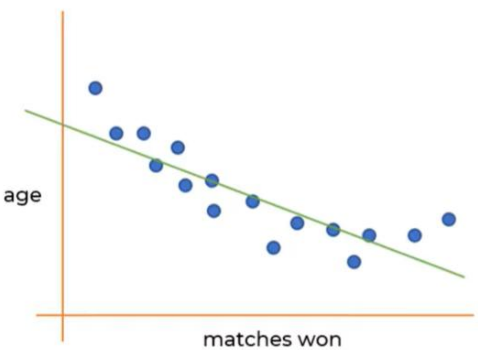
Ridge and Lasso Regression

Dr. Dipak Kumar Mohanty
KIIT University, Bhubaneswr

underfit

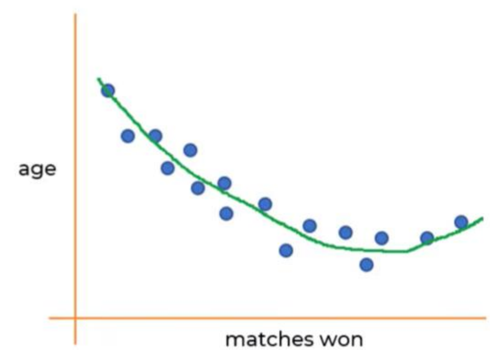


$$\text{match won} = \theta_0 + \theta_1 * age$$

overfit

$$\text{match won} = \theta_0 + \theta_1 * age \quad + \theta_2 * age^2 + \theta_3 * age^3 \ + \theta_4 * age^4$$

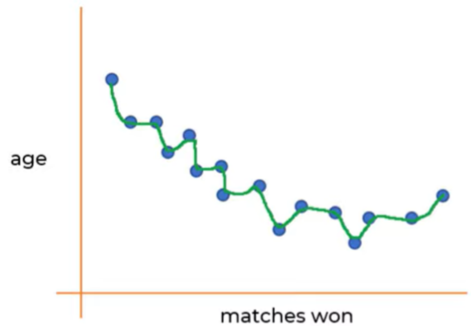
balanced fit



$$\text{match won} = \theta_0 + \theta_1 * age \quad + \theta_2 * age^2$$

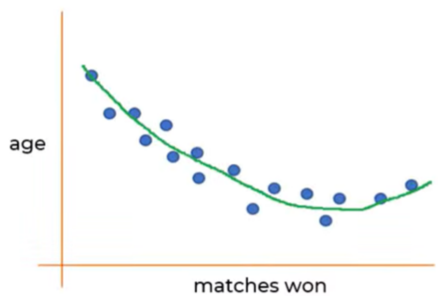
How to reduce overfitting?

$$\text{match won} = \theta_0 + \theta_1 * age \quad + \theta_2 * age^2 + \theta_3 * age^3 + \theta_4 * age^4$$



$$\text{match won} = \theta_0 + \theta_1 * age \quad + \theta_2 * age^2 + \theta_3 * age^3 + \theta_4 * age^4$$

Try to make θ_3 and θ_4 almost close to zero



$$\text{match won} = \theta_0 + \theta_1 * age \quad + \theta_2 * age^2$$

From Linear Regression we know, MSE...

Mean Squared Error

$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_{predicted}\right)^2$$

Mean Squared Error

$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h_\theta(x_i)\right)^2$$

$$h_\theta(x_i) = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3$$

Regularization

- Regularizations are of two types
- 1) Ridge Regression or L2- Regularization
- 2) Lasso Regression or L1- Regularization

