Fondamenti di Analisi dei Dati

from data analysis to predictive techniques

Prof. Antonino Furnari (antonino.furnari@unict.it)
Corso di Studi in Informatica
Dip. di Matematica e Informatica
Università di Catania

Logistic Regression

A powerful parametric classifier which can be also used to study relationships between a binary variable and a set of continuous variables.

Beyond K-NN: The Need for Parametric Models

Compare K-NN with a simple parametric predictive model:

$$f(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$$

K-NN Limitations

- Struggles with many features
- Requires memorizing entire training set
- Classification demands expensive neighbor searches

Parametric Model Advantages

- Fast and compact
- No need to store training data
- Natural classification of new examples

Linear regression excels at modeling continuous relationships, but cannot handle qualitative dependent variables. We need a new approach for classification problems.

The Breast Cancer Wisconsin Dataset

Our example uses digitized images from fine needle aspirate (FNA) of breast masses. The dataset contains measurements of various quantities with a categorical Diagnosis variable: M (malignant) or B (benign).

569

Total Samples

Patient observations in the dataset

31

Features

Measured variables per sample

2

Classes

Benign or Malignant diagnosis

	radius1	texture1	perimeter1	area1	smoothness1	concave_points3	symmetry3	fractal_dimension3	Diagnosis
0	17.99	10.38	122.80	1001.0	0.11840	0.2654	0.4601	0.11890	M
1	20.57	17.77	132.90	1326.0	0.08474	0.1860	0.2750	0.08902	M
2	19.69	21.25	130.00	1203.0	0.10960	0.2430	0.3613	0.08758	M
3	11.42	20.38	77.58	386.1	0.14250	0.2575	0.6638	0.17300	M
4	20.29	14.34	135.10	1297.0	0.10030	0.1625	0.2364	0.07678	M
...	...	...	...	...	...	...	...	...	...
564	21.56	22.39	142.00	1479.0	0.11100	0.2216	0.2060	0.07115	M
565	20.13	28.25	131.20	1261.0	0.09780	0.1628	0.2572	0.06637	M
566	16.60	28.08	108.30	858.1	0.08455	0.1418	0.2218	0.07820	M
567	20.60	29.33	140.10	1265.0	0.11780	0.2650	0.4087	0.12400	M
568	7.76	24.54	47.92	181.0	0.05263	0.0000	0.2871	0.07039	B

569 rows × 31 columns

Observing the Relationship

When we plot radius1 against Diagnosis, a clear pattern emerges. Low radius1 values tend to correspond with benign cases, while large radius1 values associate with malignant cases. But how do we quantify this relationship formally?

1

Low Radius

More benign cases observed

2

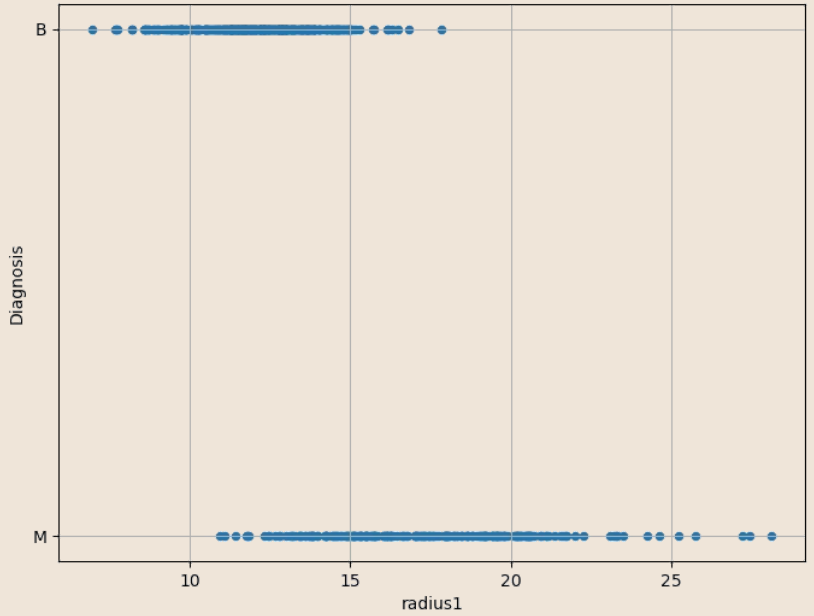
Medium Radius

Mixed distribution of cases

3

High Radius

More malignant cases observed



Why Linear Regression Fails

Converting B ⇒ 0 and M ⇒ 1 and applying linear regression produces a model that doesn't capture the true relationship. The residual plot reveals strong correlation between residuals and the independent variable—a clear sign of model inadequacy.

