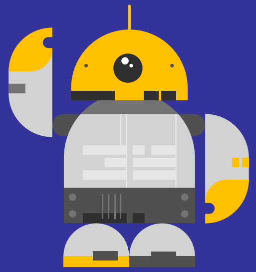


Database Systems

Lecture 9 – Cont. Machine Learning for Data Analytics

Dr. T. Akilan
takilan@lakeheadu.ca

Welcome back

- ❑ Project Stage 2 – Investigation
- ❑ Assignment 1

This Session

- Taking the missed cosine similarity

- Quick recap on Regression
  - Linear regression
  - Goodness-of-Fit - R^2

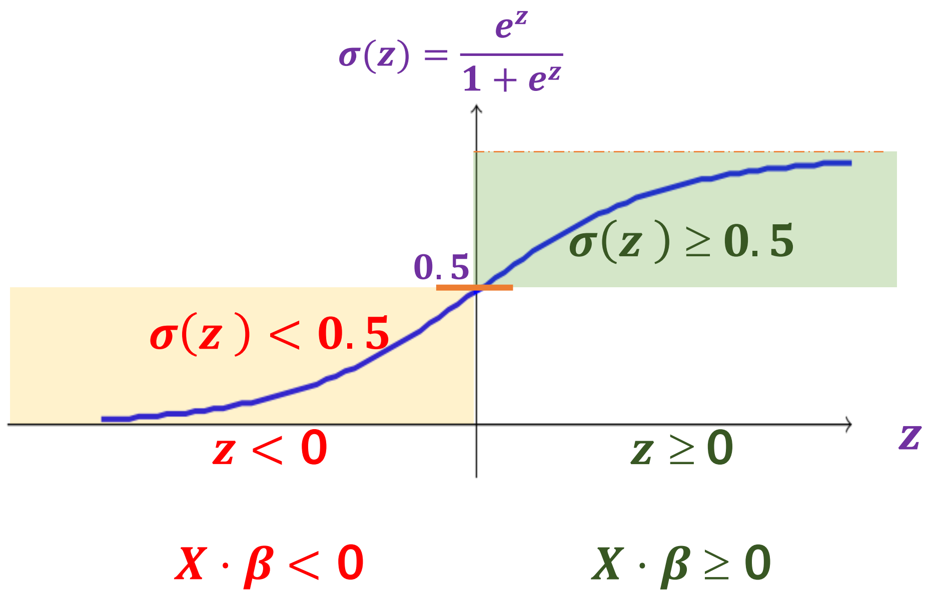
- Classification
  - Logistic regression (recap)
  - Maximum likelihood estimation

- Evaluation metrics
  - ROC
  - AUC

Logistic Regression - Recap

- It uses sigmoid or logistic function to represent the occurrence of an event.

- E.g., input variables: $X = \{x_1, x_2, \ldots, x_p\}$
  occurrence of an event: **y**



○ Probability of the event, y:

$$p(y|X;\beta\}) = \sigma(z) = \frac{e^z}{1+e^z},$$

z - a linear function of the independent predictors of X.

$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{p-1} x_{p-1} \cong X \cdot \beta.$

β_i's – a set of parameters

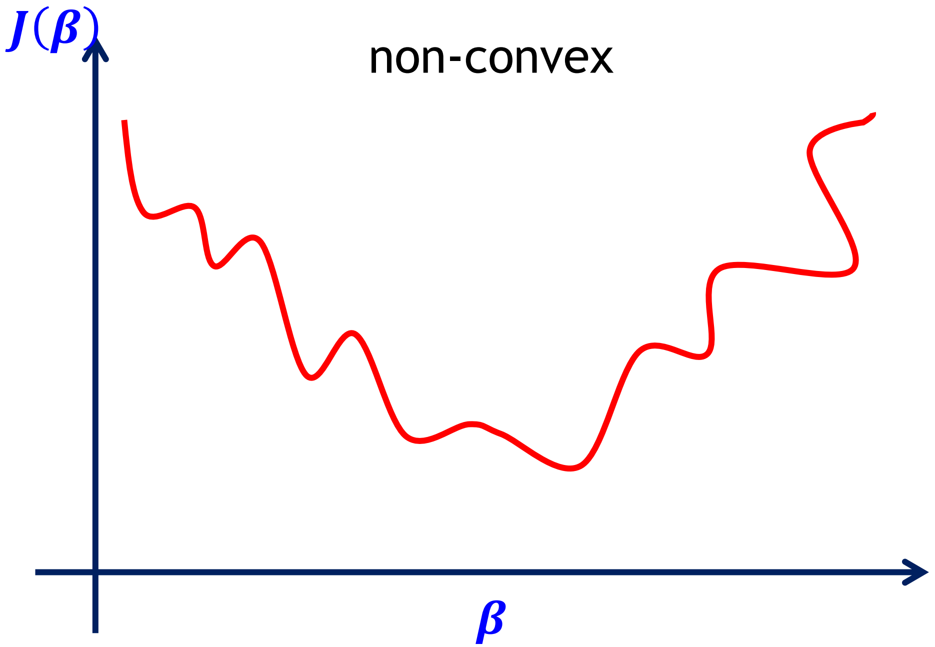
- **Advantage over linear regression**:
  - $0 \leq \sigma(z) \leq 1$

