DATASCI207-005/007 Applied Machine Learning

Vilena Livinsky, PhD(c)

School of Information, UC Berkeley

Week 5: 02/05/2025 & 02/06/2025

Today's Agenda

- Multiclass Classification & Metrics
- Walkthroughs:
  - Metrics
  - Multiclass Classification
    - + TensorFlow

Practice

TensorFlow: General Modeling Steps

Create

Create a model

- Create your model architecture
- *Functional* or *Sequential API*

```
# Use Keras Sequential API to build a linear regression model.
model = keras.Sequential()
model.add(keras.layers.Dense(
    input_shape=[num_features], # each input has num_features features
    units=1,                    # there is a single output
    use_bias=True               # include a learned bias parameter
    ))
```

Compile

Compile a model

- Decide on **loss**
- Decide on model performance metrics
- Decide on how to improve the process; choose an **optimizer**

```
model.compile(loss='mean_squared_error', optimizer='adam',
              metrics =['mean_absolute_error'])
```

loss vs. metrics?

Fit

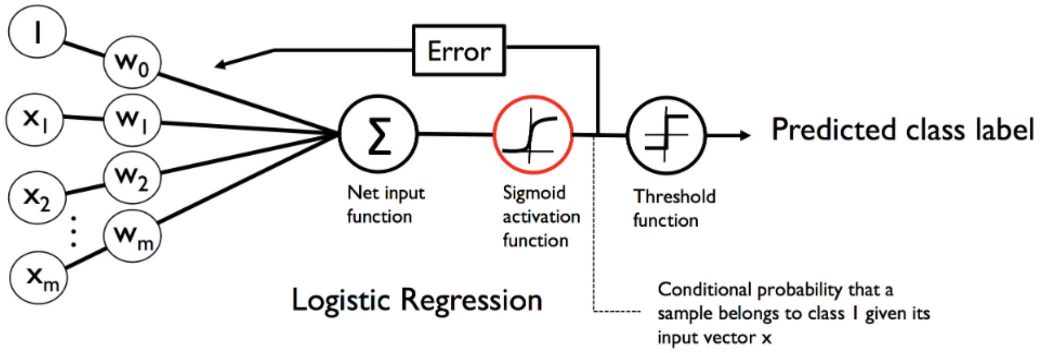
Fit a model

- Learn a function

```
# Build a model and train it. Hold out 10% of data for validation.
model = build_model(num_features=len(features))
model.fit(x=X, y=Y,
          validation_split=0.1, batch_size=16, epochs=5)

# Use the model to predict the training labels.
Y_pred = model.predict(x=X).flatten()
```

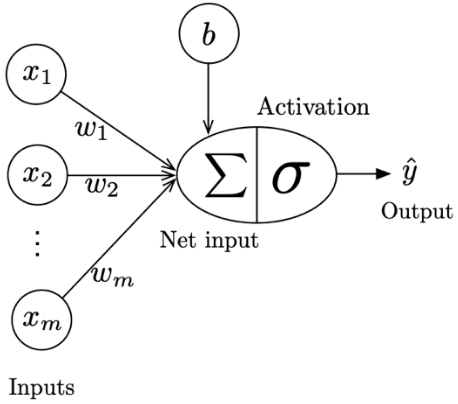
Activation Functions: Sigmoid vs. Softmax



