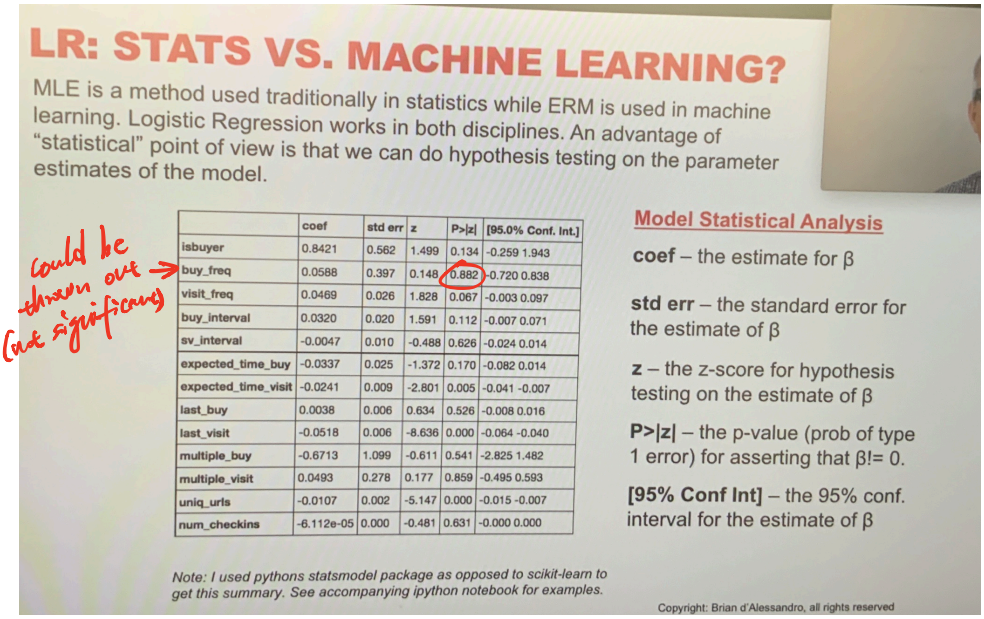
$$P(C_i|x) = f(x) = \frac{1}{1+e^{-(\alpha+\beta x)}}$$



LR: STATS VS. MACHINE LEARNING?

MLE is a method used traditionally in statistics while ERM is used in machine learning. Logistic Regression works in both disciplines. An advantage of "statistical" point of view is that we can do hypothesis testing on the parameter estimates of the model.

could be thrown out (not significant)

	coef	std err	z	P>|z|	[95.0% Conf. Int.]
isbuyer	0.8421	0.562	1.499	0.134	-0.259 1.943
buy_freq	0.0588	0.397	0.148	0.882	-0.720 0.838
visit_freq	0.0469	0.026	1.828	0.067	-0.003 0.097
buy_interval	0.0320	0.020	1.591	0.112	-0.007 0.071
sv_interval	-0.0047	0.010	-0.488	0.626	-0.024 0.014
expected_time_buy	-0.0337	0.025	-1.372	0.170	-0.082 0.014
expected_time_visit	-0.0241	0.009	-2.801	0.005	-0.041 -0.007
last_buy	0.0038	0.006	0.634	0.526	-0.008 0.016
last_visit	-0.0518	0.006	-8.636	0.000	-0.064 -0.040
multiple_buy	-0.6713	1.099	-0.611	0.541	-2.825 1.482
multiple_visit	0.0493	0.278	0.177	0.859	-0.495 0.593
uniq_urls	-0.0107	0.002	-5.147	0.000	-0.015 -0.007
num_checkins	-6.112e-05	0.000	-0.481	0.631	-0.000 0.000

Model Statistical Analysis

coef – the estimate for β

std err – the standard error for the estimate of β

z – the z-score for hypothesis testing on the estimate of β

P>|z| – the p-value (prob of type 1 error) for asserting that β!= 0.

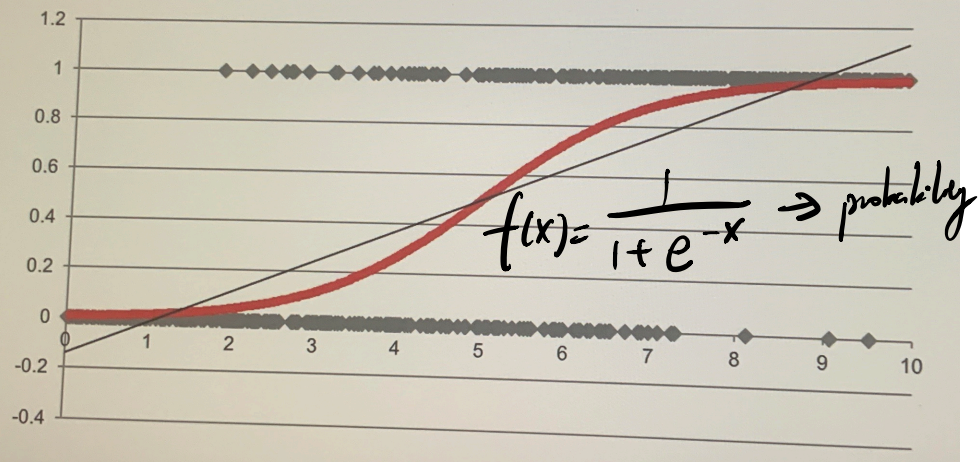
[95% Conf Int] – the 95% conf. interval for the estimate of β