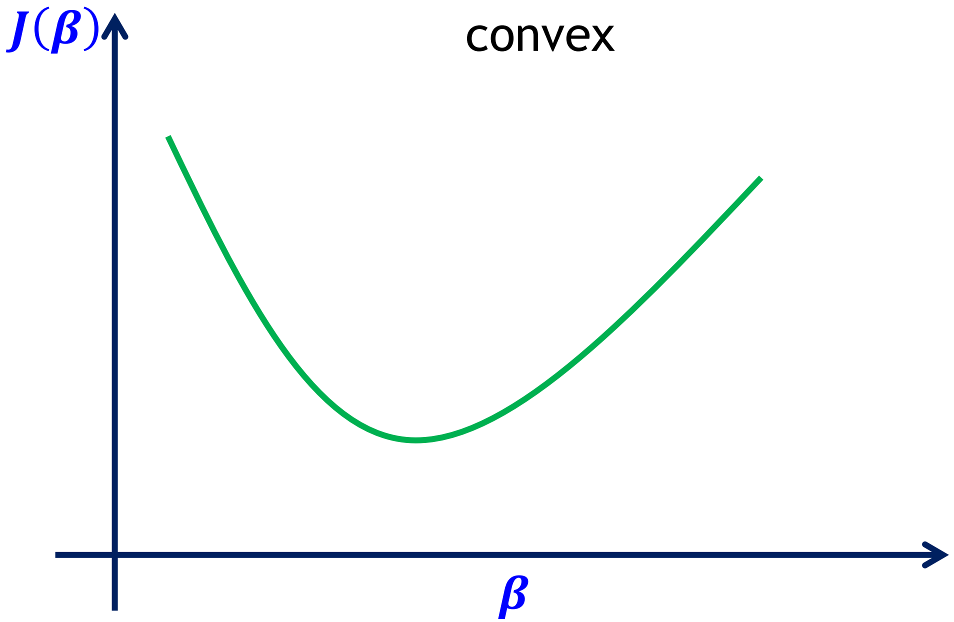
- **How do we estimate the best parameter β?**
  - Consider an objective function, $J(\beta)$
  - Apply an optimizer (min or max) accordingly

Logistic Regression – Objective Function

- From linear regression what we know: $J(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\big[\overbrace{\boldsymbol{h}_{\boldsymbol{\beta}}(\boldsymbol{x}_i)}^{\text{model}} - y_i\big]^2$

- Change the **model** to sigmoid function: $J(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{e^{z}}{1+e^{z}} - y_i\right]^2$; $\boldsymbol{z} = \boldsymbol{h}_{\boldsymbol{\beta}}(\boldsymbol{x}_i)$





Logistic Regression: Maximum Likelihood Estimator

- The sigmoid classifier is fit through **learning** the best values for the parameters $\boldsymbol{\beta}$ by **maximizing** the **log** (joint) **conditional likelihood** probabilities of the two classes.

- Given training sample $\langle \boldsymbol{x_i} | \boldsymbol{y_i} \rangle$ and assume $\boldsymbol{y}$ can only take two values of $\mathbf{0}$ or $\mathbf{1}$, the log conditional likelihood is:

  $\boldsymbol{log(p_i)} \quad \leftarrow$ if $\boldsymbol{y_i = 1}$ and

  $\boldsymbol{log(1 - p_i)} \leftarrow$ if $\boldsymbol{y_i = 0}$.

  Note: $p_i = p(y = 1 | x_i ; \beta)$, i.e., probability function of y=1 given x_i parameterized by β.