1) Ridge Regression or L2-Regularization

Ridge Regression

$$R = \text{Loss} + \lambda \|\theta\|_2^2$$

where λ = penalty,

where $\|\theta\|_2^2 = \theta_1^2 + \theta_2^2 + \dots + \theta_n^2$

Here, When θ is bigger, Error become bigger, so model will not converge

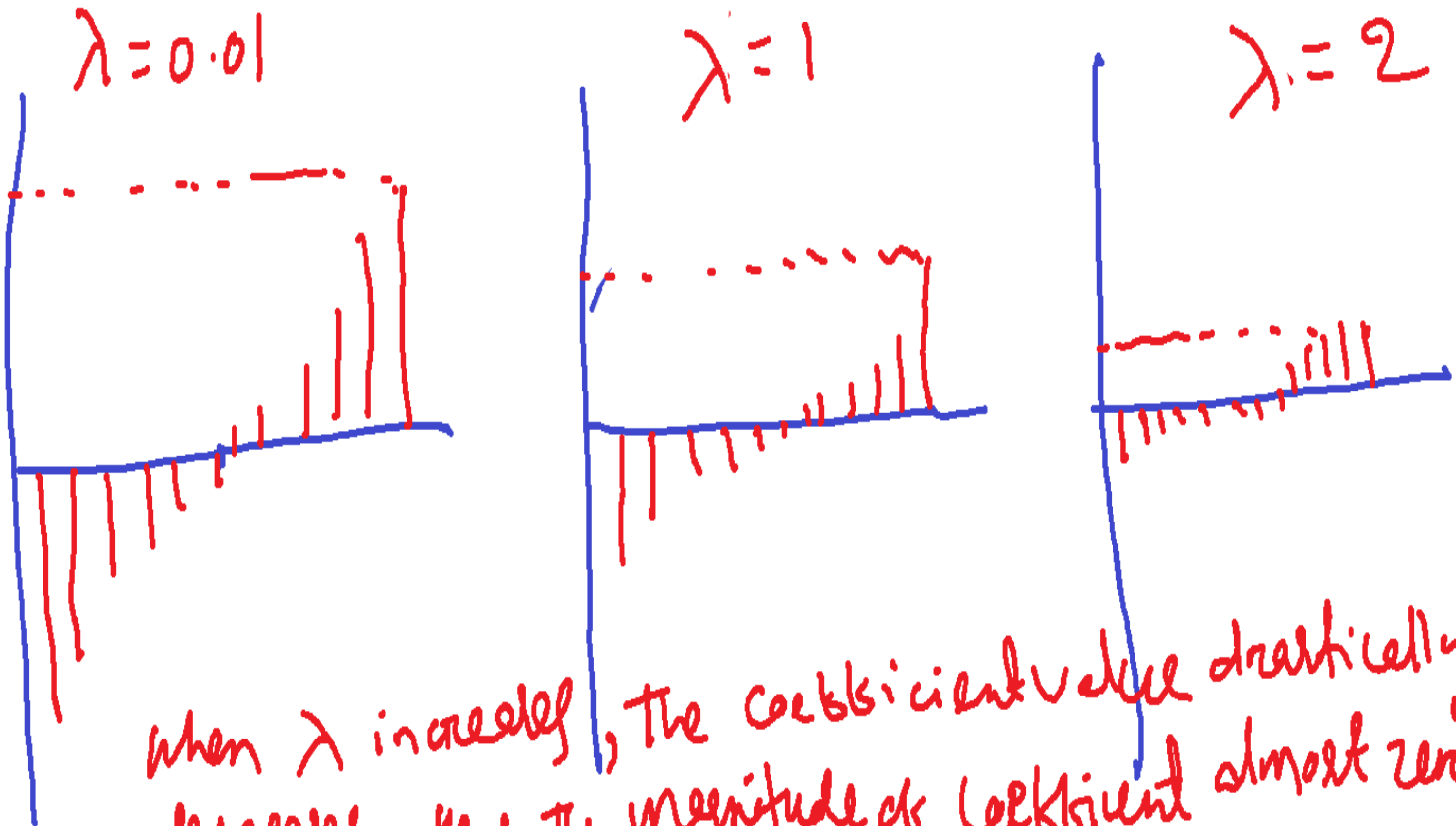
L2 Regularization

$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h_\theta(x_i)\right)^2 + \lambda \sum_{i=1}^{n} \theta_i^2$$

$$h_\theta(x_i) = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3$$

- Here, Penalty Lambda is used to ensure that θ value doesn't go too high.
- When Lambda is bigger, θ is smaller and vice versa.
- This appraoch is called L2-Regularization

Ridge: Example



when λ increases, the coefficient value drastically decreases, here the magnitude of coefficient almost zero.

