

Week 4

EE4708 - Data Analytics Laboratory 2020

Table of Contents


1. Introduction to Categorical Encoding
2. Types of Categorical Encoding
3. Overfitting vs Underfitting
4. Bias vs Variance
5. Need for Regularization
6. Ridge Regression
7. Lasso Regression
8. Need for Feature Selection
9. How Lasso Regression used as feature selection

Introduction to Categorical Encoding

- Since many ML algorithms rely on measuring distances to find optimal model, the features have to take numeric values to apply such algorithms.
- Therefore categorical features have to be encoded to take numeric values which is referred to as “**Categorical Encoding**”
 - a. **Label Encoding** : This method is used for ordinal categorical features i.e which has order among the categories of that feature. Here, each label is assigned a unique integer based on different aspects like alphabetical order, frequency of that particular category, etc.
 - b. **One hot Encoding** : This method is used for nominal categorical features i.e which doesn't have order among the categories. Here every unique value in the category is added as an additional feature.

Label-Encoding Example

ID	Feature1
0	L
1	XL
2	L
3	L
4	XL



ID	Feature1
0	1
1	2
2	1
3	1
4	2

As we know $XL > L$, so this Feature_1 has order among its categories.
So we replace 1 with L, 2 with XL.

One Hot Encoding Example

origin		origin=usa	origin=europe	origin=japan
usa		1	0	0
usa		1	0	0
europe		0	1	0
...	
usa		1	0	0
japan		0	0	1
japan		0	0	1

There is no order among countries, so this is nominal feature. So, we apply One hot encoding i.e add unique values as features.

Note : Beware of **dummy variable trap**. Refer to this link for more details

<https://www.geeksforgeeks.org/ml-dummy-variable-trap-in-regression-models/>

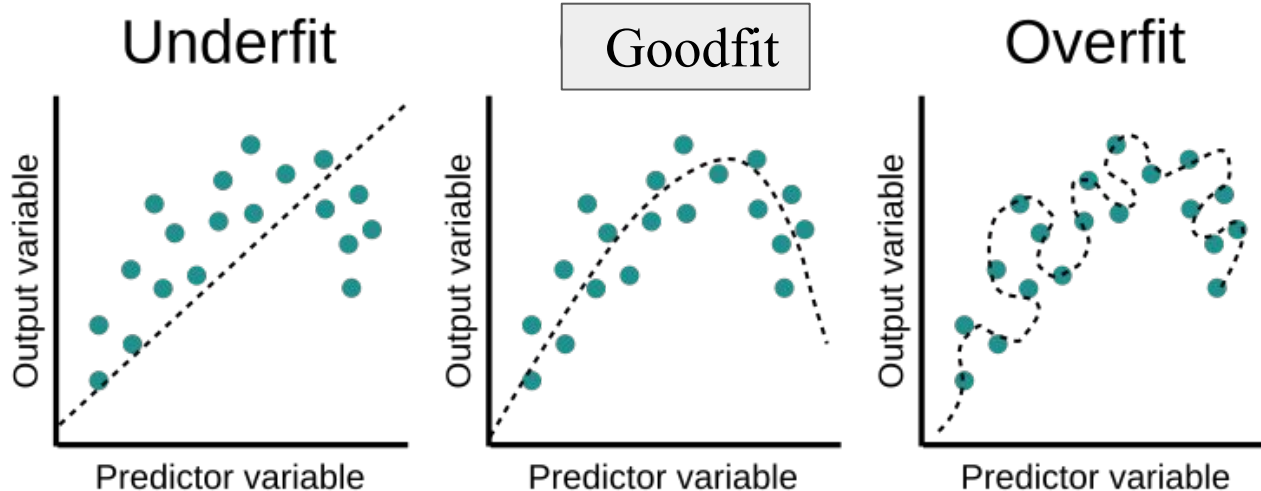
Generalization and Goodness of Fit

- ❖ Goal of machine learning is to learn a model that generalizes well from the training data and is able to make predictions on never seen data
- ❖ A function which best achieves this goal is referred to as target function
- ❖ Since supervised learning fits functions or models to data, goodness of fit refers to how well the learnt function approximates the target function
- ❖ Poor fitting is due to one of the following:
 - Overfitting
 - Underfitting
- ❖ Refer this link for multiple ways of measuring how good a regression fit is:

https://en.wikipedia.org/wiki/Goodness_of_fit

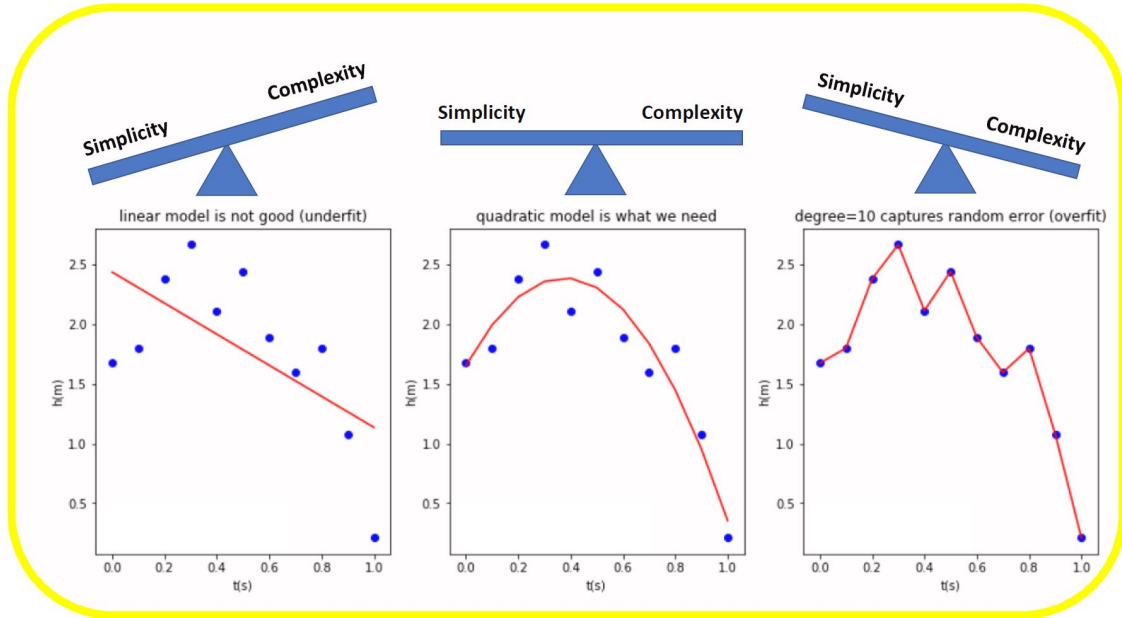
Overfitting vs Underfitting

- **Overfitting** refers to a model which learns the training data too well (including noise) and as a result it might be unable to generalise. Hence it performs poorly on unseen test data.
- **Underfitting** refers to a model which has not learned enough from training data and it performs poorly both on training as well as unseen test data



Model Complexity

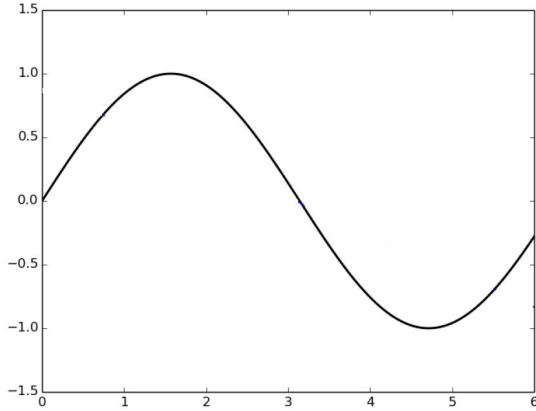
- Model complexity is a measure of how simple or complex the model is and generally varies in terms of the number of features or number of parameters
- It depends on the assumptions that are made when training the model such as:
 - Linearity assumption
 - Polynomial assumption
 - Number of features chosen



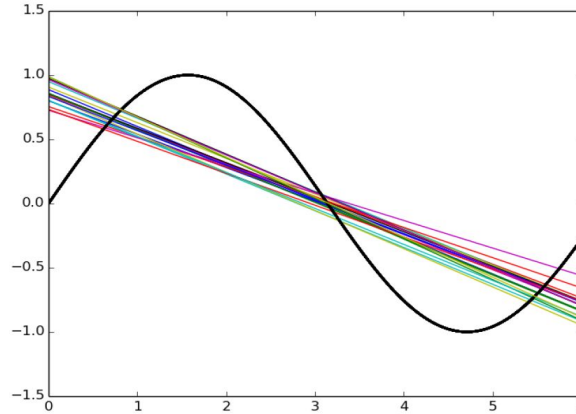
Source:

<https://towardsdatascience.com/simplicity-vs-complexity-in-machine-learning-finding-the-right-balance-c9000d1726fb>

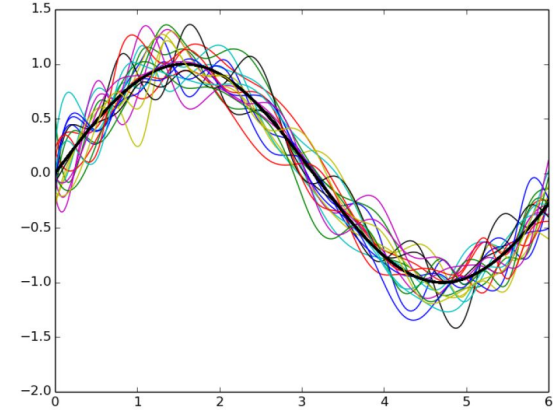
Bias Vs Variance



- Suppose the true model is sinusoidal as shown
- Let multiple training datasets be drawn from this model



- Suppose regression models are learnt using all the datasets assuming simple linear model
- Predictions (test) of the models do not vary much (low variance)
- Predictions are very far from true values (high bias)



- Suppose regression models are learnt using all the datasets assuming high degree polynomial model
- Predictions (test) of the models are very different from each other (high variance)
- Predictions are not very far from true values on average (low bias)

Bias Vs Variance

- **Bias** refers to the error in predictions introduced by approximating a complex model by a much simpler model.
- **Variance** refers to the amount by which predictions vary when different models are trained using a different training datasets, all drawn from the same underlying model of distribution.
- A highly biased model choice results in underfitting
- A high variance model choice results in overfitting
- Ideally, low bias and low variance is what is required but in most of the cases, it is not possible to minimize both of them simultaneously
- A good balance between bias and variance is required while deciding the complexity of a model to be trained

Formula for Bias & Variance

$f(x)$: Represent true values of output (unknown) for input x

$\hat{f}(x)$: Represents predicted values of output given by a learnt model for Input x

Bias: $\mathbb{E}[\hat{f}(x)] - f(x)$

Variance: $\mathbb{E}[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2]$

Note: The expectation is over different models (not structurally different but different parameters) that can be learnt