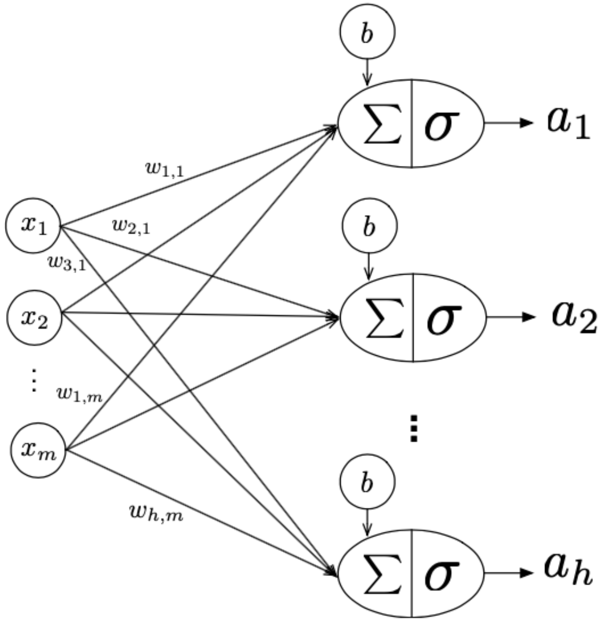
Sigmoid,

$$\varphi(z) = \frac{1}{1 + e^{-z}}$$

$$\hat{y} := \begin{cases} 1 & \text{if } \sigma(z) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$



?



?

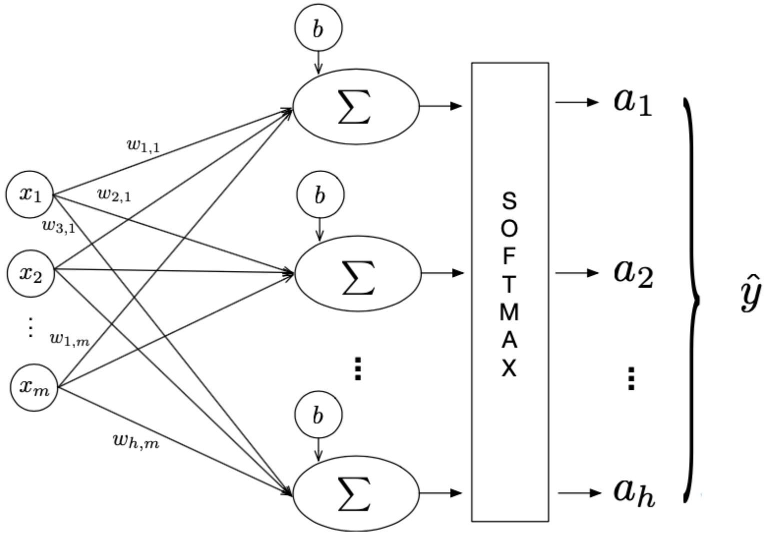


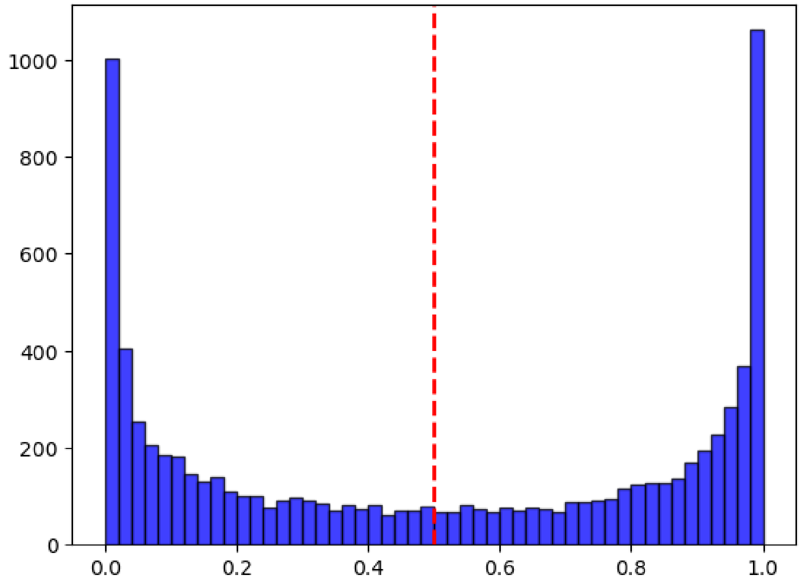
Image Ref., Edited: Raschka, S., & Mirjalili, V. (2019). Python Machine Learning, Third Edit.