L1-Regularization or Lasso Regression

$$\text{Lasso } R = \text{Loss} + \lambda \|\theta\|_1$$

$$\|\theta\|_1 = |\theta_1| + |\theta_2| + \cdots + |\theta_n|$$

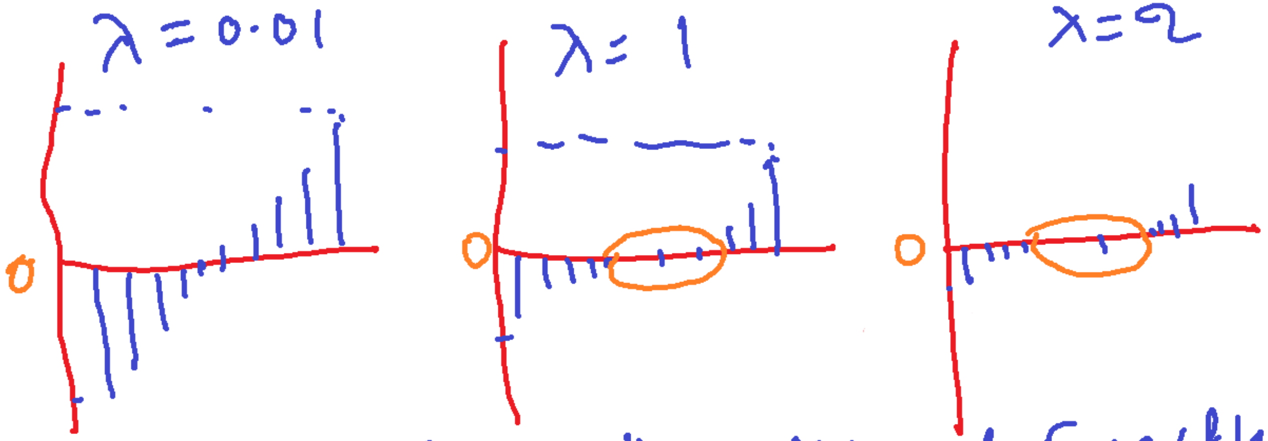
L1-Regularization

L1 Regularization

$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h_\theta(x_i)\right)^2 + \lambda \sum_{i=1}^{n}|\theta_i|$$

$$h_\theta(x_i) = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3$$

Lasso Regression: Example



Here the magnitude of the coefficient <u>Exactly zero</u>

Example

Let $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

$$y = 0.8 + 1.2x_1 + 30x_2 + 49x_3$$

After Ridge

$$y = 0.8 + 1.2x_1 + 0.2x_2 + 0.3x_3$$

After lasso

$$y = 0.8 + 1.2x_1 + 0 \cdot x_2 + 0 \cdot x_3$$

jupyter Untitled Last Checkpoint: 18 minutes ago (unsaved changes)

	Rooms	Propertycount	Distance	Bedroom2	Bathroom	Car	Landsize	BuildingArea	Price	Suburb_Aberfeldie	...	CouncilArea_Moorabool Shire Council	CouncilAre
1	2	4019.0	2.5	2.0	1.0	1.0	202.0	160.2564	1480000.0	0	...	0	
2	2	4019.0	2.5	2.0	1.0	0.0	156.0	79.0000	1035000.0	0	...	0	
4	3	4019.0	2.5	3.0	2.0	0.0	134.0	150.0000	1465000.0	0	...	0	
5	3	4019.0	2.5	3.0	2.0	1.0	94.0	160.2564	850000.0	0	...	0	
6	4	4019.0	2.5	3.0	1.0	2.0	120.0	142.0000	1600000.0	0	...	0	