5.3. Bias vs Variance Tradeoff Derivation

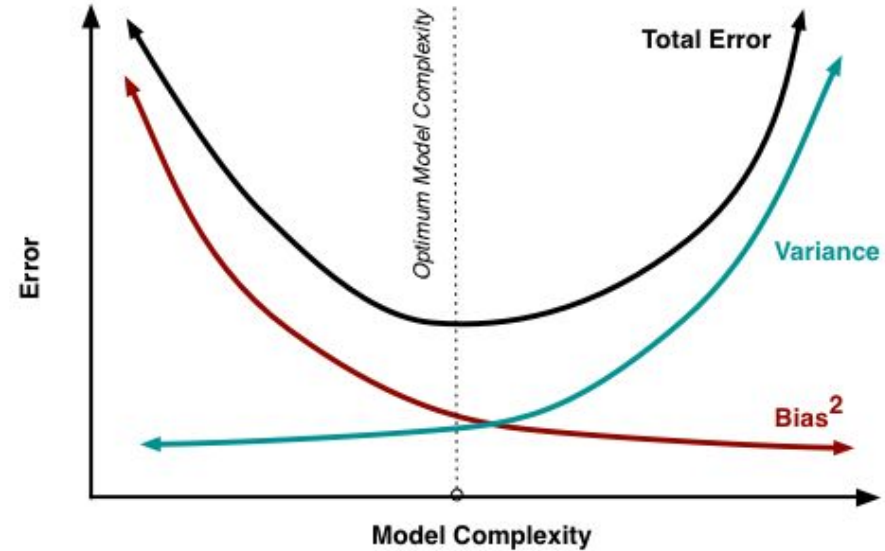
$$\begin{aligned}\mathbb{E}[(f(x) - \hat{f}(x))^2] &= \mathbb{E} \left[\left((f(x) - \mathbb{E}[\hat{f}(x)]) - (\hat{f}(x) - \mathbb{E}[\hat{f}(x)]) \right)^2 \right] \\&= \mathbb{E} \left[\left(\mathbb{E}[\hat{f}(x)] - f(x) \right)^2 \right] + \mathbb{E} \left[\left(\hat{f}(x) - \mathbb{E}[\hat{f}(x)] \right)^2 \right] \\&\quad - 2\mathbb{E} \left[\left(f(x) - \mathbb{E}[\hat{f}(x)] \right) \left(\hat{f}(x) - \mathbb{E}[\hat{f}(x)] \right) \right] \\&= \underbrace{(\mathbb{E}[\hat{f}(x)] - f(x))^2}_{=\text{bias}[\hat{f}(x)]} + \underbrace{\mathbb{E} \left[\left(\hat{f}(x) - \mathbb{E}[\hat{f}(x)] \right)^2 \right]}_{=\text{var}(\hat{f}(x))} \\&\quad - 2 \left(f(x) - \mathbb{E}[\hat{f}(x)] \right) \mathbb{E} \left[\left(\hat{f}(x) - \mathbb{E}[\hat{f}(x)] \right) \right] \\&= \text{bias}[\hat{f}(x)]^2 + \text{var}(\hat{f}(x)) \\&\quad - 2 \left(f(x) - \mathbb{E}[\hat{f}(x)] \right) \left(\mathbb{E}[\hat{f}(x)] - \mathbb{E}[\hat{f}(x)] \right) \\&= \text{bias}[\hat{f}(x)]^2 + \text{var}(\hat{f}(x))\end{aligned}$$

Note :

LHS refer to expected test MSE i.e average test MSE we would get if we repeat estimate f using different datasets

Model Complexity - Bias vs Variance

- Training error only reflects bias.
- Test error reflects bias and variance.
- This is why we can't see overfitting from training error.
- Test error goes down, then back up as variance increases



Regularization

- Overfitting during training prevents the model to generalise to perform well on unseen data
- To prevent overfitting, regularization is performed
- Regularization refers to the process of regularizing the parameters or coefficients by adding a penalising term to the cost function
- If needed, regularisation can be used to drive some of the parameters close to zero to eliminate some features or basis functions from the model

Ridge Regression(L2 Regularisation)

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}$$

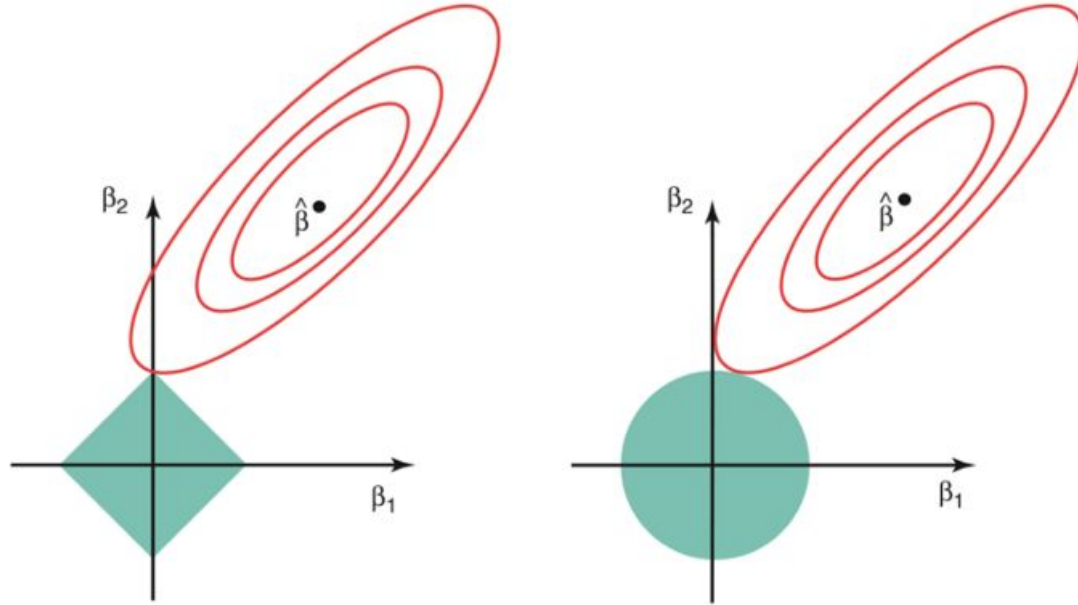
- “**Lambda**” is known as “**Regularization parameter**”. Intuitively, this lambda is the parameters which needs to be tuned to control the penalising factor
- When the value of lambda = 0, it becomes simple linear regression.
- As the value of lambda increases, the coefficients are penalised further and one needs to determine an optimal value of lambda by observing the MSE on validation or test set
- Ridge regression might drive some parameters close to zero but never absolute zero (This is the main difference with lasso regression)