Wrong Function Type

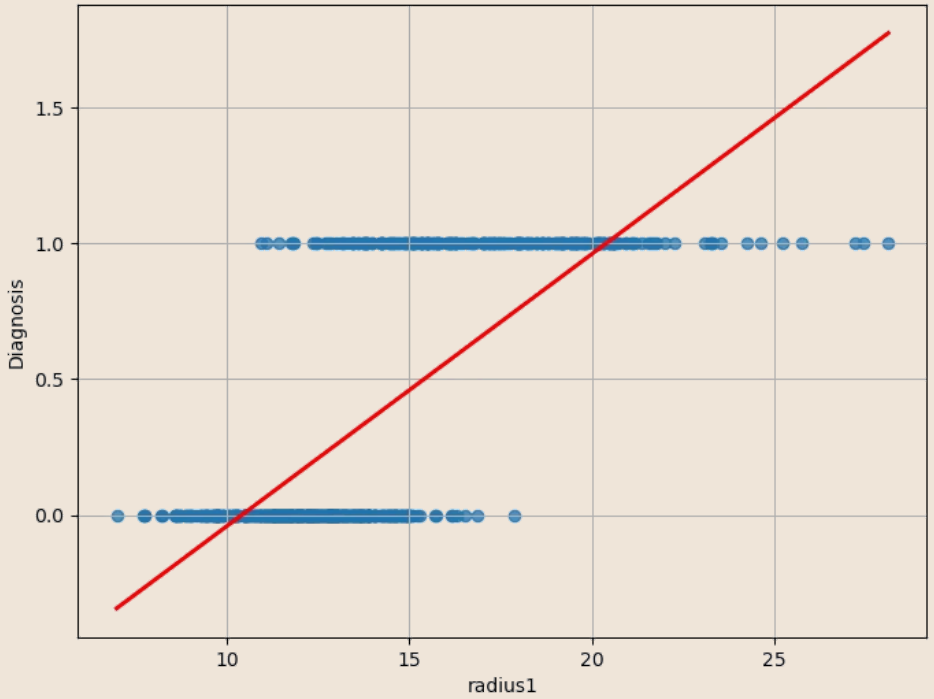
Linear regression maps $\mathbb{R} \to \mathbb{R}$, but we need $\mathbb{R} \to \{0,1\}$ for discrete classification

