Inf620 2025 Lecture Linear, Polynomial and Logistic Regression

Depto de Informática - UFV

Ricardo Ferreira
ricardo@ufv.br

2025

Lesson Plan

- Class Material (click here for the Colab)
- **Today's Lesson**: Regression Techniques
  - Linear Regression
  - Polynomial Regression
  - Logistic Regression
  - Key differences and practical considerations

What is Regression?

- A supervised learning technique to model relationships between input features and output variables
- Used for prediction, trend analysis, and inference
- Types we'll study:
  - Linear Regression
  - Polynomial Regression
  - Logistic Regression

Linear Regression

- Models the relationship as: $y = wx + b$
- Objective: Minimize the Mean Squared Error (MSE)
- Solution via:
  - Analytical method (Normal Equation)
  - **Gradient Descent**
- Output is continuous (Regression)

Polynomial Regression

- Extension of linear regression with polynomial terms:
  $y = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^d$
- Captures non-linear relationships
- Still a linear model in terms of parameters
- Risk of overfitting with high-degree polynomials

Logistic Regression

- Used for binary classification problems
- Models probability: $P(y = 1|x) = \frac{1}{1+e^{-(\mathbf{wx}+\mathbf{b})}}$
- Output is between 0 and 1 (**Classification**)
- Uses Cross-Entropy (Log Loss) instead of MSE
- Decision boundary at $P = 0.5$

Comparison of Methods

- **Linear Regression:** Predicts continuous values, assumes linearity
- **Polynomial Regression:** Enhances linear regression with non-linearity
- **Logistic Regression:** Predicts class probabilities, used for classification
- Choose based on output type and data pattern

Practical Implementations

- Use `scikit-learn` for all three methods
- Explore in Colab notebook:
  - Linear fit using `LinearRegression()`
  - Polynomial fit using `PolynomialFeatures` + `LinearRegression`
  - Logistic fit using `LogisticRegression()`
- Evaluate using appropriate metrics:
  - MSE/R^2 for regression
  - Accuracy, Precision, Recall for classification

5 years ago.....

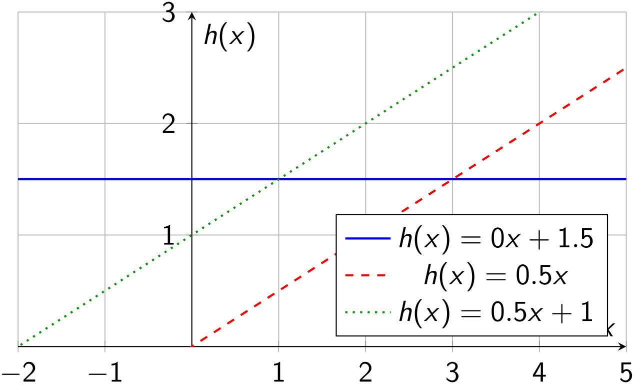
- **Top 10 Machine Learning Models for Data Science**
  1. Linear Regression – the go-to model for prediction and interpretation
  2. Logistic Regression – baseline for binary classification problems
  3. Decision Tree – easy to interpret and fast to train
  4. Random Forest – robust ensemble method improving accuracy
  5. Support Vector Machines (SVM) – powerful for classification with clear margins
  6. Gradient Boosting Machines (GBM) – competitive in Kaggle competitions
  7. XGBoost – efficient and accurate boosting method widely adopted
  8. k-Nearest Neighbors (KNN) – intuitive, lazy learner for classification/regression
  9. Clustering (e.g., K-Means) – unsupervised learning to find natural groupings
  10. Principal Component Analysis (PCA) – key tool for dimensionality reduction

5 years ago.....

