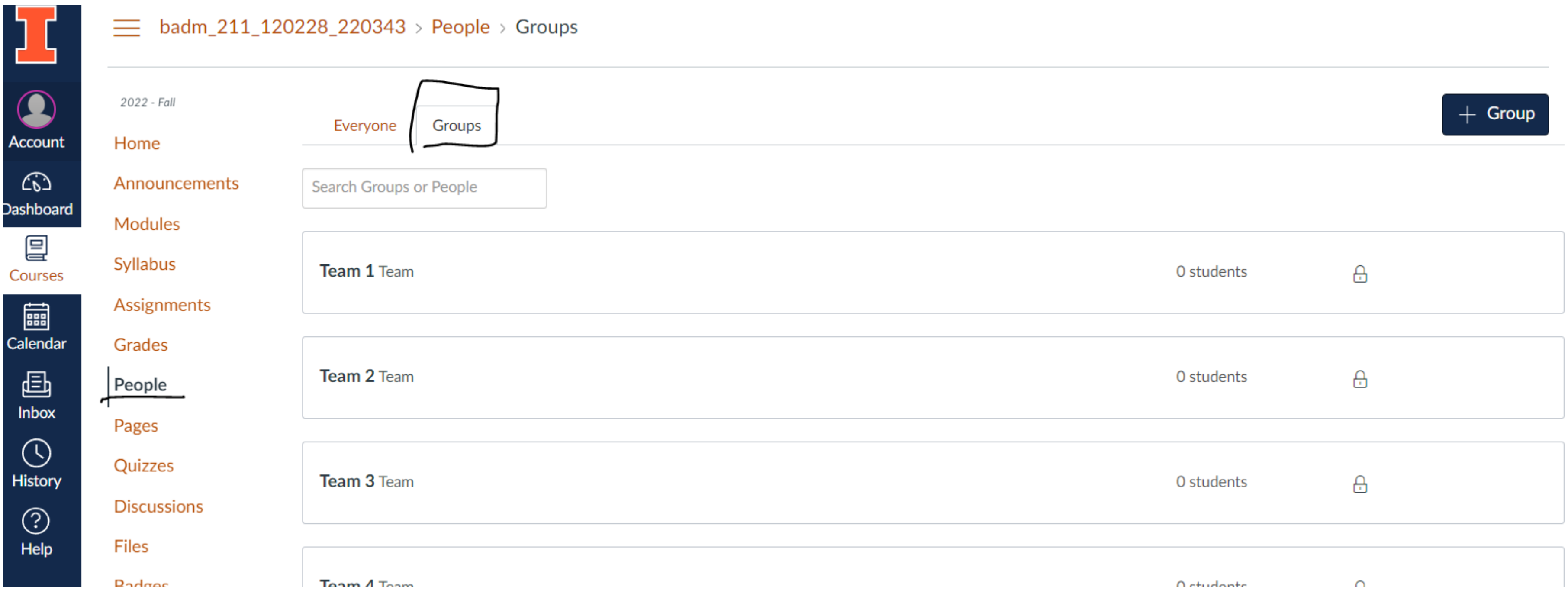
Business Analytic II – Classification and Logistic Regression

Prof. Zilong Liu | BADM 211

Topic 8A

Reminders – Final Project Group

- Form Teams for Final Project:
  - Self-enroll into a Team using Canvas. 4-5 persons per team
  - Go to People -> Groups -> Sign Up (Please pick a team number together and make sure you sign up for the right group number. Sign up at the same time as your team members).
  - **Section B(M/W class) use teams 1 to 10,**
  - **section E (T/Th Class) please use teams 11 – 20**
  - If not self-sign-up by Apr 9th, I will randomly assign you to some group needing person.



Extra Credit – Learning and Sharing

Extra Credit Assignment 2% of Total

Extra Credit - Python/Data Science Insight Sharing

Due Apr 30 at 11:59pm | 20 pts

Today’s Agenda

- **Modelling for categorical variable**
- In-Class Assignment
- Q&A

Overview

Last time(s)

What method can be used to predict continuous variables?

- Conceptual soundness of linear regression
- Estimation method (Optimization function)
- Results interpretation
- Performance metrics

How to build a linear regression model in Python?

- Data split
- Model Estimation/Performance Metrics

What we learn today

What method can be used to predict categorical variables?