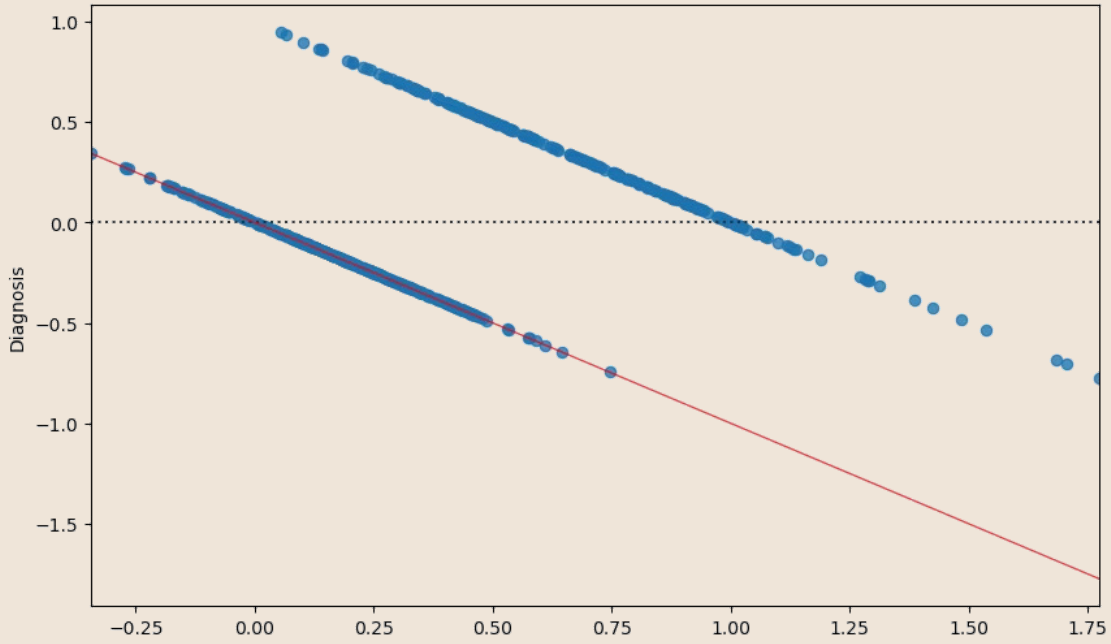
Poor Predictions

Despite $R^2 = 0.533$, the model produces values outside [0,1] range

Violated Assumptions

Residual patterns indicate the linear model is fundamentally inappropriate





From Binary Values to Probabilities

Instead of predicting discrete 0 or 1 values, we can model probabilities—"soft" values indicating our belief that Diagnosis equals 1. This transforms our problem into modeling the following quantity:

$$P(Diagnosis = 1|radius1)$$

This approach is what **discriminative models** do. However, even modeling probability directly with a linear combination

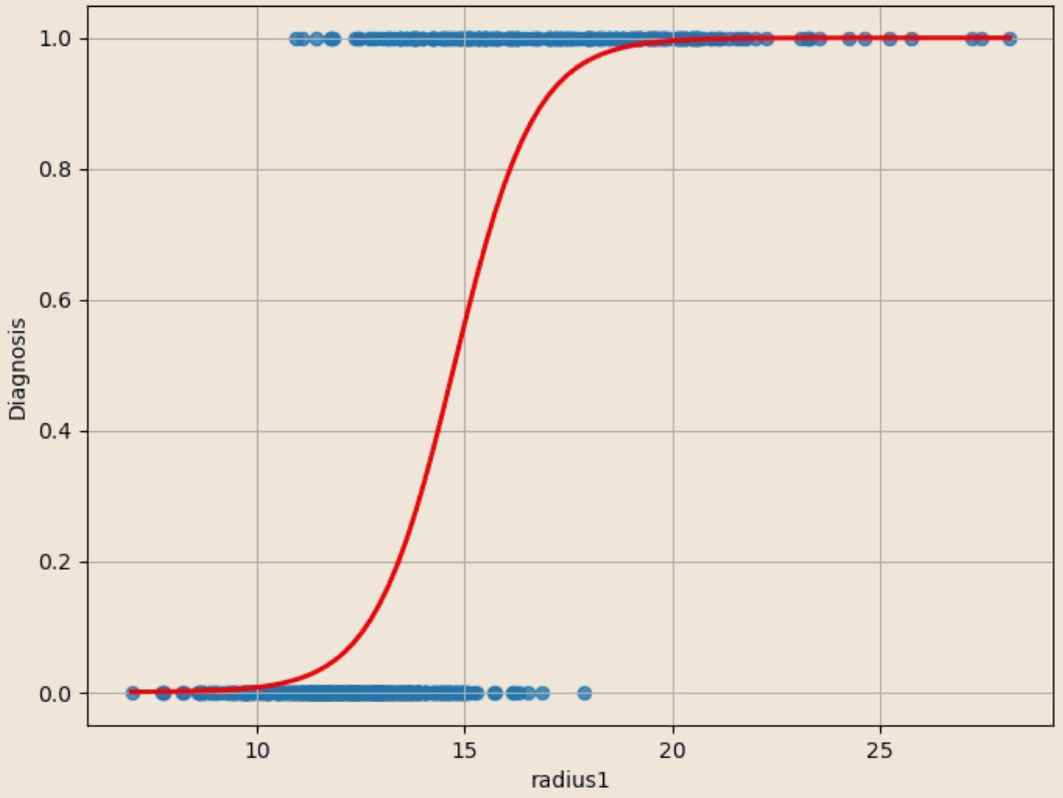
$$P(Diagnosis = 1|radius1) = \beta_0 + \beta_1 radius1$$

radius1 creates problems: the output can fall outside [0,1], producing meaningless probability values.

Also, we expect probabilities not to be linear. We need a function that naturally saturates to 0 for low values and to 1 for high values, staying within [0,1] for intermediate values.

> Intuitively: once we have enough evidence that the class is positive (a large enough radius), we cannot increase our probability value indefinitely, we need it to saturate to one.

From this intuition, it is clear that the linear regression is not an appropriate model in this case.



The Logistic Function

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

The logistic function (also called sigmoid) possesses exactly the properties we need. It's differentiable (good for optimization), maps any real number to [0,1], and saturates at the extremes.