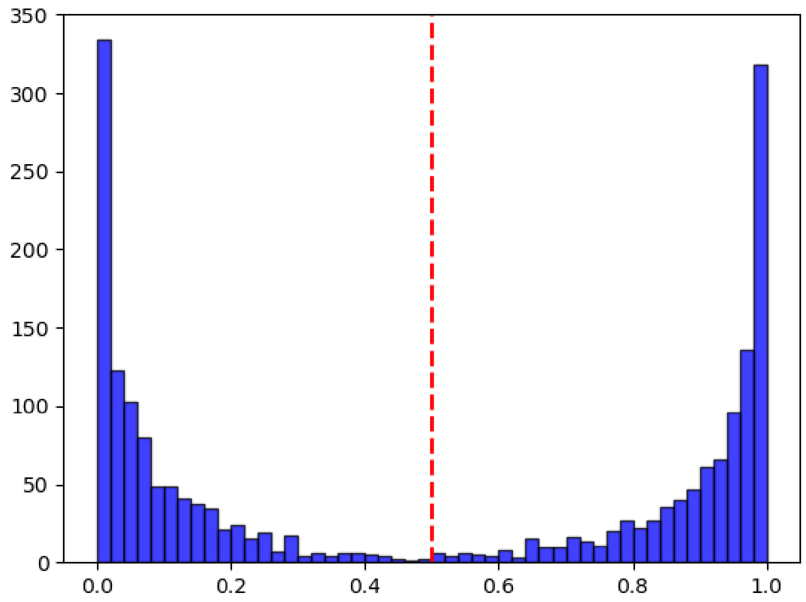
Logistic Regression: Accuracy

Accuracy

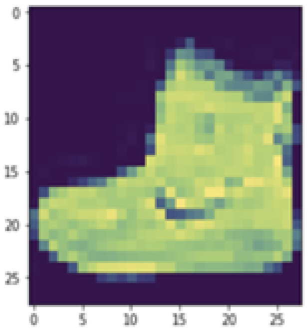
$$\frac{(TP+TN)}{(TP+TN+FP+FN)}$$

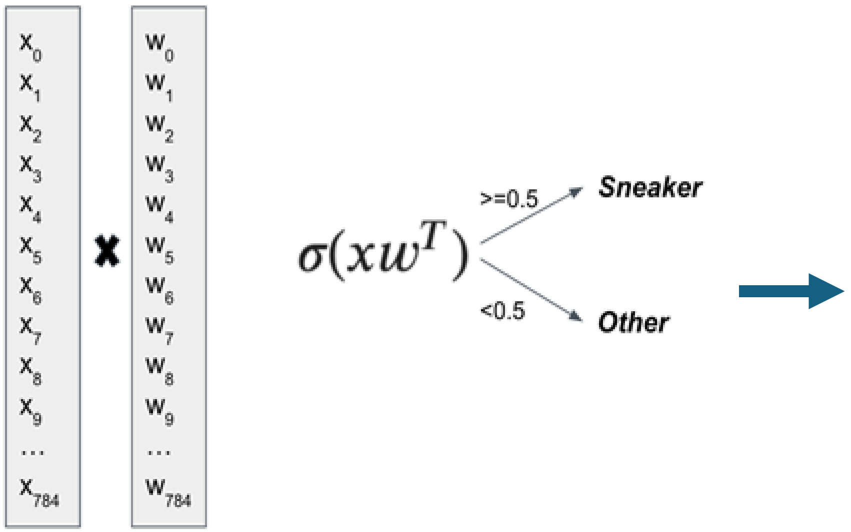
- A **single** threshold accuracy
  - Considers model quality only at one point
  - Threshold: typically, 0.5
- Compare: Model confidence
- Compare: Prediction 0.49 vs. 0.51 for y=1?