- Then total log conditional likelihood (LCL):

$$LCL = \sum_{i:y_i=1} \log p_i + \sum_{i:y_i=0} \log(1 - p_i)$$

sum of the log conditional likelihood, by grouping together, the positive and negative training samples

Logistic Regression: **MLE Cont.**

- Unifying the individual class likelihood ($\boldsymbol{l}$):

$$\boldsymbol{l(\beta) = \sum_{i=1}^{n} y_i \cdot log\big(p(x_i)\big) + (1 - y_i) \cdot log(1 - p(x_i))}$$

Note: it is equivalent to previous eq. since, if $\boldsymbol{y_i}$ = 1 (true), then $\boldsymbol{1 - y_i}$ = 0

- Now, let's substitute the expression for $\boldsymbol{p(y|x;\ \beta)}$:

$$\boldsymbol{l(\beta) = \sum_{i=1}^{n} y_i \cdot log}\left(\frac{1}{1 + e^{-\beta_i X_i}}\right) \boldsymbol{+ (1 - y_i) \cdot log}\left(\boldsymbol{1 -} \frac{1}{1 + e^{-\beta_i X_i}}\right)$$

$$\boldsymbol{l(\beta) = \sum_{i=1}^{n} y_i \cdot log}\left(\frac{1}{1 + e^{-\beta_i X_i}}\right) \boldsymbol{+ (1 - y_i) \cdot log}\left(\frac{e^{-\beta_i X_i}}{1 + e^{-\beta_i X_i}}\right)$$

Why Logistic Regression: MLE Cont.

- $l(\beta) = \sum_{i=1}^{n} y_i \cdot log\left(\frac{1}{1+e^{-\beta_i X_i}}\right) + (1 - y_i) \cdot log\left(\frac{e^{-\beta_i X_i}}{1+e^{-\beta_i X_i}}\right)$ ← from previous slide

- Now, take y_i common term:

$$l(\beta) = \sum_{i=1}^{n} y_i \left[log\left(\frac{1}{1 + e^{-\beta_i X_i}}\right) - log\left(\frac{e^{-\beta_i X_i}}{1 + e^{-\beta_i X_i}}\right)\right] + log\left(\frac{e^{-\beta_i X_i}}{1 + e^{-\beta_i X_i}}\right)$$

- Further simplify:

$$l(\beta) = \sum_{i=1}^{n} y_i\left[log\left(e^{\beta_i X_i}\right)\right] + log\left(\frac{e^{-\beta_i X_i}}{1+e^{-\beta_i X_i}}\right) = \sum_{i=1}^{n} y_i \beta X_i + log\left(\frac{1}{1+e^{\beta_i X_i}}\right)$$

$$\boxed{l(\beta) = \sum_{i=1}^{n} y_i \beta X_i - log\left(1 + e^{\beta_i X_i}\right)}$$

- **Optimal $\beta'_j s$:** Find via maximizing the objective function → $\max_{\beta} l(\beta)$

Logistic Regression: Maximum Likelihood Estimator Cont.

- $\boldsymbol{l(\beta)} = \sum_{i=1}^{n} y_i \beta X_i - \boldsymbol{log}\left(1 + e^{\beta_i X_i}\right)$

- Now, we choose values of β that make this equation as large as possible: $\boldsymbol{\beta} = \underset{\boldsymbol{\beta}}{\arg\max}\, \boldsymbol{l(\beta)}$

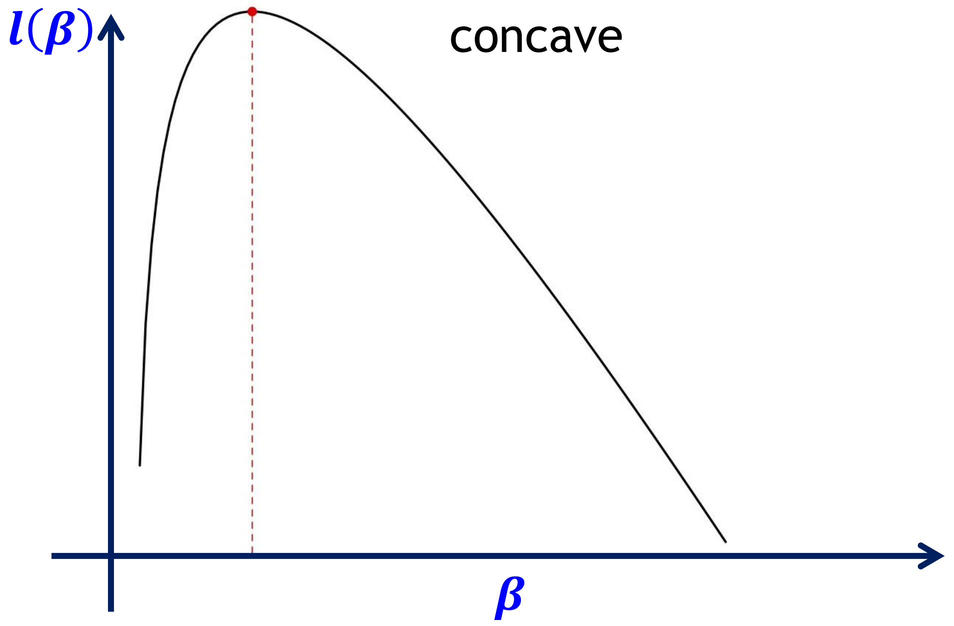
- Maximizing involves derivatives over multiple iterations.

