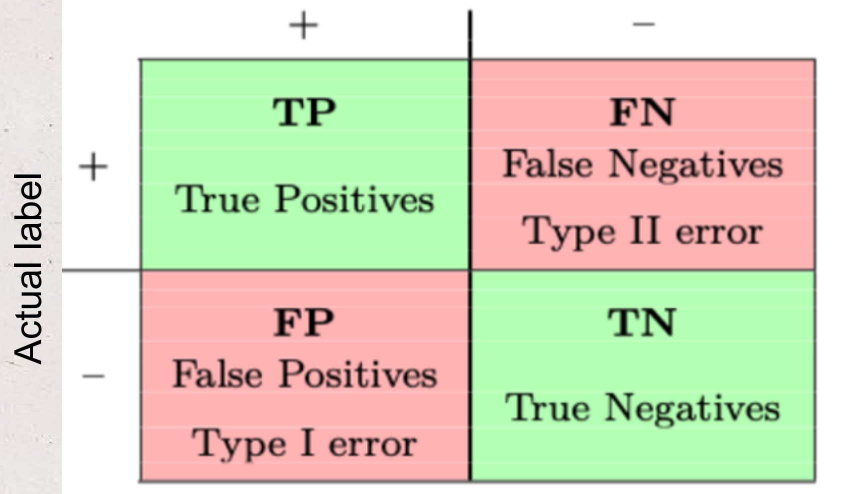
Predicted label

Metric	Formula	Equivalent
True Positive Rate TPR	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$ Denominator: P (# of positives)	Recall, sensitivity
False Positive Rate FPR	$\frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}}$ Denominator: N (# of negatives)	1-specificity

Source: here and here

Evaluation metrics – TPR vs. FPR (ROC)

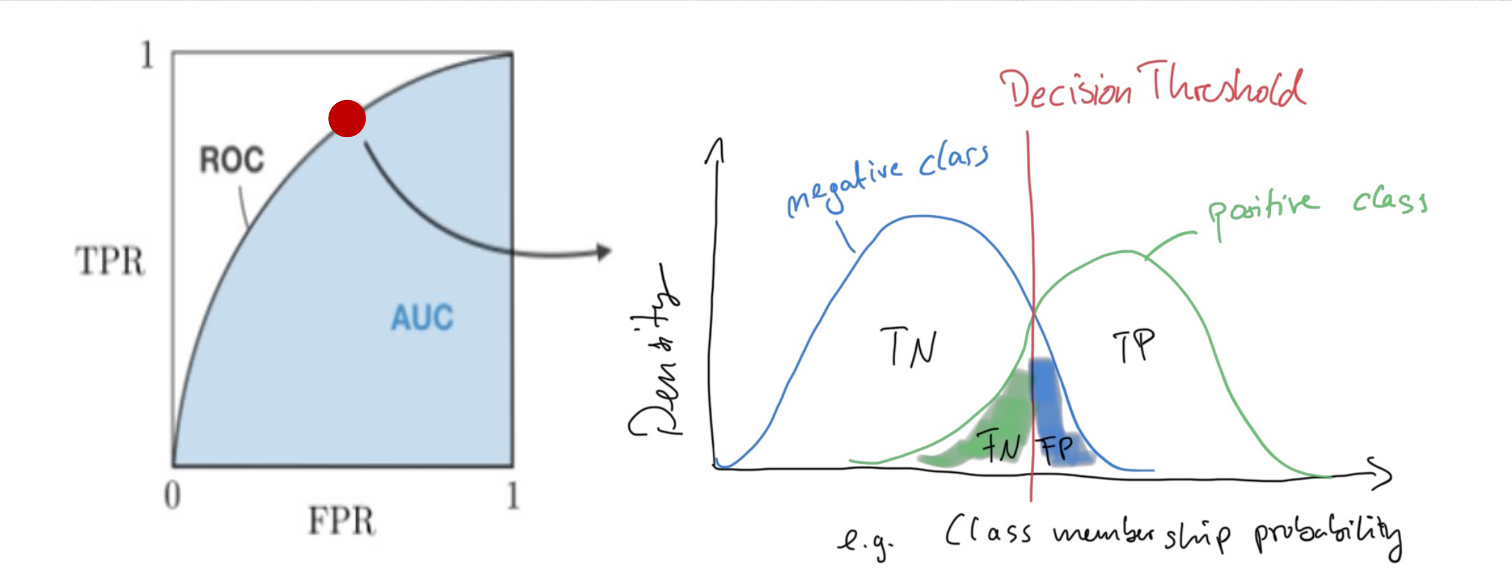
Predicted label



Binary classification

Predicted label

Metric	Formula	Equivalent
True Positive Rate TPR	$\frac{TP}{TP+FN}$ Denominator: P (# of positives)	Recall, sensitivity
False Positive Rate FPR	$\frac{FP}{TN+FP}$ Denominator: N (# of negatives)	1-specificity