- **Top 10 Deep Learning Models for Data Science**
  1. Recurrent Neural Networks (RNN) – for time series and sequence data
  2. Convolutional Neural Networks (CNN) – best for image processing
  3. Long Short-Term Memory (LSTM) – solving RNN's vanishing gradient issues
  4. Generative Adversarial Networks (GANs) – realistic image synthesis
  5. AutoEncoders – unsupervised feature learning and anomaly detection
  6. BERT – transformer-based NLP model from Google
  7. GPT-3 – large-scale generative language model revolutionizing NLP

Visualizing $w \cdot x + b$

- Click here - Lecture 3. Linear Regression - Prof. Lucas Nascimento
- Example linear functions in hypothesis space:
  - $h(x) = 0x + 1.5$: horizontal line
  - $h(x) = 0.5x + 0$: passes through origin
  - $h(x) = 0.5x + 1$: same slope, different intercept



Loss Function: Mean Squared Error (MSE)

- To evaluate how well our linear model fits the data, we use a loss function.
- For linear regression, the most common choice is the Mean Squared Error (MSE):

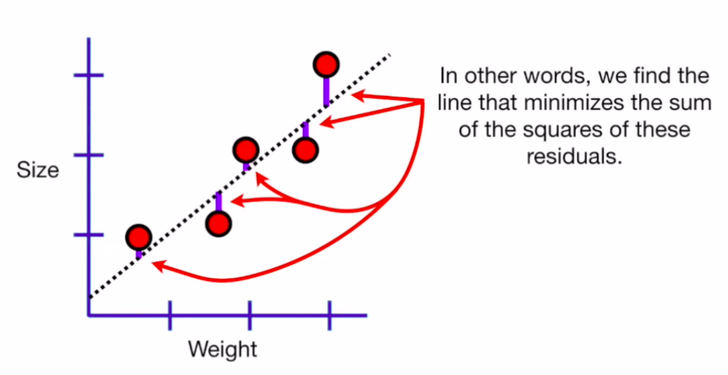
$$L(w, b) = \frac{1}{2m} \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)})^2$$

- m: number of training examples, $h(x^{(i)})$: predicted value for i
- $y^{(i)}$: true value for example i

Goal

Minimize $L(w, b)$ to make predictions as close as possible to the actual values.

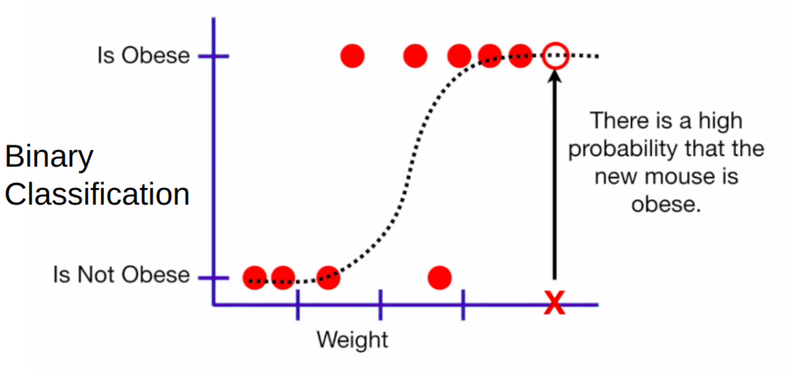
Logistic Regression Statquest - LINEAR REGRESSION