Range [0,1]

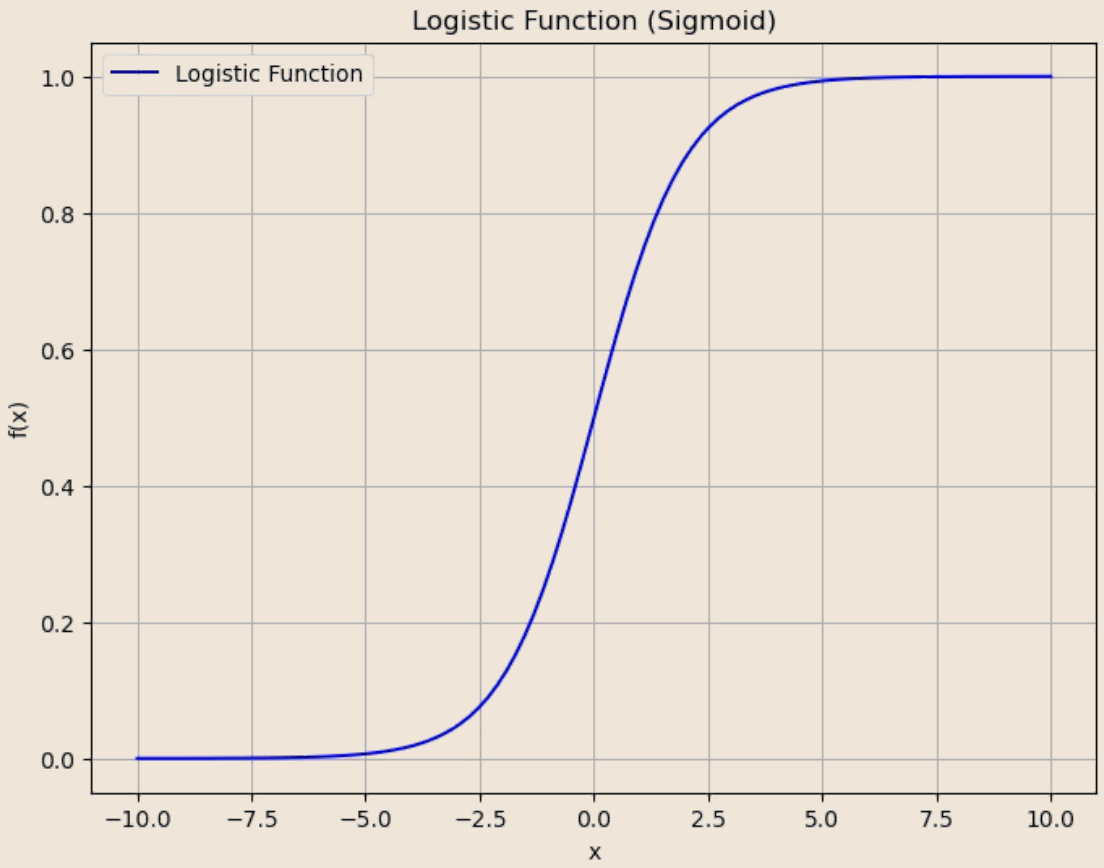
Perfect for probabilities

Saturation

Approaches 0 and 1 at extremes

Differentiable

Enables gradient-based optimization



Understanding Odds

Odds represent the ratio of the probability that an event will happen to the probability that it will not happen. They provide a direct measure of likelihood relative to the alternative outcome.

$$Odds = \frac{P(\text{event})}{1 - P(\text{event})}$$

In many cultures, especially in regions like the UK and Australia, odds are often preferred over probabilities because they directly translate to payout ratios, offering an intuitive understanding of potential returns.

Mathematically, odds are powerful because they transform the bounded probability range [0, 1] into an unbounded range [0, ∞), which is particularly useful in statistical modeling like logistic regression.

Practical Example: Soccer Betting

Consider a soccer match where a bookmaker gives odds of 3:1 for Team A beating Team B. This means:

- Team A is expected to be 3 times more likely to win than Team B.
- If you bet €10 on Team A and they win, you might collect €30 (depending on the bookmaker's payout structure).
- The probability of Team A winning is calculated as 3 / (3 + 1) = 0.75.
- Using the formula: Odds = 0.75 / (1 - 0.75) = 0.75 / 0.25 = 3 (or 3:1).