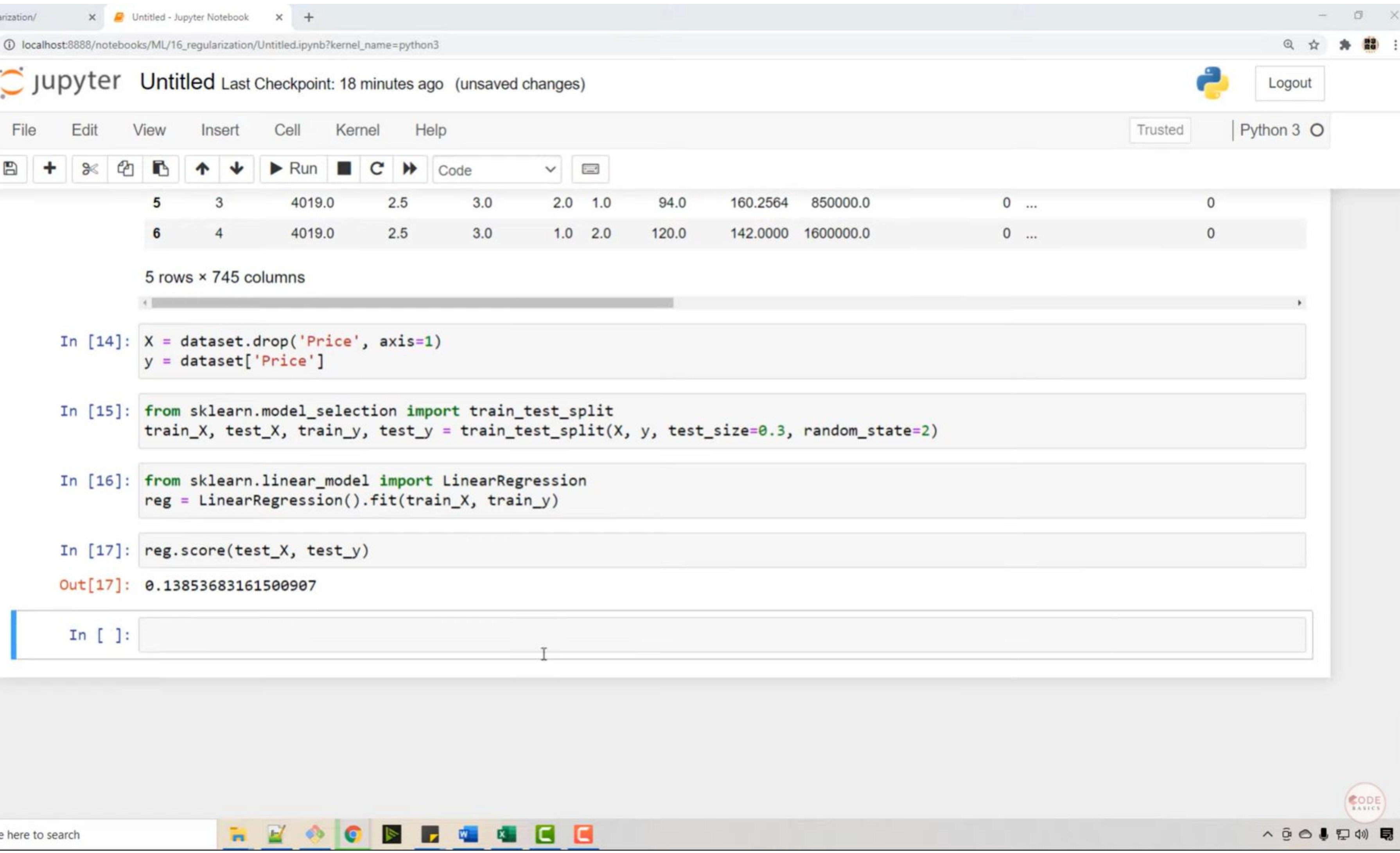
5 rows × 745 columns

In [14]:
```
X = dataset.drop('Price', axis=1)
y = dataset['Price']
```

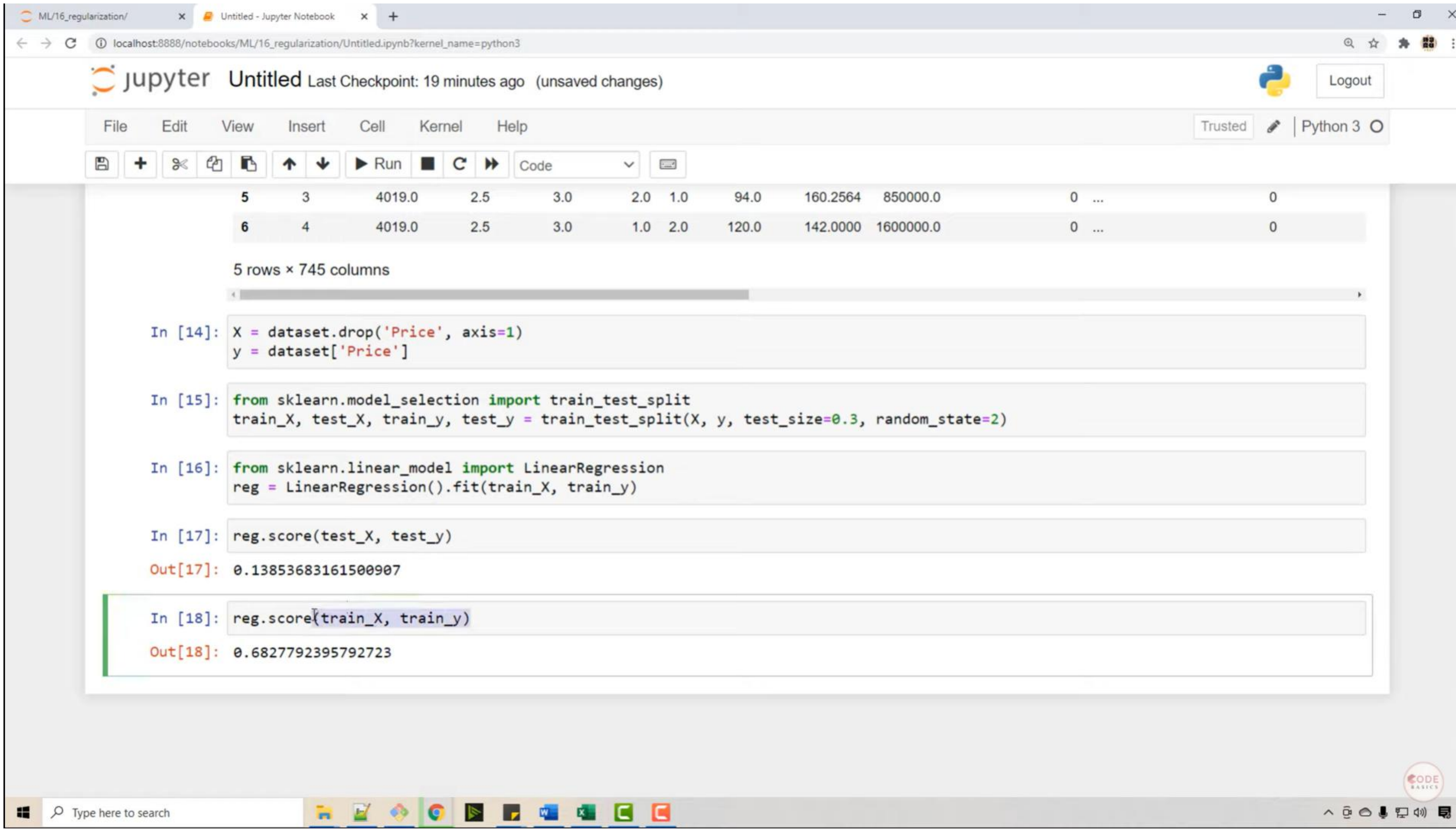
In [15]:
```
from sklearn.model_selection import train_test_split
train_X, test_X, train_y, test_y = train_test_split(X, y, test_size=0.3, random_state=2)
```

In [ ]:

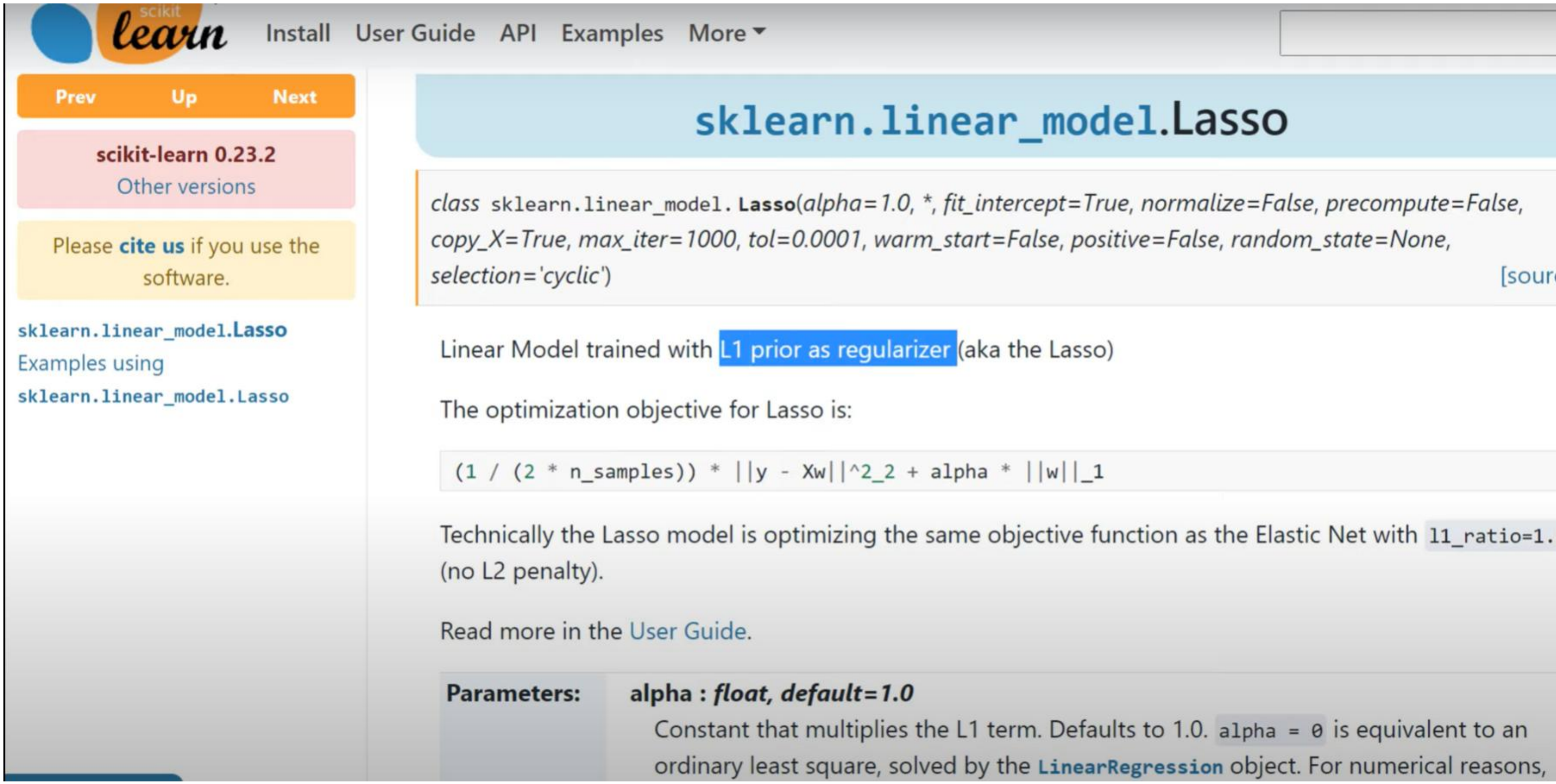
reg: Regular Linear Regression