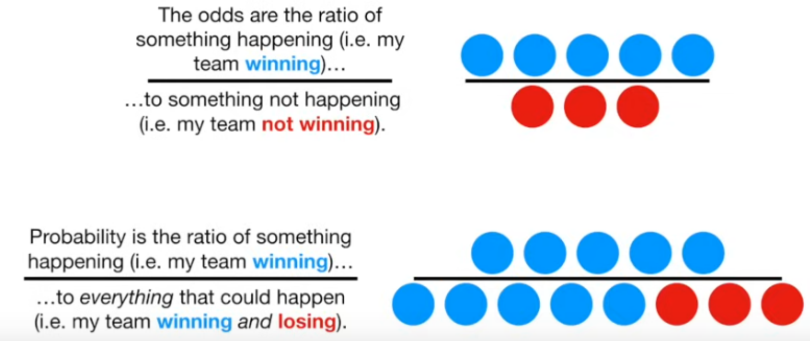
Click here to watch Stat Quest Video

Deep Learning - Slides - Prof. Lucas Nascimento + click here - video

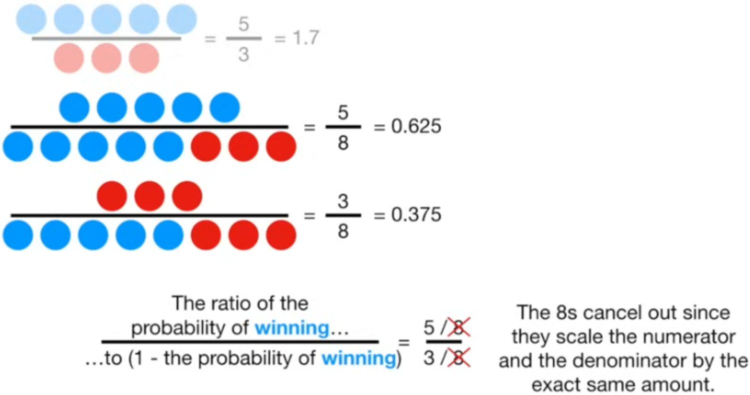
Logistic Regression Statquest



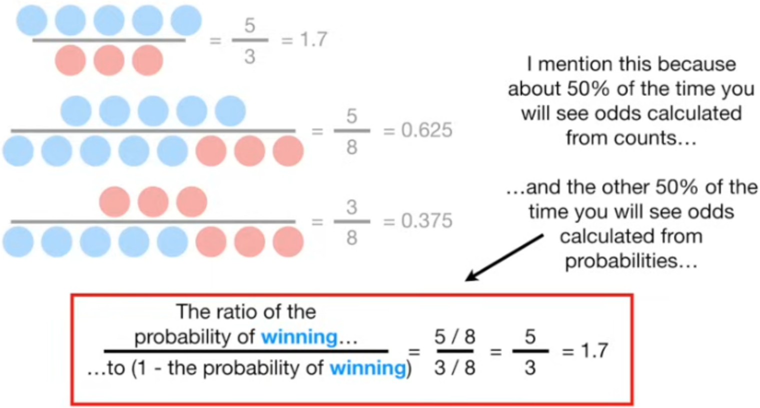
Click here to watch Stat Quest Video
inf620 Lecture 2023
Deep Learning Slides - Prof. Lucas Nascimento



Click here to watch Stat Quest Video



Click here to watch Stat Quest Video

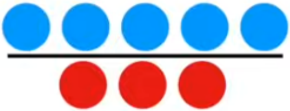


Click here to watch Stat Quest Video

In summary, the odds are just...

...the ratio of something happening (i.e. my team **winning**)...

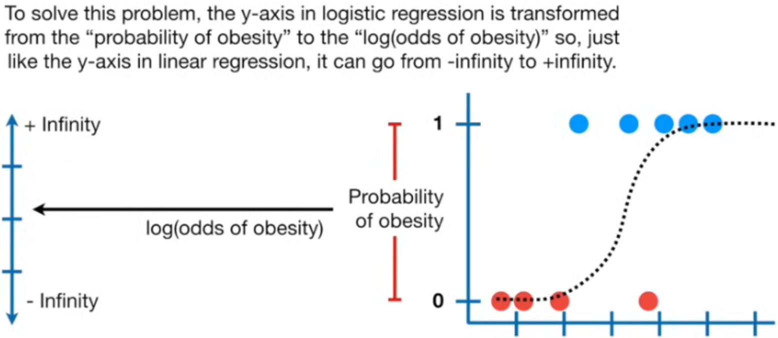
...to something not happening (i.e. my team **not winning**).



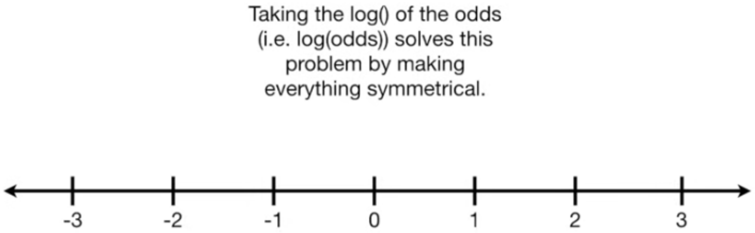
And the log(odds) is just the log of the odds.
It's no big deal!

The log(odds) makes things symmetrical, easier to interpret and easier for fancy statistics.