How to build the model in Python?

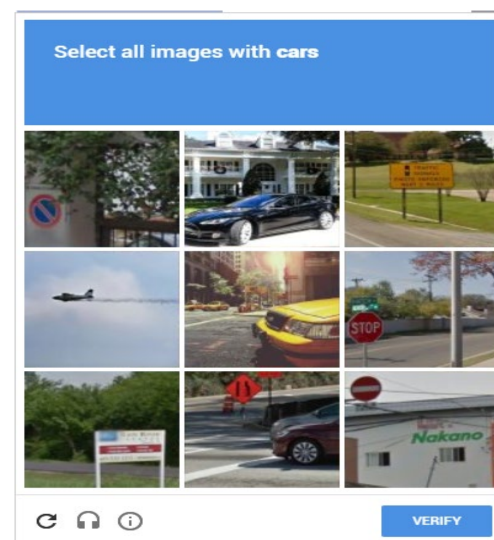
Categorical variables - Why does it matter?

Should a bank give a person a loan or not?

Whether a person get an infectious disease or not?

When we use google Colab, google asks us to identify cars in image….

…



Can we use linear regression model?

Example - Fraudulent Transaction Detection

Problem
- Detect fraudulent credit card transaction to prevent financial loss

Solution
- Build a quantitative model to predict which transaction is more likely to be fraudulent one

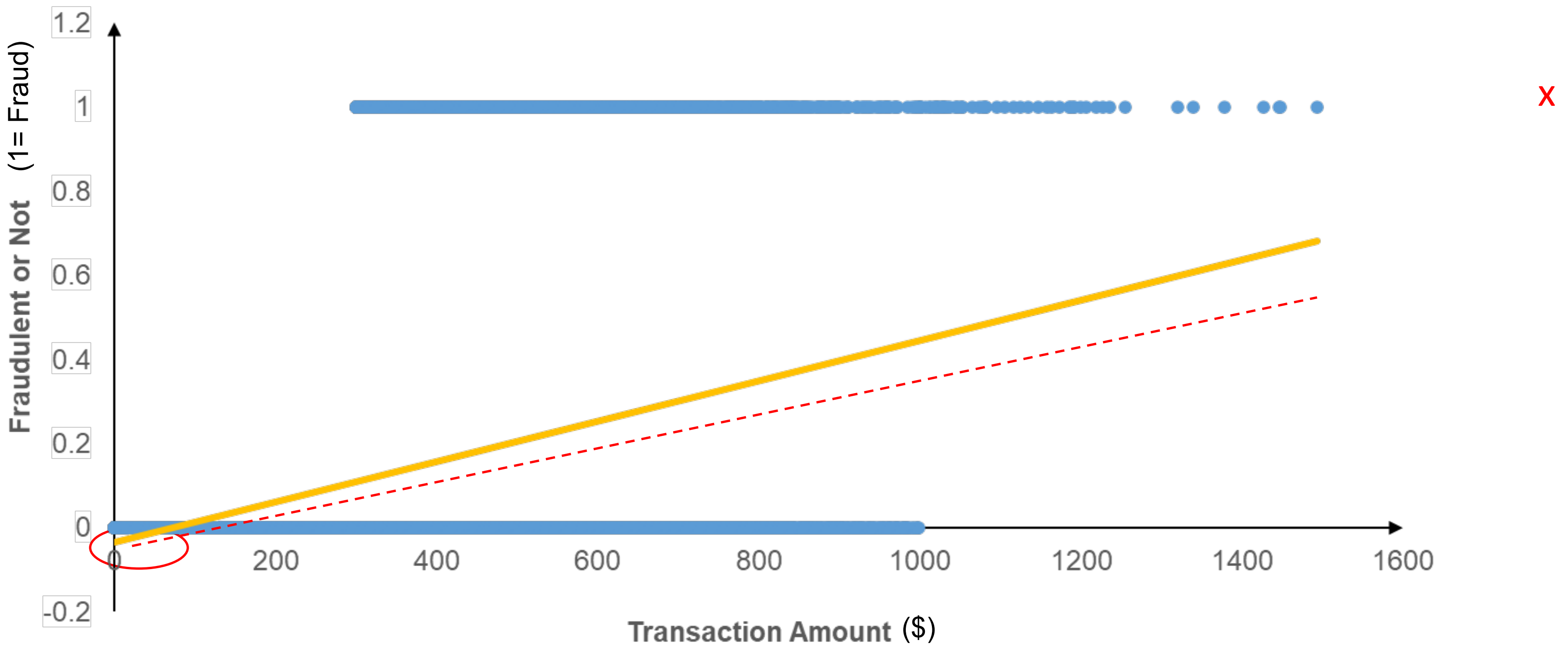
Data
- Each row represents a transaction record
- The last column is the fraud indicator
- 6 independent variables related to account and transaction details