The Confusion Matrix: Binary Classifier





	y'=Sneaker (1)	*y'=Other (0)*
y=Sneaker (1)	True Positive (TP)	False Negative (FN)
y=Other (0)	False Positive (FP)	True Negative (TN)

Example, consider:

- Goal: Want to detect sneaker
- Metric: Accuracy

Actual \ Predicted	Other
Other	990
Sneaker	10

- **Correct Predictions**: 990
- **Total Predictions**: 1000
- **Accuracy**: 99.00%

Accuracy

$$\frac{(TP+TN)}{(TP+TN+FP+FN)}$$

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \times 100$$

$$\text{Accuracy} = \frac{990}{1000} \times 100 = 99.00\%$$

ROC AUC: Binary Classifier

- Example: 4 predictions
  - Class 1 = x2 (1s)
  - Class 0 = x2 (0s)

y	$\hat{y}$
0	0.3
1	0.5
0	0.6
1	0.9

Typically, 0.5 threshold

$$\hat{y} := \begin{cases} 1 & \text{if } \sigma(z) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

ROC AUC: Binary Classifier

- Example: 4 predictions
  - Class 1 = 2 (1s)
  - Class 0 = 2 (0s)
- Consider next all predictions y’ as threshold

Typically, **0.5** threshold

y	$\hat{y}$
0	0.3
1	0.5
0	0.6
1	0.9

$\hat{y}$	0.3 threshold
0.3	0 (TN)
0.5	1 (TP)
0.6	1 (FP)
0.9	1 (TP)

<=0.3 : 0
> 0.3 : 1

0.5 threshold	0.6 threshold
0 (TN)	0 (TN)
0 (FN)	0 (FN)
1 (FP)	0 (TN)
1 (TP)	1 (TP)

Ref.: Jon Krohn, The Essential Machine Learning Foundations: Math, Probability, Statistics, and Computer Science

ROC AUC: Binary Classifier

- Example: 4 predictions
  - Class 1 = 2 (1s)
  - Class 0 = 2 (0s)
- Consider next all predictions y' as threshold

Typically, **0.5** threshold

y	$\hat{y}$
0	0.3
1	0.5
0	0.6
1	0.9

≤ 0.3 : 0
> 0.3 : 1

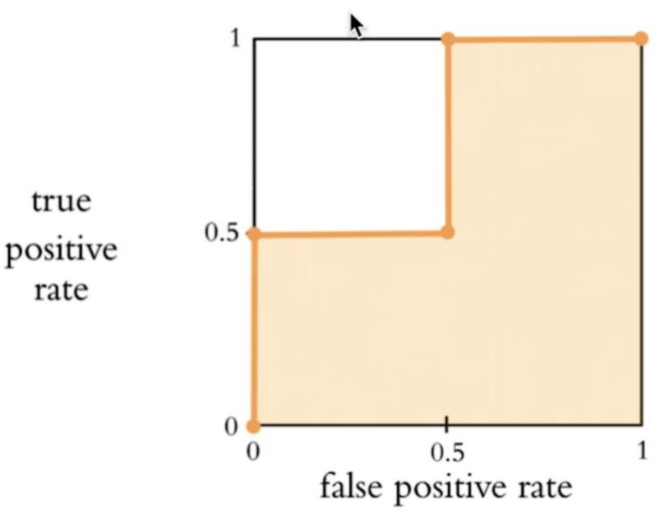
$\hat{y}$	0.3 threshold	0.5 threshold	0.6 threshold
0.3	0 (TN)	0 (TN)	0 (TN)
0.5	1 (TP)	0 (FN)	0 (FN)
0.6	1 (FP)	1 (FP)	0 (TN)
0.9	1 (TP)	1 (TP)	1 (TP)

	0.3 threshold	0.5 threshold	0.6 threshold
True Positive Rate $= \frac{TP}{TP+FN}$	$\frac{2}{2+0} = 1.0$	$\frac{1}{1+1} = 0.5$	$\frac{1}{1+1} = 0.5$
False Positive Rate $= \frac{FP}{FP+TN}$	$\frac{1}{1+1} = 0.5$	$\frac{1}{1+1} = 0.5$	$\frac{0}{0+2} = 0.0$

