Classification

Supervised learning

- In supervised learning, you have a target (a thing that you want to predict)
  - AI is extremely good at solving something extremely narrow
  - It can only take very limited range of variables and provide a very limited answer to a very limited problem
    - Predicting default vs. not default
  - We call this supervised learning because we know the outputs for a set of points beforehand. Or else it would be called unsupervised learning.

Regression for Classification

- Logistic regression was not created to model probabilities at first, it was created to model populations

Customer	Age	Income	Gender	G/B
John	30	1200	M	B
Sarah	25	800	F	G
Sophie	52	2200	F	G
David	48	2000	M	B
Peter	34	1800	M	G

→

Customer	Age	Income	Gender	G/B	Y
John	30	1200	M	B	0
Sarah	25	800	F	G	1
Sophie	52	2200	F	G	1
David	48	2000	M	B	0
Peter	34	1800	M	B	1

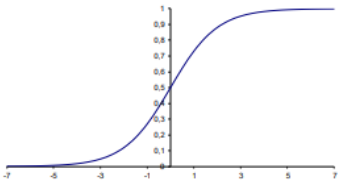
Linear regression gives: Y = β0 + β1Age + β2Income + β3Gender

Can be estimated using OLS

Two problems:

- No guarantee that Y is between 0 and 1 (i.e. a probability)
- Target/Errors not normally distributed
- Using a bounding function to limit the outcome between 0 and 1:

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



- Logistic regression is basically a conditional probability estimation over test set
  - What is does is saying what's the probability of you having one characteristic
    - E.g. customer being good or customer being bad

Linear regression with a transformation such that the output is always between 0 and 1 and thus can be interpreted as a probability (e.g. probability of good customer)

$$P(\text{customer= good}\,|\,\text{age, income, gender,..}) = \frac{1}{1+e^{-(\beta_0+\beta_1 age+\beta_2 income+\beta_3 gender...)}}$$

Or, alternatively

$$\ln\left(\frac{P(\text{customer} = \text{good}\,|\,\text{age, income, gender,...})}{P(\text{customer} = \text{bad}\,|\,\text{age, income, gender})}\right) = \beta_0 + \beta_1 age + \beta_2 income + \beta_3 gender + ...$$

Once the model has been estimated using historical data, we can use it to score or assign probabilities to new data

Logistic regression model

$$Y = \{0,1\}$$

$$p = P(Y = 1\,|\,X_1,..., X_n) = \frac{1}{1+e^{-(b_0+b_1X_1+...+b_nX_n+e)}} = \frac{e^{b_0+b_1X_1+...+b_nX_n+e}}{1+e^{b_0+b_1X_1+...+b_nX_n+e}}$$

Logit

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = b_0 + b_1X_1 + ... + b_nX_n + e$$

Odds

$$\frac{p}{1-p} = \exp(b_0 + b_1X_1 + ... + b_nX_n + e)$$

- Note that this is still a linear model
  - That's why we still call this a transparent model
    - Because we can see that every variable has a straight impact in the capacity to predict of our model
    - Given that we are using WoE, this is going to be simplified

Odds Ratio

$O_A = \frac{P(A)}{1-P(A)}$ – the odds of the event A

$O_B = \frac{P(B)}{1-P(B)}$ – the odds of the event B

$OR_{AB} = \frac{O_A}{O_B} = \frac{P(A)}{1-P(B)} \div \frac{P(B)}{1-P(B)}$ – the odds ratio

$$\frac{p}{1-p} = \exp(b_0 + b_1 X_1 + \ldots + b_n X_n + e)$$

a new value of X_1

$$\frac{p^*}{1-p^*} = \exp(b_0 + b_1 X_1^* + \ldots + b_n X_n + e)$$

$$OR = \frac{p^*}{1-p^*} \div \frac{p}{1-p} = \frac{\exp(b_0 + b_1 X_1^* + \ldots + b_n X_n + e)}{\exp(b_0 + b_1 X_1 + \ldots + b_n X_n + e)} =$$

$$= \exp(b_1 X_1^* - b_1 X_1) = (\exp(b_1))^{X_1^* - X_1}$$

odds ratio only depends on the difference between the old and new values of X_1

If $X_1^* - X_1 = 1$ then $OR = \exp(b_1)$

- Learning in logistic regression depends on the differences between the elements
  - The elements are going to be different by a whole bunch of characteristics
  - So the final odds in comparison is going to be
    - You are higher on this one, that gives you higher points
    - You are lower on that one, then that lowers your points
    - And you multiply those odds to get your comparison against the baseline
  - What's why logistic regression is also a relative model
    - The probabilities are not calibrated
    - Because what logistic regress is doing is giving you the ratio in comparison to the average value in the sample
    - So it tells you whether you are above or below the risk of the average of the sample

Example

The odds ratio estimate equals 1.945 for age.
It means that if the customer's age increases by one year,
the odds of defaulting increase almost 2 times (provided that all other customer's characteristics are the same)