- E.g., stochastic gradient **ascent**:

$$\beta_j := \beta_j + \lambda \frac{\partial}{\partial \beta_j} LCL$$

$\boldsymbol{l(\beta)}$

  - The gradient-based update of the parameters
  - Slightly changes the parameter values to increase the log likelihood based on one example at a time.



Logistic Regression: Diagnostics

- **What we know:**
  - **Sigmoid classifier** is to assign class labels based on the **predicted probability,** $\sigma(z)$.
  - E.g., a customer can be classified with the label called "**Churn**" if $\sigma(z) \geq \tau$ (a high probability).
  - Otherwise, i.e., $\sigma(z) < \tau$ a "**Remain**" label is assigned to the customer.
  - Generally, τ **= 0.5** is used as the **default threshold** to distinguish between any two class labels.
- **Application specific** τ**:**
  - to **avoid false positives** (e.g., predict *Churn* when actually the customer will *Remain*)
  - to **avoid false negatives** (e.g., predict *Remain* when the customer will actually *Churn*).
- **How do we set an application specific** τ**:**
  - Using ROC graph, we can find it.
  - A ROC graph is a 2D plot that **summarizes a classifier performance over various threshold values** with **false positive rate** on the x axis **against true positive rate** on the y axis

Receiver Operating Characteristic (ROC) Curve

- Let, binary class labels: C and !C, where "!C" denotes "not C"