Probability to Odds Transformation Examples

1

Probability: 0.2

Odds = 0.2 / (1 - 0.2) = 0.2 / 0.8 = 0.25 (or 1:4)

2

Probability: 0.5

Odds = 0.5 / (1 - 0.5) = 0.5 / 0.5 = 1 (or 1:1)

3

Probability: 0.75

Odds = 0.75 / (1 - 0.75) = 0.75 / 0.25 = 3 (or 3:1)

4

Probability: 0.9

Odds = 0.9 / (1 - 0.9) = 0.9 / 0.1 = 9 (or 9:1)

The Logistic Regression Model

We define our model by applying the logistic function to a linear combination of features:

$$P(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}}$$

Through algebraic manipulation, we can show that:

$$e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n} = \frac{P(y = 1|\mathbf{x})}{1 - P(y = 1|\mathbf{x})}$$

The Odds

The right side represents the **odds** of P(y=1|x)—the ratio of positive to negative outcomes. If we believe an event occurs 3 times out of 10, the odds are 3/7.

The Logit

Taking the logarithm gives us the **logit** (log-odds), showing logistic regression as linear regression on the log-odds scale.

Parameter Estimation via Maximum Likelihood

To determine the optimal values for the parameters β in logistic regression, we employ a statistical technique called **Maximum Likelihood Estimation (MLE)**.

Unlike linear regression, where we directly fit to continuous values, here we are modeling probabilities from binary outcomes (0 or 1). Note that we are not observing the true probability values (only binary outcomes), so we cannot simply apply an MSE loss to predict probabilities.

Our objective is to find the parameters β that maximize the probability of observing our given dataset under the logistic model. The probability of an individual observation y given the features $\mathbf{x}$ and parameters β can be expressed compactly as:

$$P\left(y|\mathbf{x};\beta\right)=\left(f_{\beta}\left(\mathbf{x}\right)\right)^{y}\left(1-f_{\beta}\left(\mathbf{x}\right)\right)^{1-y}$$

Where $f_{\beta}(\mathbf{x})=\sigma(\beta^{T}\mathbf{x})$ is the predicted probability of $y=1$.

Assuming all training examples are independent, the overall likelihood function $L(\beta)$ for the entire dataset is the product of these individual probabilities:

$$L\left(\beta\right)=\prod_{i=1}^{N}f_{\beta}\left(\mathbf{x}^{(i)}\right)^{y^{(i)}}\left(1-f_{\beta}\left(\mathbf{x}^{(i)}\right)\right)^{1-y^{(i)}}$$

Maximizing this likelihood function directly can be computationally challenging. A common practice is to minimize its negative logarithm, known as the negative log-likelihood (nll) or cross-entropy loss. This transformation simplifies the optimization process while preserving the objective, as taking the logarithm converts products into sums, and maximizing a function is equivalent to minimizing its negative counterpart.

Thus, our cost function $J(\beta)$ for logistic regression becomes:

$$J\left(\beta\right)=-\sum_{i=1}^{N}[y^{(i)}\log\sigma\left(\beta^{T}\mathbf{x}^{(i)}\right)+\left(1-y^{(i)}\right)\log\left(1-\sigma\left(\beta^{T}\mathbf{x}^{(i)}\right)\right)]$$