Ref.: Jon Krohn, The Essential Machine Learning Foundations: Math, Probability, Statistics, and Computer Science

ROC AUC: Binary Classifier

- Example: 4 predictions
  - Class 1 = 2 (1s)
  - Class 0 = 2 (0s)
- Consider next all predictions y' as threshold



Typically, **0.5** threshold

y	$\hat{y}$
0	0.3
1	0.5
0	0.6
1	0.9

<=0.3 : 0
> 0.3 : 1

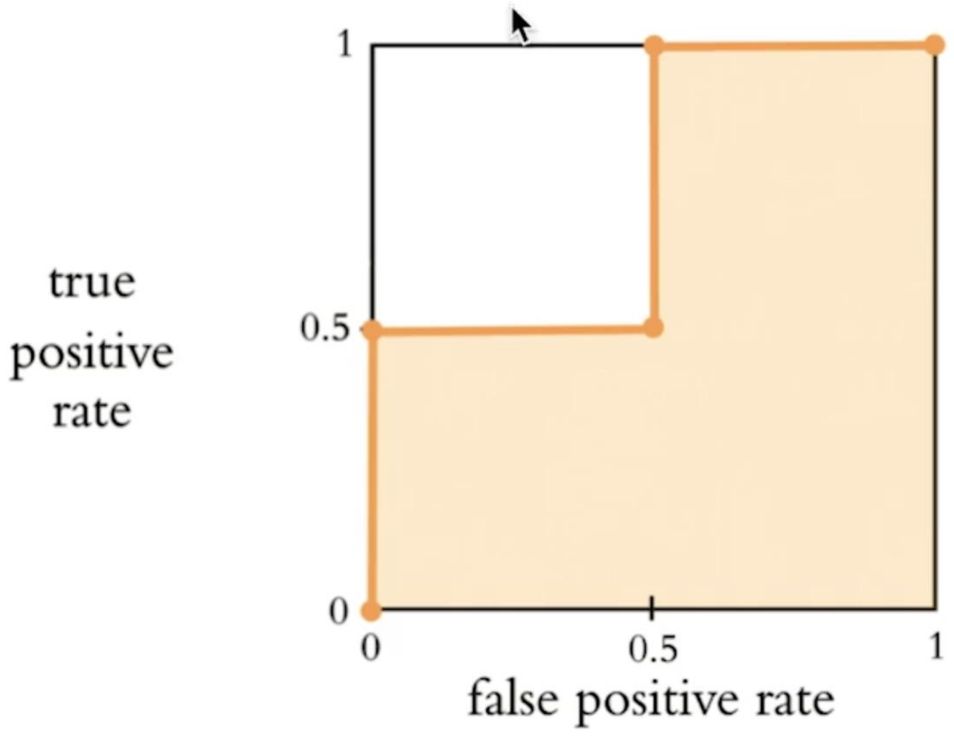
$\hat{y}$	0.3 threshold
0.3	0 (TN)
0.5	1 (TP)
0.6	1 (FP)
0.9	1 (TP)

0.5 threshold	0.6 threshold
0 (TN)	0 (TN)
0 (FN)	0 (FN)
1 (FP)	0 (TN)
1 (TP)	1 (TP)

	0.3 threshold	0.5 threshold	0.6 threshold
True Positive Rate $= \frac{TP}{TP+FN}$	$\frac{2}{2+0} = 1.0$	$\frac{1}{1+1} = 0.5$	$\frac{1}{1+1} = 0.5$
False Positive Rate $= \frac{FP}{FP+TN}$	$\frac{1}{1+1} = 0.5$	$\frac{1}{1+1} = 0.5$	$\frac{0}{0+2} = 0.0$

Ref.: Jon Krohn, The Essential Machine Learning Foundations: Math, Probability, Statistics, and Computer Science