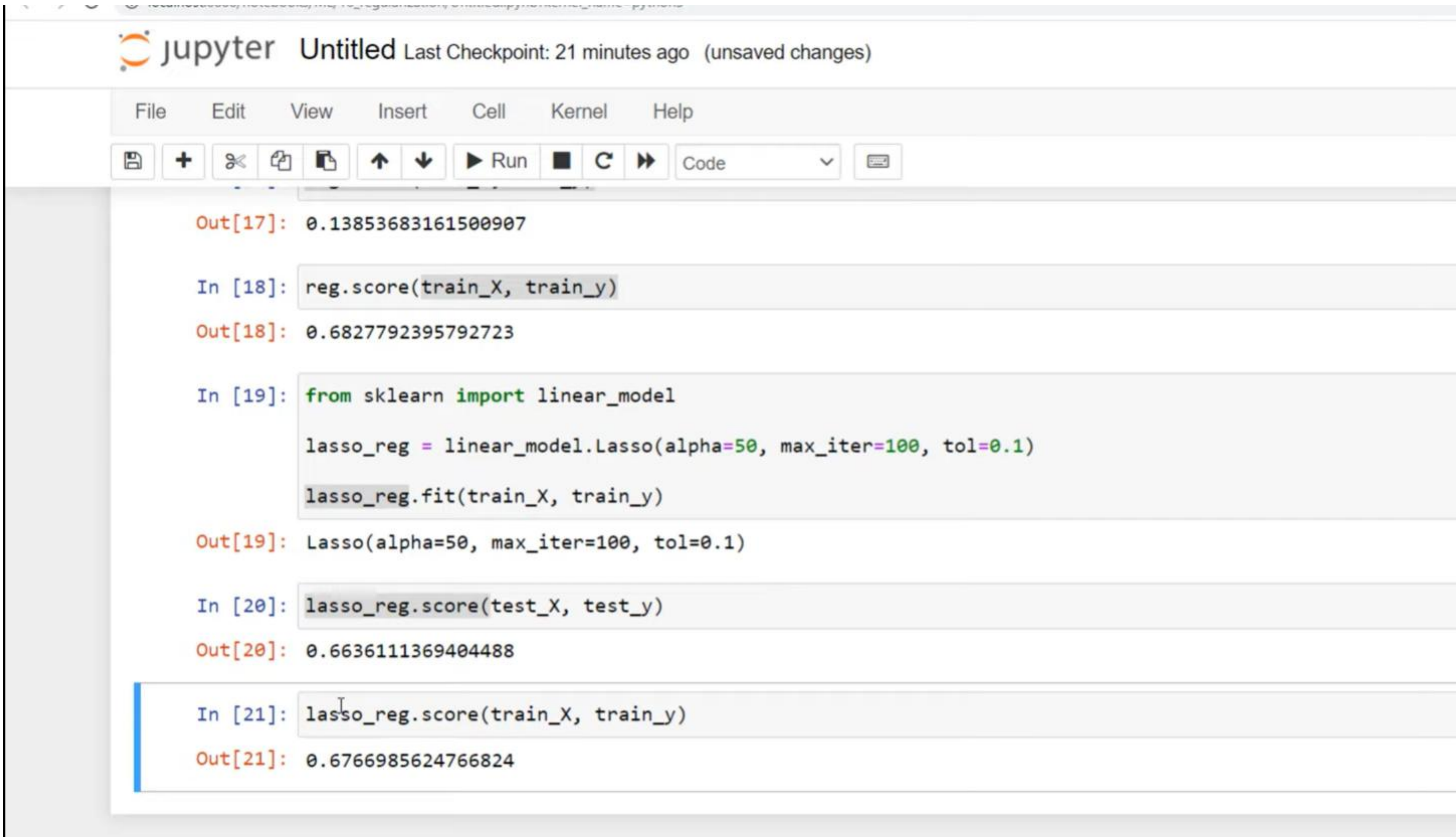
Overfitting case: train accu=0.68, test accu= 0.13,