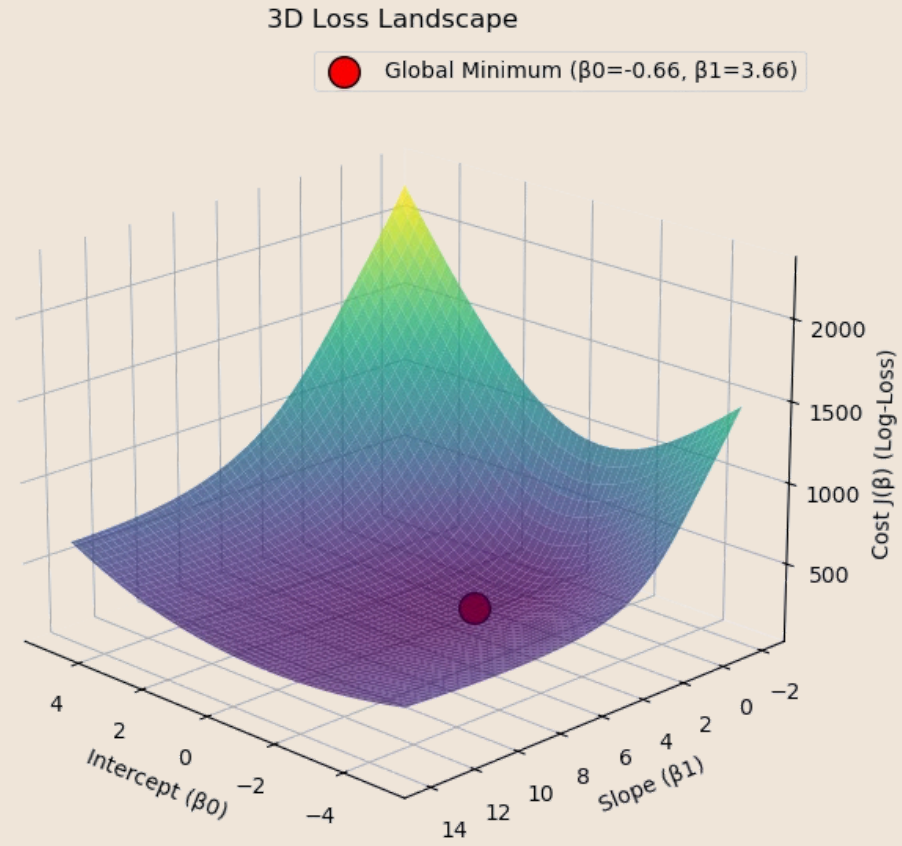
Minimizing this cost function allows us to find the parameter vector β that best fits our training data, effectively estimating the relationship between our features and the log-odds of the binary outcome.

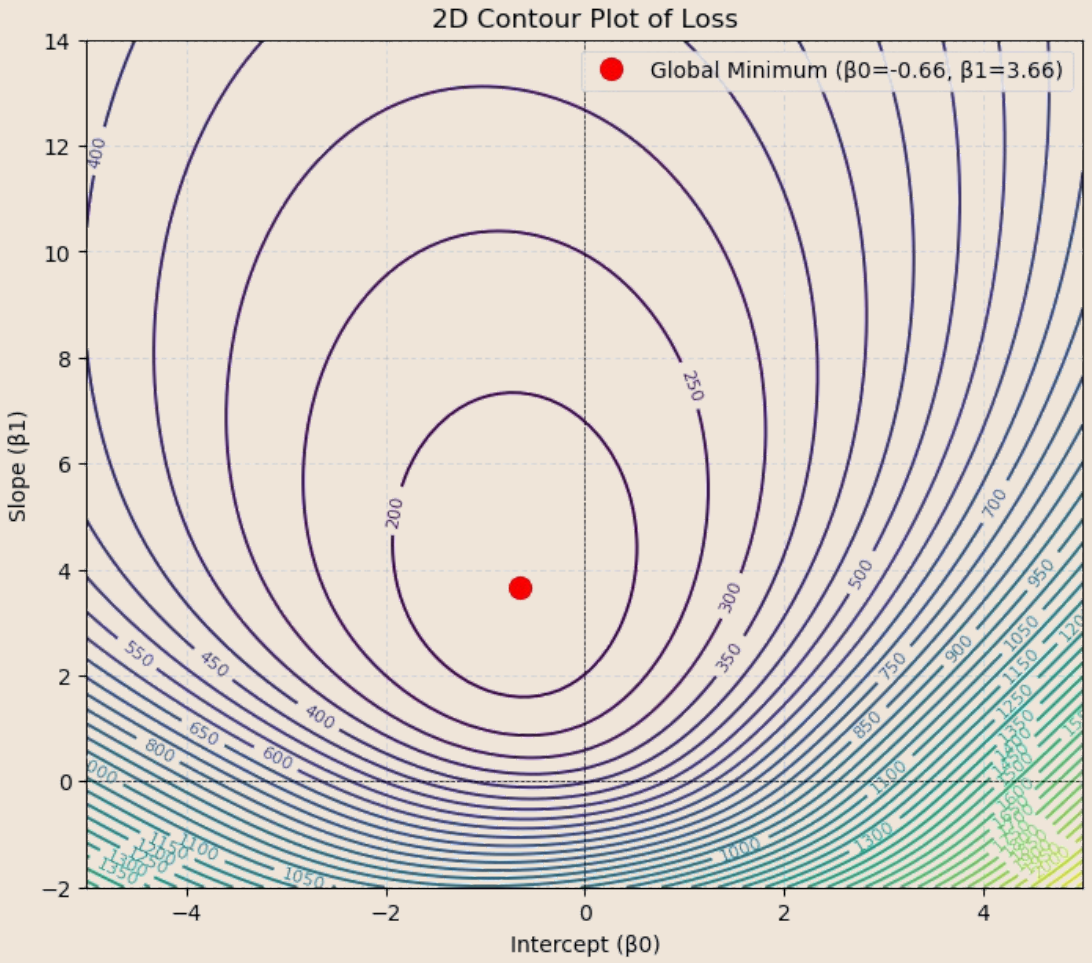
While this loss function is convex, there is no closed form solution, hence we have to resort to iterative optimization algorithms such as gradient descent.