Source: [here]() and [here]()

Evaluation metrics – TPR vs. FPR (ROC AUC)

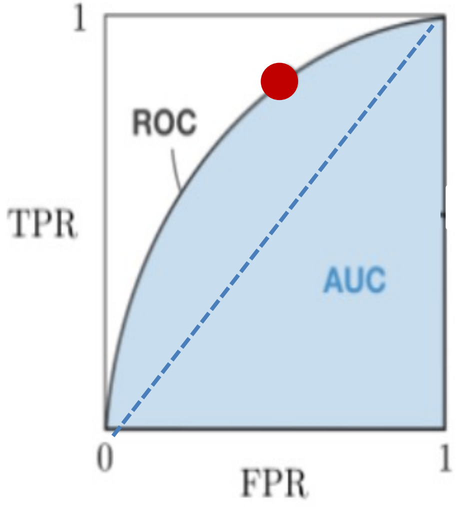
Predicted label

Actual label	+	−
+	**TP** True Positives	**FN** False Negatives Type II error
−	**FP** False Positives Type I error	**TN** True Negatives

Binary classification

Predicted label

Metric	**Formula**	**Equivalent**
True Positive Rate TPR	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$ Denominator: P (# of positives)	Recall, sensitivity
False Positive Rate FPR	$\frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}}$ Denominator: N (# of negatives)	1-specificity



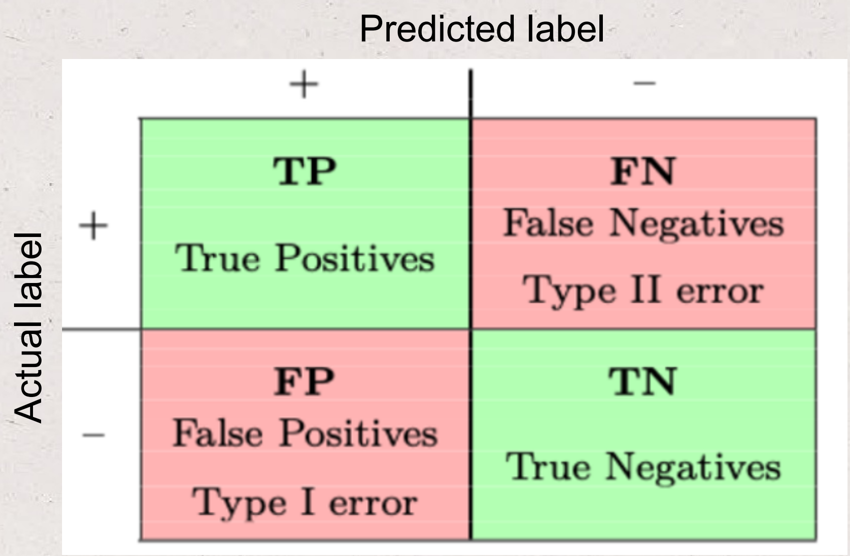
ROC AUC: Area under the Curve

The higher the AUC value for a classifier, the better its ability to distinguish between positive and negative classes.

When **AUC=0.5**, then the classifier is not able to distinguish between Positive and Negative class points. The classifier is **predicting random class (dashed line)**

Source: here and here

Evaluation metrics – Matthew's correlation coef



Binary classification

- Useful for unbalanced classification settings
- MCC is a specific case of a linear correlation coefficient (Pearson r) for a binary classification setting
- The previous metrics take values in the range between 0 (worst) and 1 (best)
- The MCC is bounded between the range 1 (perfect correlation between ground truth and predicted outcome) and -1 (inverse or negative correlation) — a value of 0 denotes a random prediction.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Source: here and here

Dealing with class imbalance

- Important to recognize the problem!
- Imagine a dataset with 90 % healthy patients (class 0), 10% unhealthy (class 1). You don't need a ML algorithm, assign class 0 to all examples -> accuracy =90%
- Upsample the minority class, downsample the majority class, generate synthetic training examples (SMOTE)
- If you fit a model, class imbalance will influence the learning during the fitting stage. You optimize a reward/cost function -> decision rule likely to be biased towards the majority class. Assign a larger penalty to wrong predictions on the minority class during model fitting
- Focus on other metrics than accuracy (see previous slides)

Dealing with class imbalance

- TensorFlow example (highly recommended):
  https://www.tensorflow.org/tutorials/structured_data/imbalanced_data

Multiclass Logistic Regression

Q1: What are the main differences between binary and multi-class logistic regression?