		$\hat{Y}$	
		C	!C
Y	C	TP	FN
	!C	FP	TN

TP FN = # of actual Positives

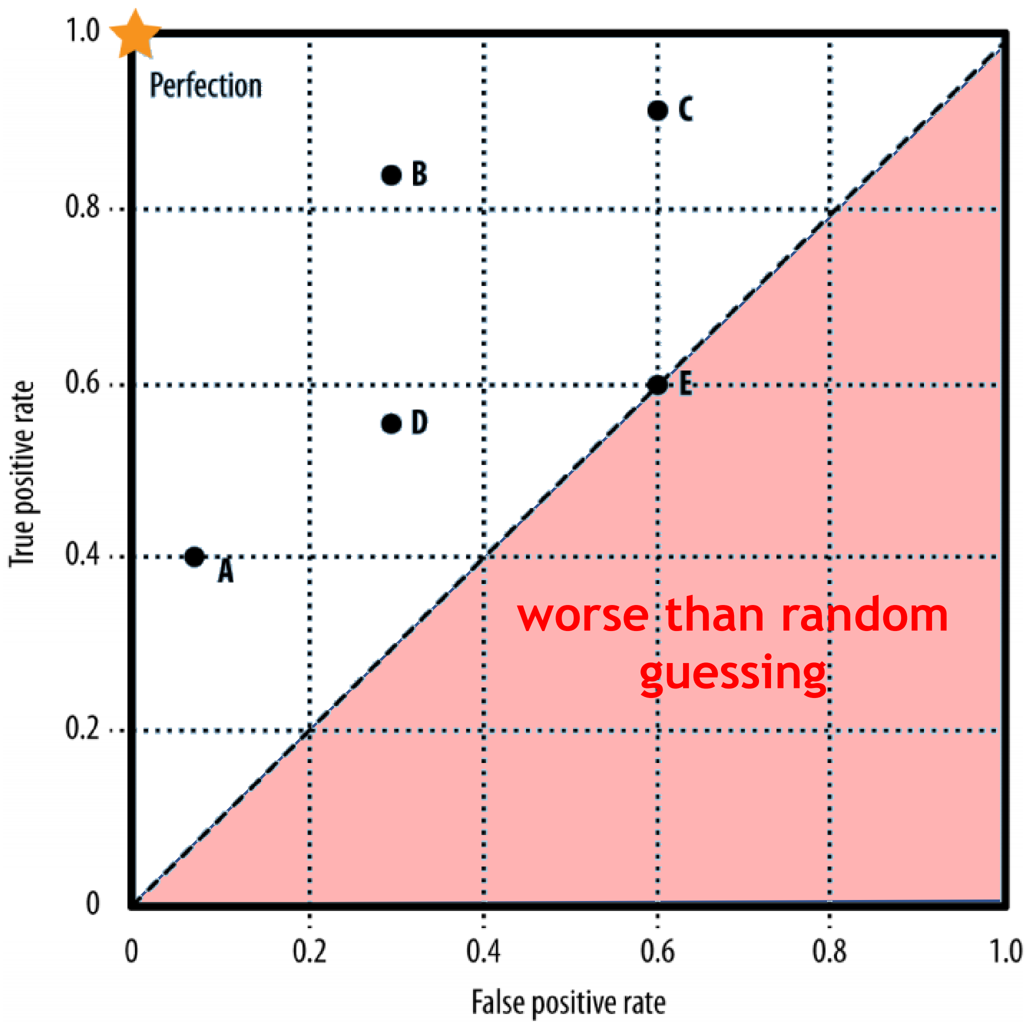
FP TN = # of actual Negatives

- $True\ Positive\ Rate\ (TPR) = \frac{\#\ of\ true\ positives}{\#\ of\ actual\ positives}$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} = 1 - \mathrm{FNR}$$

- $False\ Positive\ Rate\ (FPR) = \frac{\#\ of\ false\ positives}{\#\ of\ actual\ negatives}$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} = 1 - \mathrm{TNR}$$



- Scores were generated by $\sigma(z)$ on a testing set consist of **100** positives (p) and **100** negatives (n) samples.
  - True class: {p, n}
  - Prediction: {Y, N}

- Pick a generated score as threshold and compute:
  - Confusion matrix
  - FPR = $\frac{\text{FP}}{\text{N}}$
  - TPR = $\frac{\text{TP}}{\text{P}}$