Visualizing the Cost Function

The optimization function of a logistic regression is **convex**, guaranteeing a single global minimum. To truly understand what this "loss landscape" looks like, let's visualize it. We'll consider a simple model, `Diagnosis ~ radius1`, which involves two parameters: β_0 (intercept) and β_1 (slope).

By plotting the Cost J for various combinations of β_0 and β_1, we can observe the characteristic "bowl" shape of a convex function.





The plots clearly reveal a single, smooth, **convex "bowl"**. This means:

- There are no "local minima" or "bumpy" areas where an optimizer could get stuck.
- This confirms why the logistic regression cost function is guaranteed to be convex.
- The red dot at the very bottom of this bowl signifies the one optimal set of parameters (β_0, β_1) that minimizes the cost.

An iterative solver like **Gradient Descent** is guaranteed to find this minimum, much like a marble released anywhere on the bowl's rim is guaranteed to roll directly to the bottom.

Interpreting the Intercept β_0

The logistic regression model reveals meaningful interpretations for each coefficient. Starting with the intercept:

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 x$$

When x = 0, this simplifies to:

$$\frac{p}{1-p} = e^{\beta_0}$$

The term p/(1-p) is the **odds** that y=1 when x=0. For example, if odds are 3/1, the event is 3 times more likely to occur than not occur.

Therefore, when x=0, it is e^{β_0} times more likely that y=1 rather than y=0.

> **Example** Suppose we found this logistic regressor:
>
> $$P(y=1|x) = \sigma(1.02 + 0.3x)$$
>
> We have $\beta_0 = 1.02$ and $\exp(\beta_0) \approx 2.77$. This suggests that if x is zero, then the probability of $y=1$ is 2.77 times more likely than not. The base value starts from a very high odd, so if $x=0$, we are almost certain that the example is a positive one.

Interpreting Variable Coefficients β_i

For any coefficient β_i, we can derive its interpretation through the log-odds:

$$\log \text{odds}(p|x+1) - \log \text{odds}(p|x) = \beta_1$$

Exponentiating both sides reveals:

$$\frac{\text{odds}(p|x+1)}{\text{odds}(p|x)} = e^{\beta_1}$$

Multiplicative Effect

A one-unit increase in x multiplies the odds by e^{β_1}

Percentage Interpretation

If $e^{\beta_1} = 1.05$, odds increase by 5% per unit of x

This interpretation extends naturally to multiple logistic regression with many predictors.

> **Example** Let's get back to our logistic regressor example:
>
> $$P(y=1|x) = \sigma(1.02 + 0.3x)$$
>
> We have $\beta_1 = 0.3$ and $\exp(\beta_1) \approx 1.35$. This suggests that an increase in one unit in x has a multiplicative increase of 1.35 in the odds, and hence an increase of about 35%.

Example: Breast Cancer

Let's briefly see an example in which we try to build the logistic regressor `Diagnosis ~ radius1 + texture1 + perimeter1 + area1 + smoothness1 + compactness1 + concavity1 + symmetry1`. Here we set `B=1` and `M=0`.

Initial regressor

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.6591	0.224	-11.896	0.000	-3.098	-2.220
radius1	0.4688	0.133	3.532	0.000	0.208	0.730
texture1	0.0219	0.003	7.376	0.000	0.016	0.028
perimeter1	-0.0473	0.021	-2.272	0.023	-0.088	-0.006
area1	-0.0009	0.000	-3.985	0.000	-0.001	-0.000
smoothness1	5.1389	1.221	4.208	0.000	2.740	7.538
compactness1	0.3080	0.854	0.360	0.719	-1.370	1.986
concavity1	2.0973	0.414	5.065	0.000	1.284	2.911
symmetry1	1.2739	0.568	2.244	0.025	0.159	2.389