Lasso Regression(L1 Regularisation)

$$\hat{\beta}^{\text{lasso}} = \arg \min_{\beta} \sum_{n=1}^N \frac{1}{2} (y_n - \beta x_n)^2 + \lambda \sum_{i=1}^p |\beta_i|$$

- LASSO (Least Absolute Shrinkage Selector Operator), is quite similar to ridge. Difference between them is the penalising factor for Lasso is based on L1 norm while for ridge regression, it is based on L2 norm
- Lasso can drive the coefficients of some features to absolute zero
- Therefore, Lasso tends to give sparser solutions
- This property of Lasso can be used in feature selection

Why Lasso gives sparser solution?



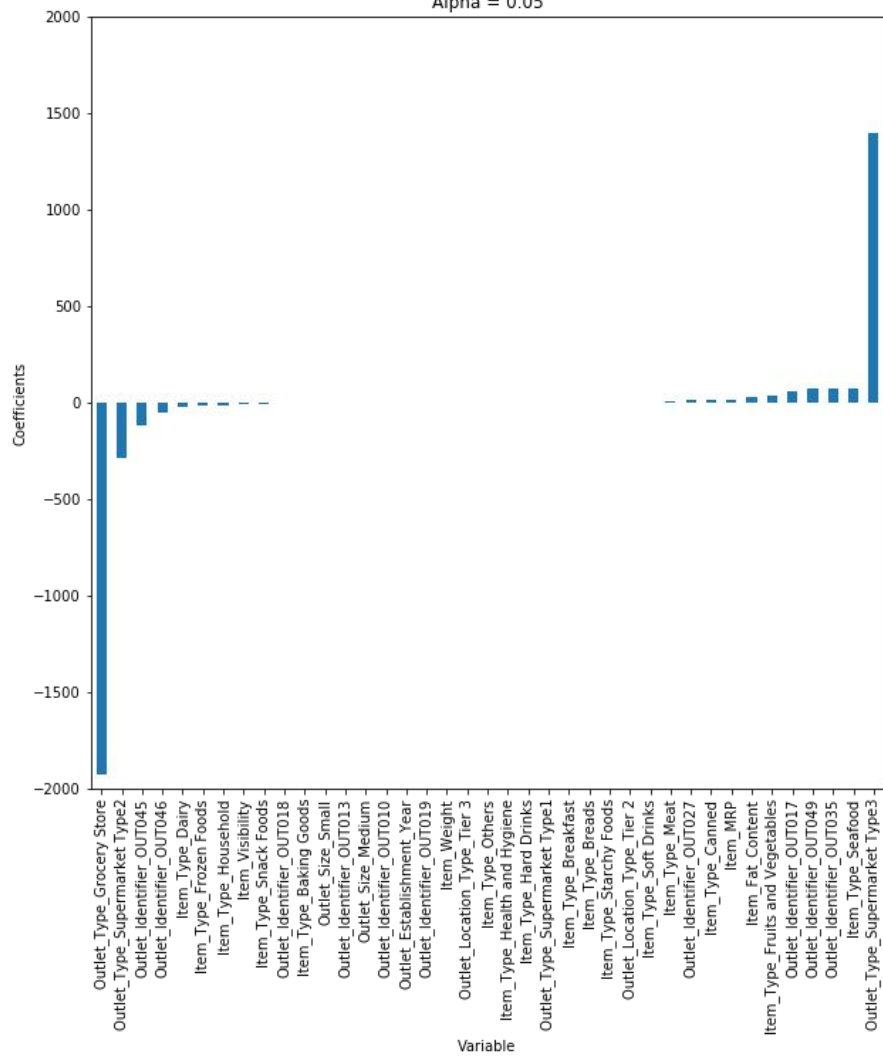
Note :