ROC AUC: Binary Classifier



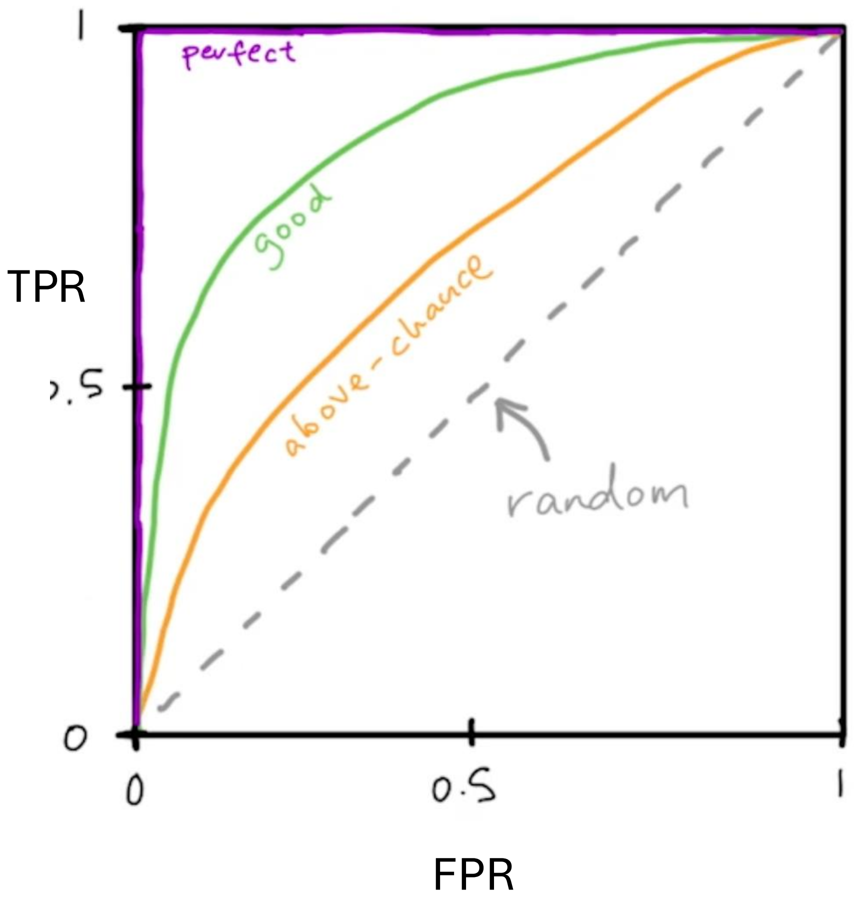


Image Ref.: Jon Krohn, The Essential Machine Learning Foundations: Math, Probability, Statistics, and Computer Science