The odds ratio estimate equals 0.358 for income.
It means that if the customer's income increases by $1000, the odds of defaulting decrease almost 3 times (provided that all other customer's characteristics are the same)

- Note that betas from logistic regression using WoE always have the same sign
  - Because the sign of WoE already tells the trend of the variable
  - However, if the variables are correlated, signs are expected to shift

Maximum Likelihood

- We use maximum likelihood to train the model

Sample with logistic estimators:

$$p(x_i) = p(y = 1|x_i) = \frac{\exp(\beta_0 + \sum_{j=1}^{V} \beta_j x_{i,j})}{1 + \exp(\beta_0 + \sum_{j=1}^{V} \beta_j x_{i,j})}$$

We use máximum likelihood to estimate it:

$$\max_{\beta} \mathcal{L}(X) = \left(\prod_{i \backslash y_i = 1} p(x_i) \right)^{n_1} \left(\prod_{i \backslash y_i = 0} (1 - p(x_i)) \right)^{N - n_1}$$

- n_1: Number of positive cases.
- N: Total number of cases.

Beta parameters are the outputs of this problem.

Assumption: Sample and cases independent. **No distributional assumptions**!

- If we know that someone is a one
  - Then p would be higher

- Properties of maximum likelihood
  - In the infinite, logistic regression is just as good as a random forest and a deep learning model
  - In theory, those models converge to be perfect with infinite amount of data, but a more sophisticated model will get there faster than logistic regression
  - Betas are normally distributed

Functional invariance: Be it θ a parameter, $\hat{\theta}$ its estimate and $f(\cdot)$ a function.

$$y = f(\theta) \Rightarrow \hat{y} = f(\hat{\theta})$$

- Estimate of parameter function is the function of the estimate.

Asymptotic properties:

- Estimate is asymptotically unbiased.
- Estimate is asymptotically efficient.
  - Efficient: Reaches Cramer – Rao lower bound as sample size goes to infinity (theoretically mínimum square error over every function possible).
- Estimate is asymptotically normally distributed.

- When we want to estimate confidence intervals for our estimates
  - We should not use p-values
    - If we have a gigantic population, you can find patterns at a 5% just by mere chance
    - The combinatorial is so big that we can pretty much find whatever we want inside that data
    - We will calculate empirical confidence intervals for our measures
      - And those are used to define operative ranges for our models
  - We will measure uncertainty in different ways, we will bootstrap, sample, cross validate

Penalized Regression

Bias-Variance Trade-off

- For a regression, the more variable you have, the more information can be included in the model, so the accurate the prediction.
- But the extra variable will increase dispersion and variability
- We want to find the group of variables that give us low bias and low variance
- This is why WoE is a superior transformation than dummy variables
  - ➢ Dummy variables create a lot more variables, increase the variance
  - ➢ Weight of evidence already qualifies all the information that you need

- There are three components in the error
  - ➢ Biase + variance
  - ➢ And irreducible error, an amount that won't be reduced no matter how much information that we have because they come from the fact that we cannot control for what we don't know in the future

- The idea of penalized regression
  - ➢ Explicitly model simultaneously via some variable until we find the level of error that we can be happy with

Lasso, Ridge and Elastic-Net

- To make a bias-variance tradeoff explicit model, we need to explicitly include the variance of the model
  - ➢ The likelihood is only about bias

We can add a **penalization** to the likelihood to force the model to reduce complexity.

- LASSO penalty:

$$\text{Error} = \text{Likehood (bias)} + \lambda \sum |\beta_i|$$

- Ridge penalty:

$$\text{Error} = \text{Likehood (bias)} + \lambda \sum \beta_i^2$$

- Elastic Net:

$$\text{Error} = \text{Likehood (bias)} + \lambda \left(\frac{1-\alpha}{2} \sum \beta_i^2 + \alpha \sum |\beta_i| \right)$$

- Here λ is the weight parameter of the likelihood vs the penalty and α the weight between the ridge and lasso regressions.

  - ➢ We now will minimize the balance between the bias, which is the likelihood, and the variance, which is the penalty on the complexity of the modal
  - ➢ Ridge penalty is very bad at actually eliminating variables
  - ➢ Lasso can remove variables but it can't deal with correlations, ridge can deal with correlation
    - ✧ Ridge takes two perfectly correlated variables and take the beta split it between the two, so

two beta parameters where each have half the weight,

- ✧ Unbalanced correlations will balance the beta out, so even with WoE, the betas can have negative signs

- Which should I choose
  - ➢ Lasso is good at eliminating useless variables, but it is harder to optimize, it does not work well with correlation
    - ✧ Never use lasso when there are more variables than examples
  - ➢ Ridge converge faster because it is parabolic and well formed, deals with correlated variables perfectly, not good at eliminating variables
    - ✧ Danger: the betas will be split between the correlated variables, betas not interpretable by themselves
  - ➢ Elastic net, need to deal with extra parameter, deal with correlation and variable selection at the same time
    - ✧ But sometimes, it will not do either of things
  - ➢ Use Lasso if can, require computational power, or go elastic net.