Data Dictionary

Variable Name	Description
creditLimit	Credit Limit
availableMoney	Available Balance
transactionAmount	Transaction Amount
cardPresent	Card Present Transaction 1= Yes, 0= No
CVVcorrect	CVV verified 1=Yes, 0=No
account_age	Account age in months
isFraud	Fraudulent Transaction 1= Yes, 0= No

Index	Credit Limit ($)	Available Money ($)	Transaction Amount ($)	Card Present	CVV correct	Account Age (months)	Is Fraud = 1
0	15000	9193.34	394.01	1	1	28	1
1	250	151.98	321.21	0	1	32	0
2	2500	2448.24	312.77	0	1	20	0
3	20000	1255.53	311.63	0	1	14	0
4	5000	1594.94	675.46	0	1	4	1
...	...	...	...	...	...	...	...
49997	15000	10825.28	188.36	0	1	24	0
49998	7500	5493.00	146.64	0	1	31	0
49999	7500	830.85	30.85	1	1	3	0

Fraudulent Transaction Detection – Linear Regression

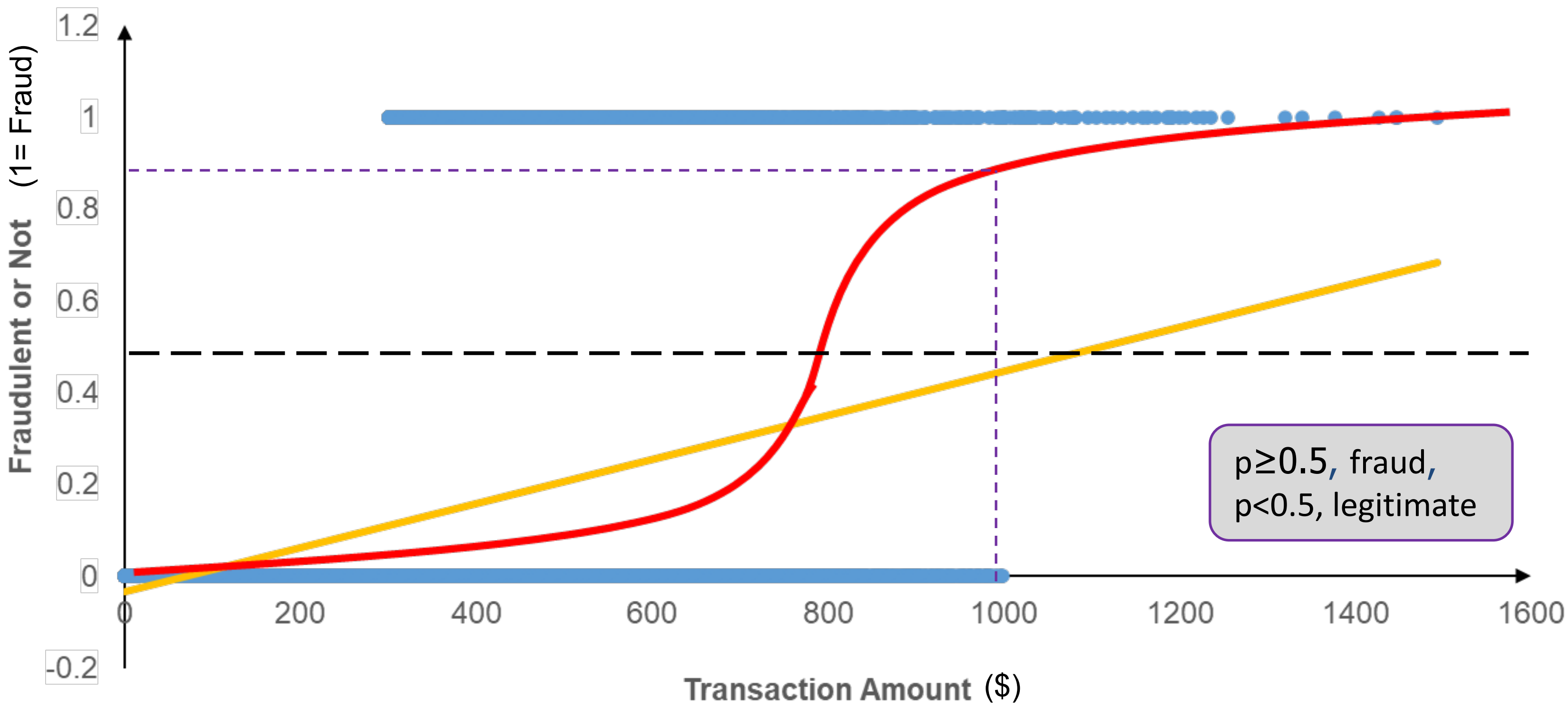


The blue dot represent the data, I fit a linear line (yellow) to model the capture the relationship b/w fraud and transaction amount. Is it a good fit?

Limitations:

- ➢ Not fit the categorial data well
- ➢ Negative value (and plus and minus infinity) can occur.
- ➢ Sensitive to outliers