- Repeat

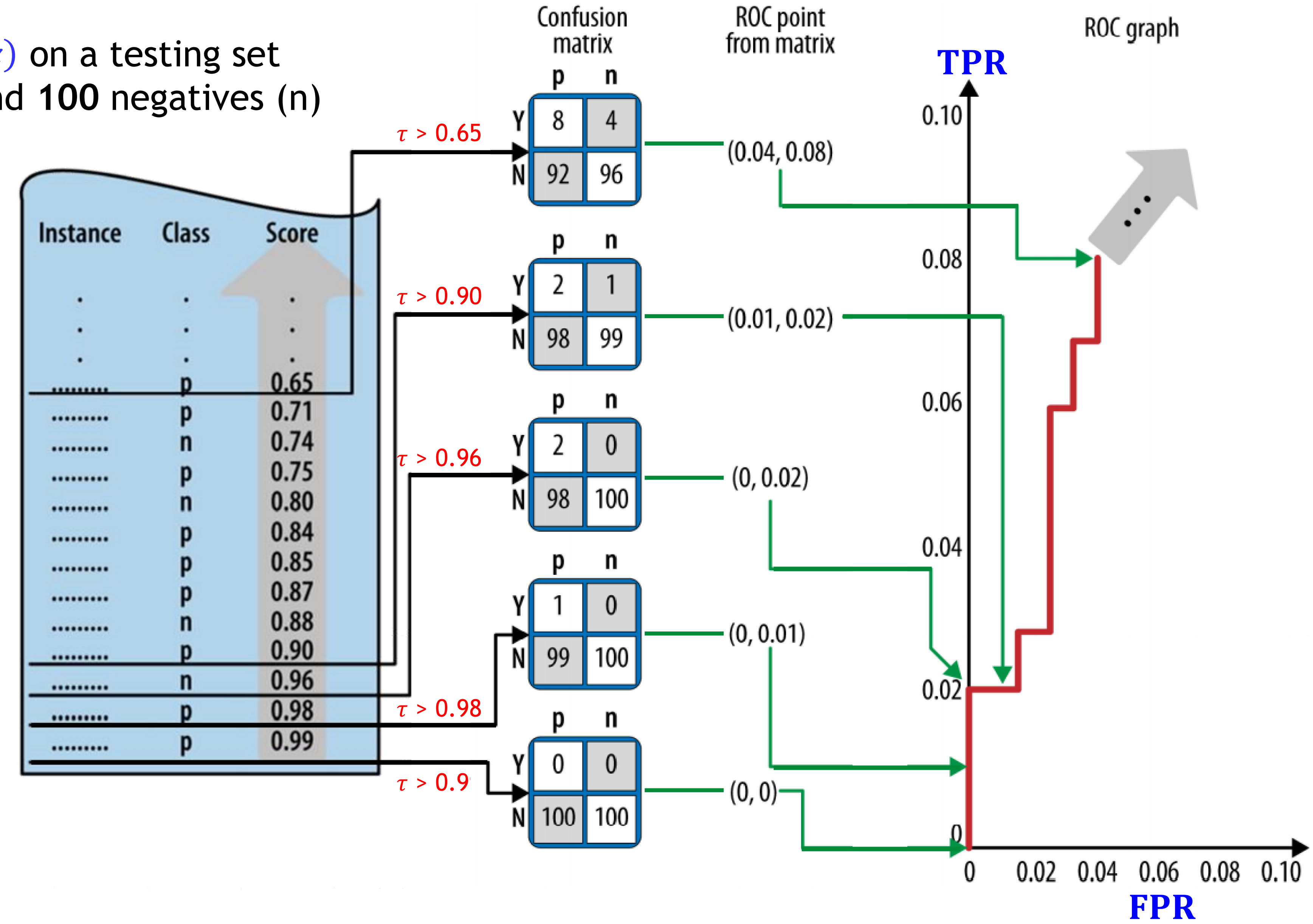
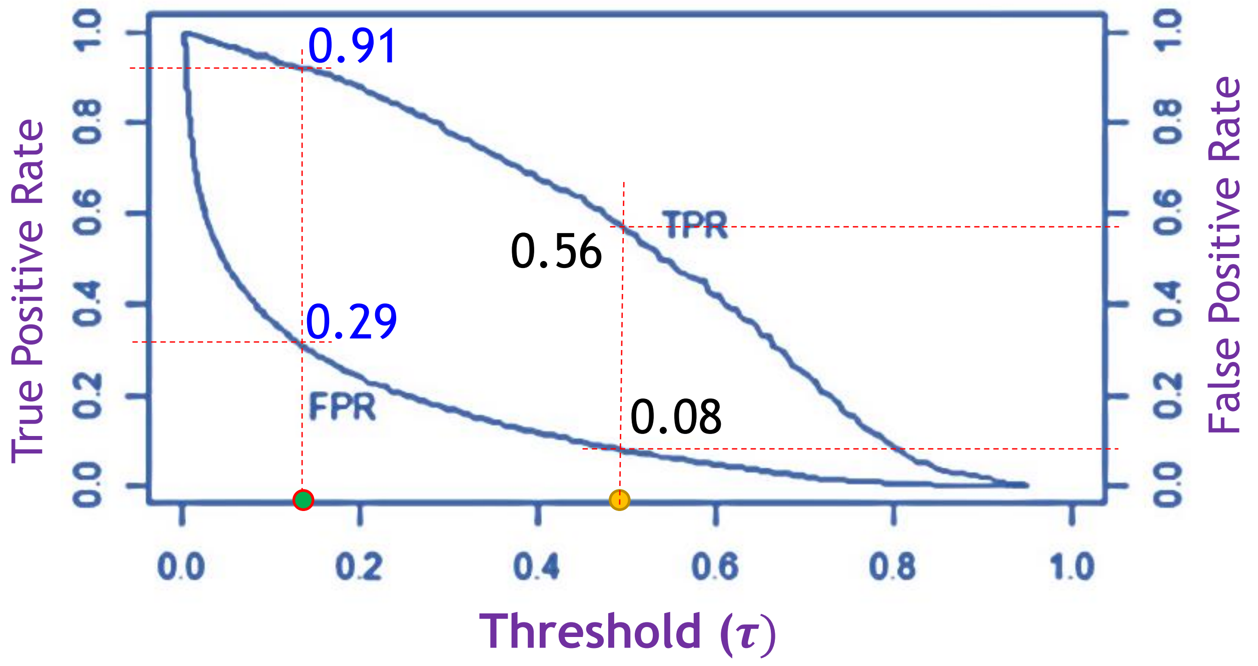


Image source - [1]

Logistic Regression: Diagnostics – Setting the Threshold

- **Default setting: τ** = 0.5
  - TPR = 0.56 and a FPR = 0.08.
    - ✓ 56% of customers who will churn are correctly classified with 'Churn'
    - ✓ 8% of the customers who will remain as customers are improperly labeled as 'Churn'.