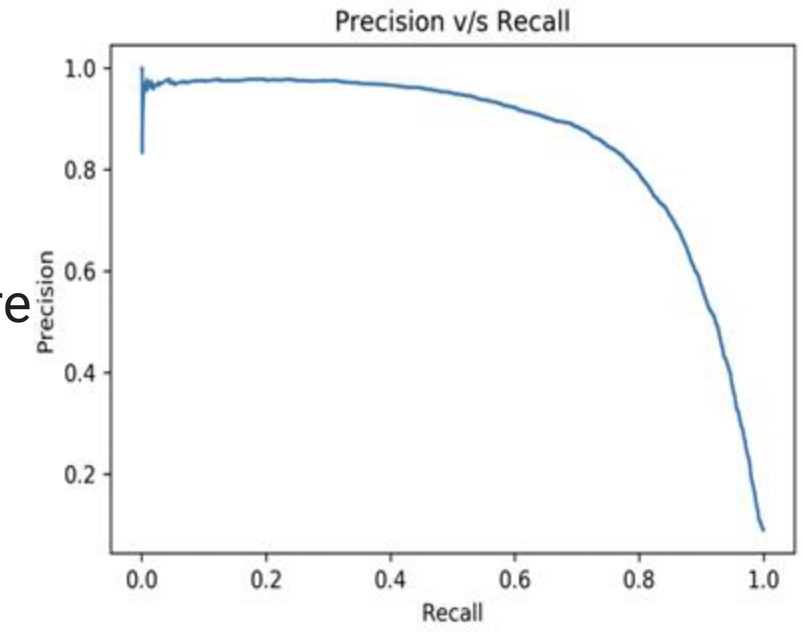
Multiclass Confusion Matrix

- Classifier over multiple classes
  - Ex.: shirt-shirt = 514 correct predictions of shirt
  - Ex.: shirt-tshirt = 108 times predicted shirt as tshirt
- Multiclass ROC
  - Scikit-learn: Multiclass ROC
- Confusion Matrix:
  - Sklearn: Confusion Matrix

	y' = 1	y' = *0*
y = *1*	TP	FN
y = *0*	FP	TN

True Label \ Predicted Label	t-shirt	trouser	pullover	dress	coat	sandal	shirt	sneaker	bag	ankle boot
t-shirt	787	6	12	59	6	1	108	0	20	1
trouser	6	930	16	39	6	0	1	0	2	0
pullover	15	0	669	8	188	1	101	0	18	0
dress	32	18	7	849	42	1	47	0	4	0
coat	0	3	93	39	771	0	88	0	6	0
sandal	1	0	0	2	0	811	0	103	10	73
shirt	148	3	133	36	131	1	514	0	34	0
sneaker	0	0	0	0	0	36	0	898	0	66
bag	2	1	12	11	3	6	24	9	931	1
ankle boot	0	0	0	0	0	12	1	54	1	932

Precision, Recall, F1



emphasize correctness if we predict outcome is xyz (cost: a high num of FN)

- **Precision** is the percentage of **predicted** positives that were correctly classified
  - how good a model is at predicting the positive class
  - concerned with accuracy of the positive predictions
  - increasing precision reduces recall and vice versa

optimizing for recall helps with minimizing the chance of not detecting xyz

- **Recall** is the percentage of **actual** positives that were correctly classified

- Consider when: Class imbalance

Consider Use-Cases:
- Fraud detection
- Subscribe or not to a product (Customer Acquisition)
- Medical Field: detection of a tumor with xyz imaging

Precision	**Recall**
TP / (TP+FP)	TP / (TP+FN)

$$F_1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{TP}{TP + \frac{FN+FP}{2}}$$

Logistic Regression: Softmax Regression

$$z = w_1x_1 + \ldots + w_mx_m + b = \sum_{l=1}^{m} w_l x_l + b = \mathbf{w}^T\mathbf{x} + b$$

Sigmoid

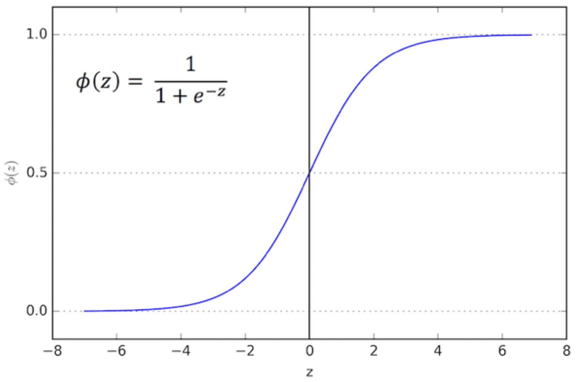
$$\text{logit}(p(y = 1 \mid \mathbf{x})) = z$$