Note: I used pythons statsmodel package as opposed to scikit-learn to get this summary. See accompanying ipython notebook for examples.

- $|\beta_1| > |\beta_2|$ does not guarantee that feature X_1 is more predictive than X_2. The magnitude of β is inversely proportional to the scale of X, so comparing betas only makes sense when the features have the same scale. (such as binary)
- Sign (β) does tell you whether Y is positively or negatively correlated with X. However if the feature have a lot of multi-collinearity, sign(β) can be misleading
- Multi-collinearity in X means the betas will have covariance with each other. The betas will "split" the effects. Sometimes they will split the effect as positive numbers ($1 = 0.5 + 0.5$) and other times they will split as negative ($1 = -1 + 2$) This makes interpreting β much more difficult.

SOMETHING BETTER?

It would be better if we had some function that was linear in its parameters, but behaved better as a probability estimator.



$$P(C|X) = f(X) = \frac{1}{1+e^{-(\alpha+\beta X)}}$$

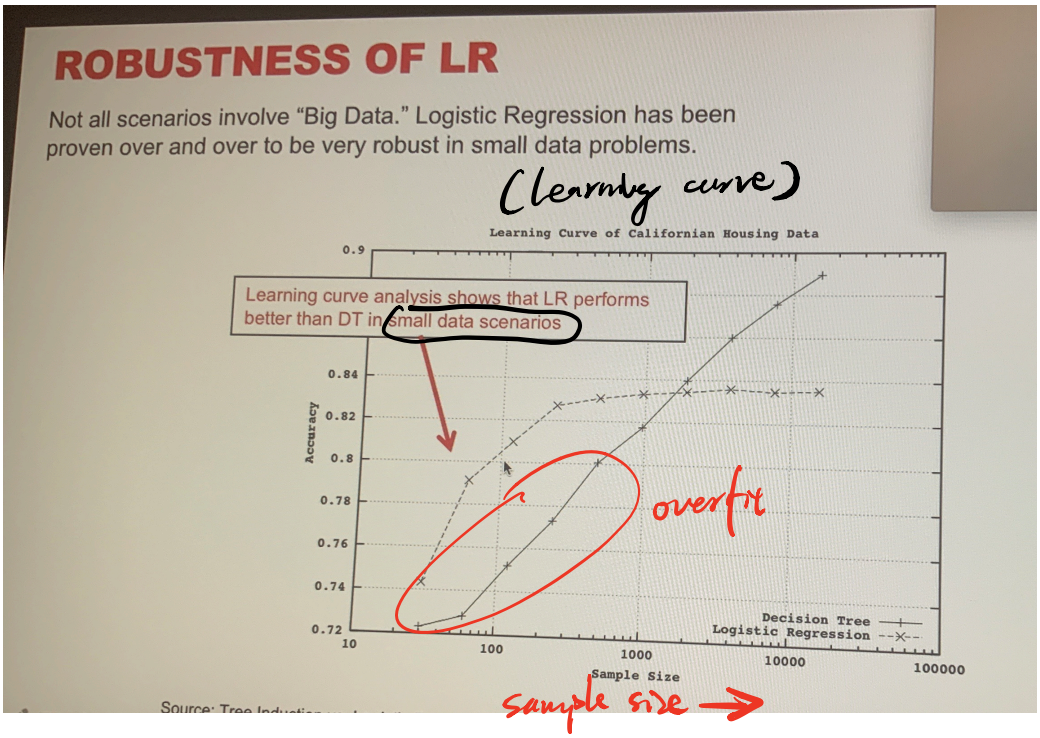
$$\ln \frac{P(C|X)}{1-P(C|X)} = \ln \frac{1}{1+e^{-(\alpha+\beta X)}} = \alpha + \beta X$$

Essentially we are fitting a linear model to the log-odds of $P(C|X)$

What does β actually mean?

Recall that $\ln \frac{p}{1-p} = \alpha + \beta X$. So one unit change in x changes the log odds by the value β.

Logistic Regression learning curve



SVM:

If the data is not in the same scale and not choose the right regularization parameter, it could take long time to train.

SVM is one of the most powerful non-linear classifier, and Gaussian kernel has the most flexibility and most power in capturing non-linear boundary.