Fraudulent Transaction Detection – Logistic Regression



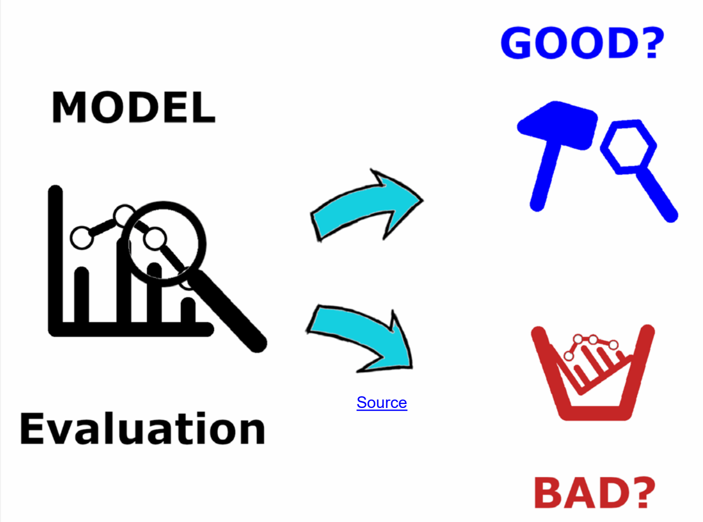
Now let us fit a s-shaped line (red) into the graph, does it fit the data better?

Enhancements:

- Better fit than linear regression
- Range between 0 and 1
- Less sensitive to outliers

Intuition of Logistical Regression

- Often used to solve classification problems (i.e., Yes/No).
  If outcome is a binary category -> binary logistic regression **(focus)**
  if more than two categories -> multinomial logistic regression



- Extend the idea of linear regression to situations where dependent variables are categorical

- The idea behind logistical regression is simple, **convert the linear line into an s-shaped line using a logistical function** (also known aa s sigmoid function). In other words, logistical regression transforms the linear regression output into a probability (0 to 1) by using a logistical function.

Function of Logistical Regression

Logistic (Sigmoid) Function:

$$p = \frac{1}{1 + e^{-f(x)}}$$

$where$

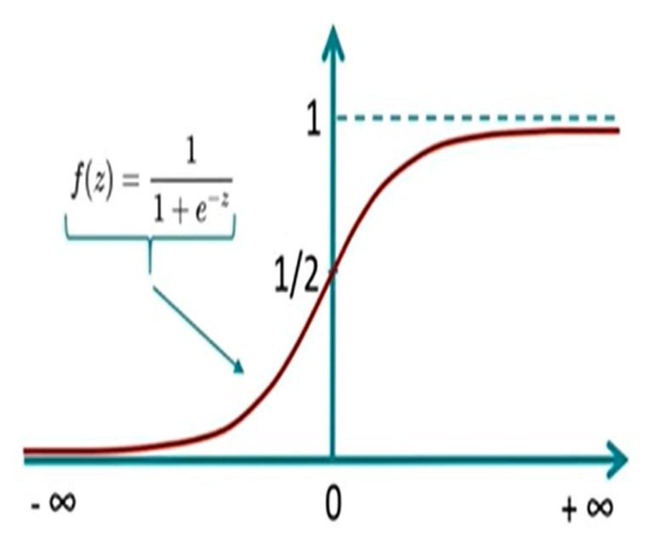
p is the probability of being 1 (i.e., is fraud)