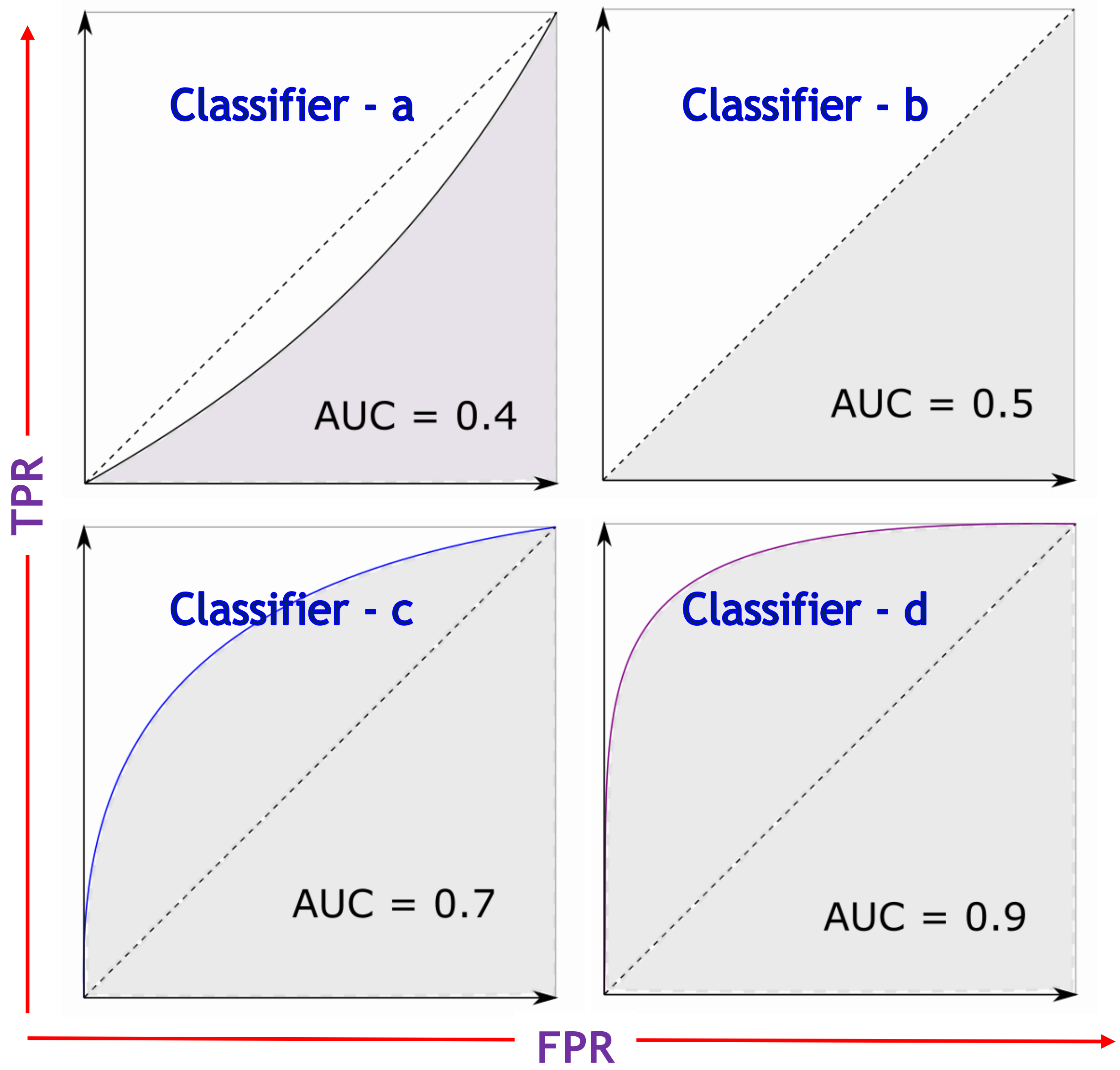
- **Is it good enough?**
  - What is the purpose?
  - Do you want to take a proactive step to stop churning customers?
  - Need to correctly classify more churning customers, then the τ **shoud be lowered**.
  - E.g., **application-specific setting: τ** = 0.15
    - ✓ TPR = **0.91**and FPR = **0.29**
    - ✓ **91%** of the customers who will churn are **correctly identified**,
    - ✓ Cost - **misclassifying 29%** of the customers who will remain are classified as 'Churn'

Logistic Regression: Diagnostics – Compute AUC on ROC Graph

- AUC - a useful metric that computes the **area under** the **ROC** graph to pick the best model for the same classification problem (E.g., logistic regression vs random forest)

- A preferred classifier:
  - to have a **low FPR** and a **high TPR**.
  - Going from **left to right** on the **FPR** axis, an appropriate classifier should have the **TPR rapidly approach values close to 1**, with only a small change in FPR (δ_{fpr}).
  - The closer the **curve tracks** along the **vertical axis** and approaches the **upper-left** hand of the **plot**, i.e., near the point (**0**,**1**), the **better** the model/classifier.

E.g., **Classifier - d**



Logistic Regression: AUC for Classifier Comparison

- **AUC is useful when a single number** is needed to summarize performance of the classifiers.

- **Skewed data:** using accuracy as a **single performance metric is a poor choice** and misleading
- E.g., a dataset that has 93% negative and 7% positive samples.