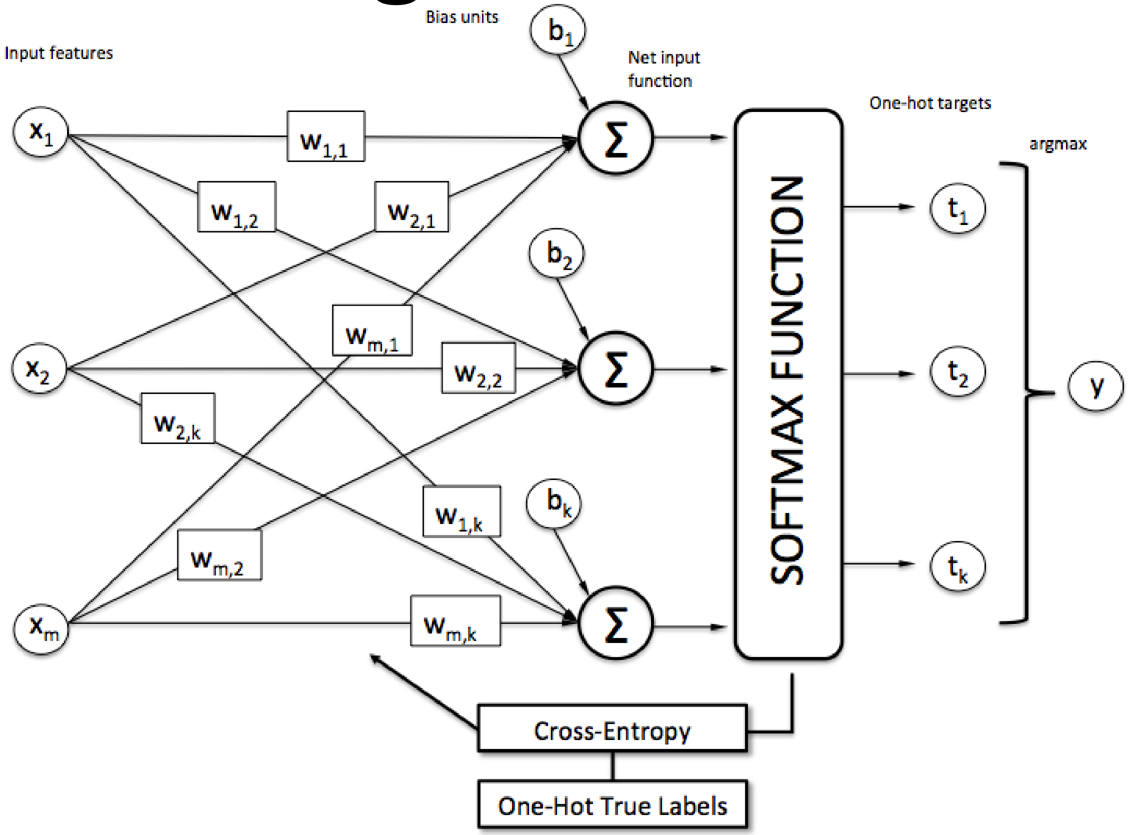
Inverse

$$\varphi(z) = \frac{1}{1 + e^{-z}}$$

Softmax

$$P(y = j \mid z^{(i)}) = \varphi_{softmax}(z^{(i)}) = \frac{e^{z^{(i)}}}{\sum_{j=0}^{k} e^{z_k^{(i)}}}$$





$$\sigma_{\text{softmax}}(z_t^{[i]}) = \frac{e^{z_t^{[i]}}}{\sum_{j=1}^{h} e^{z_j^{[i]}}}$$

one-hot encode labels

$$\mathcal{L} = \sum_{i=1}^{n} \sum_{j=1}^{h} -y_j^{[i]} \log\left(a_j^{[i]}\right)$$

over all training examples

h different class labels

- Another example: walkthrough by Rashka: [Multinomial Logistic]

Loss: Categorical Cross-Entropy

- Categorical cross-entropy
  - minimizes the distance between the probability distributions output by the model and the true distribution of the targets
- Handling labels in multiclass classification:
  - Encoding via categorical encoding (also known as one-hot encoding)
    - Use: *categorical_crossentropy* as a loss function
  - Encoding the labels as integers
    - Use: *sparse_categorical_crossentropy* loss function

Ref.: Chollet, F. (2021). Deep Learning with Python Second Edition, Shelter Island.