$f(x) = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$, $f(x)$ is a linear regression

Or $e^{f(x)} = \frac{p}{1-p}$, called the odds ratio

$f(x) = \ln(\frac{p}{1-p})$, called the log odds ratio

The odds ratio is simply the probability of 1 (i.e., win) divided by the probability of 0 (i.e., loss)



Algorithm for Logistic Regression

In prior statistical class, we might learn that maximum likelihooh (MLE) could use to estimate the coefficient for logistical regression.

In machine learning, we can minimize the log loss function with gradient descent for logistical regression to solve the coefficients.

$$J(\mathbf{x}) = -\frac{1}{m}\sum_{j=1}^{m} y^j \log\left(\hat{y}^j\right) + (1-y^j)\log\left(1-\hat{y}^j\right)$$

$$J(\mathbf{x}) = -\frac{1}{m}\sum_{j=1}^{m}\left(y^j \log\left(\frac{1}{1+e^{-\sum_{i=0}^{n}\beta_i x_i^j}}\right) + (1-y^j)\log\left(1-\frac{1}{1+e^{-\sum_{i=0}^{n}\beta_i x_i^j}}\right)\right)$$

Don’t worry about fully understanding this gradient descent. In practice we never have to implement it ourselves. Python package are there for us.

Main takeaway should be the relationship between log odds and probability.

Conceptual Soundness

Key Assumptions

Goodness of Fit

Performance Metric

Model Interpretation

Implementation in Python

Case Study in Google Colab

In-Class Excise

Key Assumptions

Compared to linear regression, logistical regression has few assumptions. The error in logistical regression does not need to be normally distributed given the binary nature of the data.

- **No Multicollinearity (Most important):** All independent variables are not highly correlated
- **Linearity:** Linearity of independent variables and log-odds
- **Independence:** The observations (records) are independent of each other

How to check the validity of those assumptions?
Various statistical tests outside of the scope of this class!

Confusion Matrix

The output of logistical regression is a probability, while the observed outcome is a binary variable (1/0). How to compare them?

Define threshold to cutoff positive or negative

Predicted probability (p)	Predicted Outcome ($\hat{y}$)
0.4	0
0.8	1
0.9	1
0.2	0
0.1	0
0.05	0

Actual Outcome (y)
0
1
0
0
1
0

What should I do next?
Let me know what you think

We can use a confusion Matrix to measure the performance of the any classification problem. (not only limit to logistical regression)

- Conceptual Soundness
- Key Assumptions
- Goodness of Fit
- **Performance Metric**
- Model Interpretation
- Implementation in Python
- Case Study in Google Colab
- In-Class Excise

Confusion Matrix

➢ **We need to know all the performance metric below**

		Predicted condition		Sources: [10][11][12][13][14][15][16][17][18] view · talk · edit	
	Total population $= P + N$	**Positive (PP)**	**Negative (PN)**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)**	**True positive (TP),** hit	**False negative (FN),** type II error, miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)	**False positive (FP),** type I error, false alarm, overestimation	**True negative (TN),** correct rejection	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2 PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers

Just kidding! But there are so many metrics to evaluate classification performance. We will focus on some widely used ones in next slides.

Confusion Matrix

- **True Positive:** Predicted positive, and the actual is positive
- **True Negative:** Predicted negative, and the actual is negative
- **False positive:** Predicted positive, but the actual is negative
- **False negative:** Predicted negative, but the actual is positive

	Predicted: No	**Predicted: Yes**
Actual: No	**Ture Negative (TN)**	**False Postive (FP) Type 1 error**
Actual: Yes	**False Negative (FN) Type 2 error**	**True Positive (TP)**