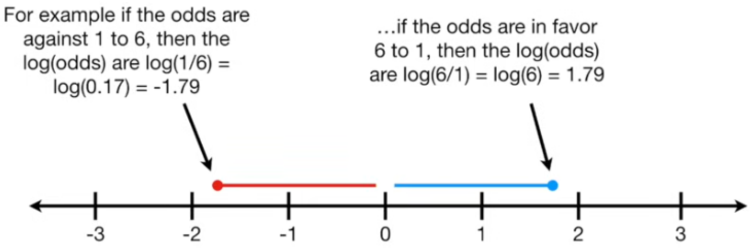
Click here to watch Stat Quest Video



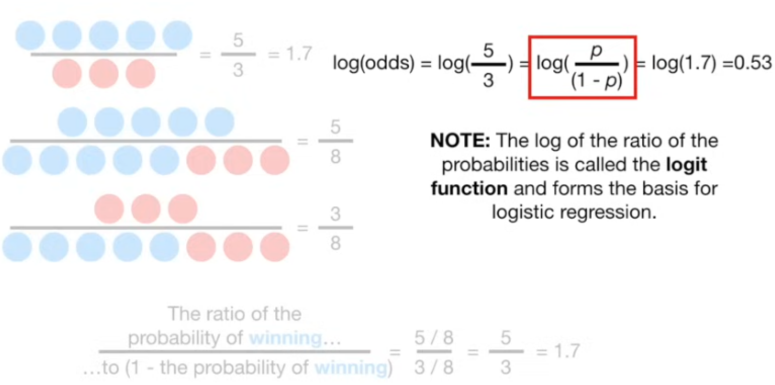
Click here to watch Stat Quest Video



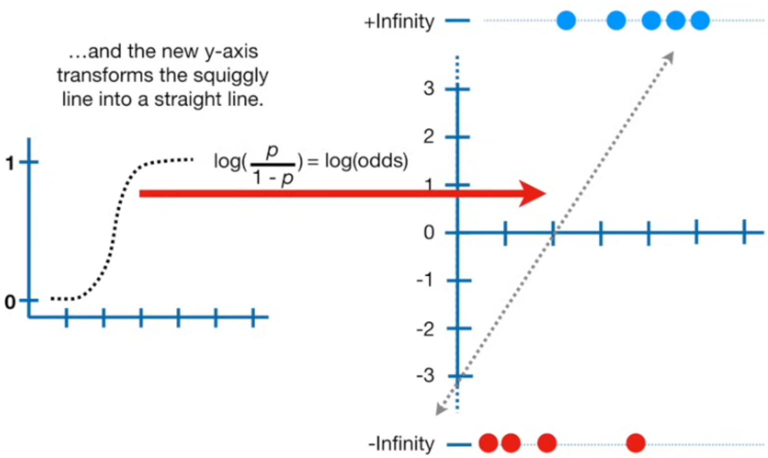
Click here to watch Stat Quest Video



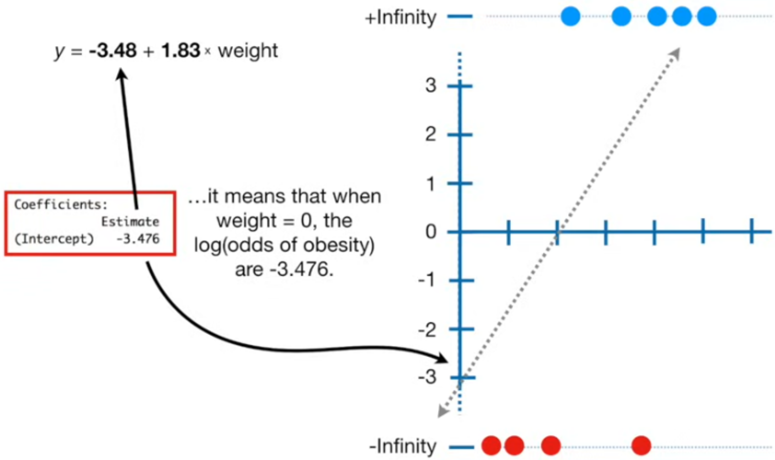
Click here to watch Stat Quest Video



Click here to watch Stat Quest Video



Click here to watch Stat Quest Video



Click here to watch Stat Quest Video