Model	Accuracy (%)	AUC
Classification Tree	91.8 ± 0.0	0.614 ± 0.014
Logistic Regression	93.0 ± 0.1	0.574 ± 0.023
k-Nearest Neighbor	93.0 ± 0.0	0.537 ± 0.015
Naive Bayes	76.5 ± 0.6	0.632 ± 0.019

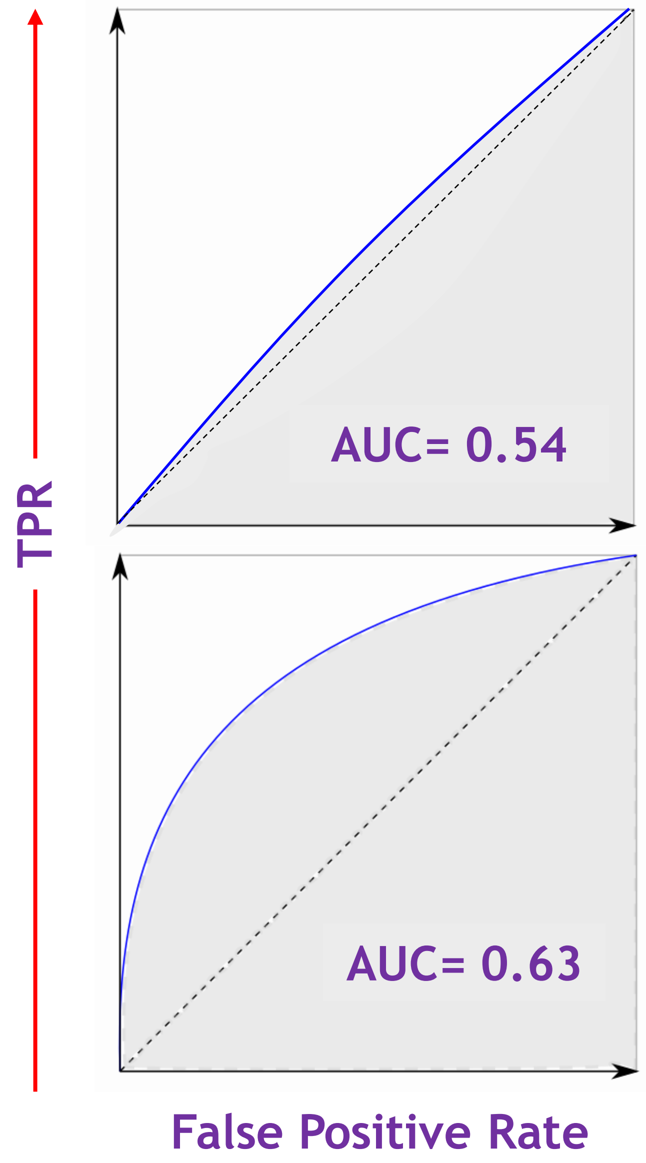
k-NN

	p	n
Y	3 (0%)	15 (0%)
N	324 (7%)	4351 (93%)

Naive Bayes

	p	n
Y	127 (3%)	848 (18%)
N	200 (4%)	3518 (75%)

- $$Accuracy = \frac{TP + TN}{P + N}$$ $$\text{FPR} = \frac{\text{FP}}{\text{N}} \quad ; \quad \text{TPR} = \frac{\text{TP}}{\text{P}}$$



Summary

Linear regression

- **Nature:** The outcome variable is a **continuous unbounded** value:

$$-\infty \leq h_{\beta}(x_i) \leq +\infty$$

- **Application:** Predicting real value of a response variable. E.g., modeling the relationship between age and education to income.
- **Objective function:** Ordinary least square (OLS)
- **Optimizer:** Gradient Decent

Logistic regression

- **Nature:** The outcome variable is **continuous bounded** value:

$$0 \leq \sigma(z) \leq 1$$

- **Application:** Predict the **likelihood** of an **outcome** based on the input variables. E.g., financial status identification, like **wealthy** or **poor, b**ased on a person's income.
- **Objective function:** log conditional likelihood (LCL)
- **Optimizer:** maximum likelihood estimator (MLE), like Gradient Ascent.

Assignment 1

- Let’s move on to assignment 1 discussion

References

[1] Data Science for Business, by Foster Provost and Tom Fawcett, First Edition, ISBN: 978-1-449-36132-7

[2] Jiawei Han, Micheline Kamber, & Jian Pei, *Data Mining: Concepts and Techniques*, 3rd Edition, Morgan Kaufmann, ISBN: 978-0-12-381479-1