After backward elimination

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.6708	0.221	-12.086	0.000	-3.105	-2.237
radius1	0.4360	0.097	4.517	0.000	0.246	0.626
texture1	0.0219	0.003	7.405	0.000	0.016	0.028
perimeter1	-0.0419	0.014	-2.915	0.004	-0.070	-0.014
area1	-0.0010	0.000	-4.477	0.000	-0.001	-0.001
smoothness1	5.3093	1.125	4.719	0.000	3.099	7.519
concavity1	2.1479	0.389	5.517	0.000	1.383	2.913
symmetry1	1.3132	0.557	2.359	0.019	0.220	2.407

These are now all statistically relevant. For instance, we can see that:

- When all variables are set to zero, the odds of the benign tumor are $e^{-2.6708} \approx 0.07$, or $\frac{7}{100}$. This is a base value.
- An increment in one unit of `texture1` increments the odds of a benign tumor multiplicatively by a factor of $e^{0.0219} \approx 1.02$ (a $+2\%$), when all other variables are constant.
- An increment of one unit of `perimeter1` decrements the odds of benign tumor multiplicatively by a factor of $e^{-0.0419} \approx 0.96$ (a -4%), when all other variables are constant.

Evaluating Logistic Regression Models

Why R^2 Doesn't Work

Linear regression's R^2 measures fit by comparing residual sum of squares (RSS) to total sum of squares (TSS). But logistic regression predicts *probabilities*, not continuous values, and minimizes negative log-likelihood, not squared residuals.

Linear Regression

Minimizes squared residuals (RSS)

$$R^2 = 1 - \frac{RSS}{TSS}$$

Logistic Regression

Minimizes negative log-likelihood

Uses Pseudo R^2 instead

McFadden's Pseudo R²

McFadden's Pseudo R² compares our model's log-likelihood to a baseline "intercept-only" model:

$$R^2_{McFadden} = 1 - \frac{LL_{Model}}{LL_{Null}}$$

The intercept-only mode:

$$P(y = 1|x) = \sigma(\beta_0)$$

Example

Let's consider the following example:

- **Log-Likelihood:** -105.19 (LL_Model)
- **LL-Null:** -452.39 (LL_Null)
- **Pseudo R-sq.:** 0.23

This value was calculated as $1 - \frac{-105.19}{-452.39} \approx 0.77$

It is interpreted similarly to R²: "Our model's features explain about **77%** of the 'deviance' (uncertainty) in the outcome, compared to a model that just guesses the average."

Important: Pseudo R² cannot be compared to linear regression R². Use it only for comparing logistic models.

Predictive Evaluation Metrics

When using logistic regression for prediction, we can apply standard classification metrics beyond Pseudo R^2:

Accuracy

Overall proportion of correct predictions across all classes

Precision

Of predicted positives, how many are truly positive

Recall

Of actual positives, how many did we correctly identify

F1 Score

Harmonic mean of precision and recall

Confusion Matrix

Complete breakdown of true/false positives and negatives

Logistic Regression in Python

Dep. Variable:	cl	**No. Observations:**	683
Model:	Logit	**Df Residuals:**	676
Method:	MLE	**Df Model:**	6
Date:	Sat, 15 Nov 2025	**Pseudo R-squ.:**	0.8788
Time:	16:22:57	**Log-Likelihood:**	-53.572
converged:	True	**LL-Null:**	-442.18
Covariance Type:	nonrobust	**LLR p-value:**	1.294e-164

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-9.7671	1.085	-9.001	0.000	-11.894	-7.640
V1	0.6225	0.137	4.540	0.000	0.354	0.891
V3	0.3495	0.165	2.118	0.034	0.026	0.673
V4	0.3375	0.116	2.920	0.004	0.111	0.564
V6	0.3786	0.094	4.035	0.000	0.195	0.562
V7	0.4713	0.166	2.837	0.005	0.146	0.797
V8	0.2432	0.109	2.240	0.025	0.030	0.456

Conclusions and Next Steps

We Have Explored:

- Benefits of parametric models for classification
- Limits of linear regression for classification
- The logistic regression model
- Statistical interpretation of the coefficients of a logistic regressor
- Evaluating a logistic regressor
- Examples

In the next lectures, we will look at multiclass logistic regression

References

- Chapter 4 of [1]

[1] James, Gareth Gareth Michael. An introduction to statistical learning: with applications in Python, 2023.https://www.statlearning.com