Conceptual Soundness

Key Assumptions

Goodness of Fit

Performance Metric

Model Interpretation

Implementation in Python

Case Study in Google Colab

In-Class Excise

Confusion Matrix

Fraud Detection N= 10,000	Predicted: Non-frauds	Predicted: Frauds
Actual: Non-frauds (Total=9,354)	**9,199 (TN)**	**155 (FP)**
Actual: Frauds (Total=646)	**486 (FN)**	**160 (TP)**

Use the fraud detection example:

Accuracy: Overall rates of corrected classification

(TP+TN)/Total= (160+9,199)/10,000=93.59%

Misclassification rate: Overall error rates

(FP+FN)/Total= (155+486)/10,000=6.41%

True Positive rate (Sensitivity/Recall): how often an actual fraud (positive) can be detected

TP/Actual Positive = 160/(160+486)=24.76% (detection rate)

False Positive rate: how often a non-fraud is being incorrectly classified as fraud

FP/Actual Negative = 155/(9199+155)=1.66%

- Conceptual Soundness
- Key Assumptions
- Goodness of Fit
- **Performance Metric**
- Model Interpretation
- Implementation in Python
- Case Study in Google Colab
- In-Class Excise

Confusion Matrix

Fraud Detection N= 10,000	Predicted: Non-frauds	Predicted: Frauds
Actual: Non-frauds (Total=9,354)	9,199 (TN)	155 (FP)
Actual: Frauds (Total=646)	486 (FN)	160 (TP)

Use the fraud detection example:

True Negative rate (specificity): how often an actual non-fraud (negative) been correctly classified

TN/Actual Negative = 9199/(9354)=98.34%

False negative rate: how often a fraud is being incorrectly classified as non-fraud

FN/Actual Positive= 486/646=75.2%

Precision:

TP/Predicted Positive = 160/(160+155)=50.1%

Conceptual Soundness

Key Assumptions

Goodness of Fit

Performance Metric

Model Interpretation

Implementation in Python

Case Study in Google Colab

In-Class Excise

Confusion Matrix - Excise

Actual Outcome (y)	**Predicted Outcome ($\hat{y}$)**
0	0
1	1
0	1
0	0
1	0
0	0

Let us do a small exercise together.

Calculate TP, FN and Accuracy in the data?

		Predicted 1	**Predicted** 0
Actual	1	(TP) ?	(FN) ?
Actual	0	(FP) ?	(TN) ?

N = number of obs. = TP + FN + FP + TN

Accuracy = (TN + TP)/N

Solution

Conceptual Soundness

Key Assumptions

Goodness of Fit

Performance Metric

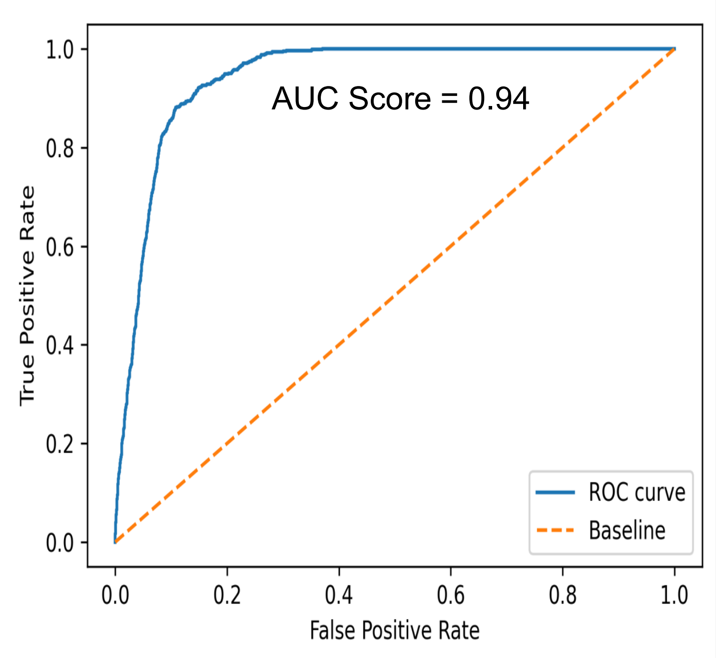
Model Interpretation

Implementation in Python

Case Study in Google Colab

In-Class Excise

Performance Metrics – ROC/AUC



Receiver operating characteristic (ROC) curve

- A plot of true positive rate vs. false positive rate at different classification thresholds.
- A higher curve (toward the northwestern corner) indicates a better model performance.