1. Red curves refer to MSE
2. Green diamond refers to L1 term in the cost function
3. Green circle refers to L2 term in the cost function

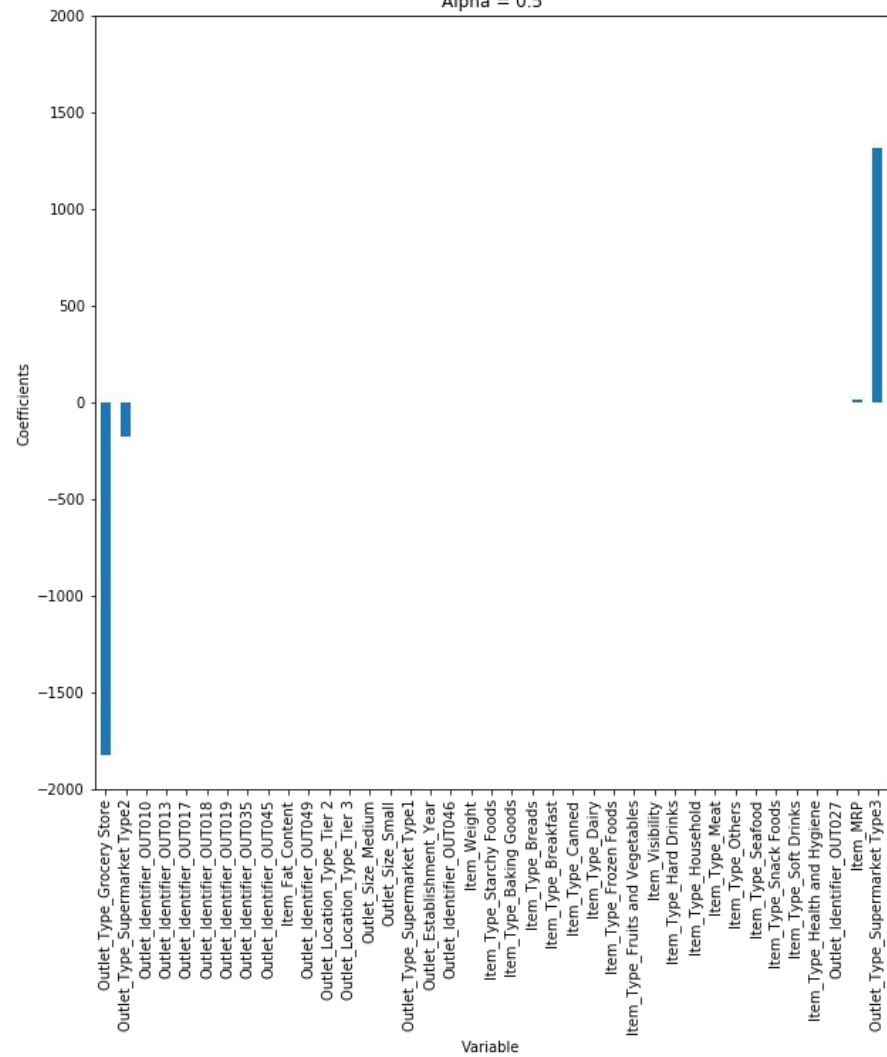
Feature Selection

- Feature selection is relevant when the number of given features is high and not all of them might be relevant for the model
- In such a case, it would be highly inefficient to use all the features and it would also result in overfitting
- Therefore, there is a need to select the right set of feature which give us an a good fit model
- There are different techniques like wrapper based and filter based for feature selection (Not covered in this course)
- As we discussed earlier, Lasso regression can be used as feature selection technique by eliminating features with zero or close to zero coefficients

Alpha = 0.05



Alpha = 0.5



Conclusions

Regression = { Prediction }

Regression + Ridge = { Prediction } + { Regularization }

Regression + Lasso = { Prediction } + { Regularization } + { Feature Selection }