Area under curve (AUC) score

- Ranges from 0 to 1
- Higher value indicates better performance
- 0.5 means random guess, which likes a fair coin flipping (Baseline)

Summary Quiz

1. Logistic regression is used to predict ____________.
   a) Continuous variables
   b) Categorial variables
   c) Count variables

2. In the fraud detection example, the fraud detection rate is also referred to as ______________ in the confusion matrix.
   a) True positive rate
   b) False positive rate
   c) True negative rate
   d) False negative rate

Summary Quiz

3. In the fraud detection example, the false alarm (legitimate transaction being incorrectly classified as fraud) is also called ______________ in the confusion matrix.
    a) True positive rate
    b) False positive rate
    c) True negative rate
    d) False negative rate

Today's Agenda

- Modelling for categorical variable
- **In-Class Assignment**
- Q&A

In-Class Assignment

Let’s go to Canvas and open the in-class excise for today’s class

Next Session

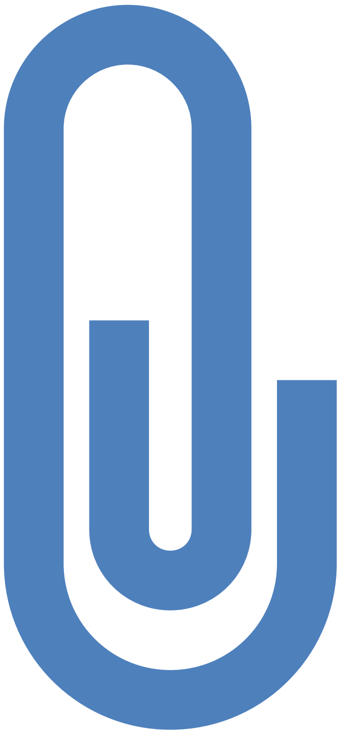
Topics: Implementation of Logistic Regression

Assignments: HW9 – One Notebook + 1 Quiz. Due next Wednesday, Nov. 9th

Thank you for your time!

Q&A

Appendix 1 : Solution to Excise



Goodness of Fit

As in linear regression, goodness of fit in logistic regression attempts to get at how well a model fits the data.

- Pseudo R square
- Likelihood ratio
- Chi-square goodness of fit tests and deviance
- Hosmer-Lemeshow tests

Note: not the focus of our class.

TP

- Let us illustrate this using our example

Actual Outcome (y)	Predicted Outcome ($\hat{y}$)
0	0
1	1
0	1
0	0
1	0
0	0

Predicted

		1	0
Actual	1	TP 1	FN
	0	FP	TN

N = number of obs. = TP + FN + FP + TN

Accuracy= (TN + TP)/N

TN

- Let us illustrate this using our example

Actual Outcome (y)	Predicted Outcome ($\hat{y}$)
0	0
1	1
0	1
0	0
1	0
0	0

		Predicted	
		1	0
Actual	1	TP 1	FN
	0	FP	TN 3

N = number of obs. = TP + FN + FP + TN

Accuracy= (TN + TP)/N

FN

- Let us illustrate this using our example

Actual Outcome (y)	Predicted Outcome $(\hat{y})$
0	0
1	1
0	1
0	0
1	0
0	0

		Predicted	
		1	0
Actual	1	TP 1	FN 1
	0	FP	TN 3

N = number of obs. = TP + FN + FP + TN

Accuracy= (TN + TP)/N

FP

- Let us illustrate this using our example

Actual Outcome (y)	Predicted Outcome ($\hat{y}$)
0	0
1	1
0	1
0	0
1	0
0	0

		Predicted	
		1	0
Actual	1	TP 1	FN 1
	0	FP 1	TN 3

N = number of obs. = TP + FN + FP + TN

Accuracy= (TN + TP)/N

Accuracy

- Let us illustrate this using our example

Actual Outcome (y)	Predicted Outcome ($\hat{y}$)
0	0
1	1
0	1
0	0
1	0
0	0

		Predicted	
		1	0
Actual	1	TP 1	FN 1
	0	FP 1	TN 3

N = number of obs. = TP + FN + FP + TN

Accuracy= (TN + TP)/N

N = number of obs. = 6

Accuracy= (TN + TP)/N = (3 + 1)/6