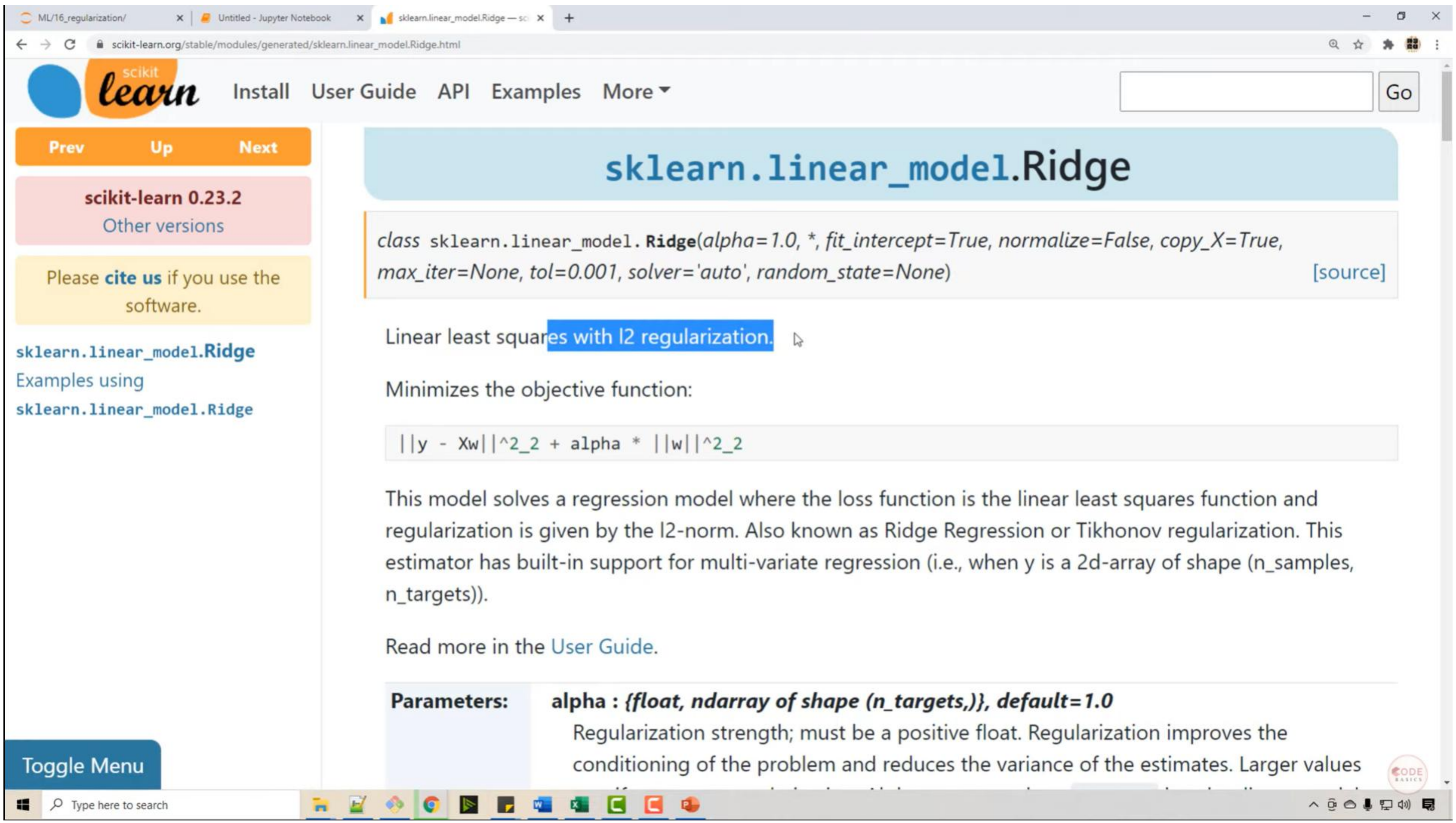
Lasso: L1 regularization



Result in Lasso: Good with 66%,67% accuracy



Ridge Regression: L2 Regularization



Result in Ridge: Good with 66%,66% accuracy

```
In [20]: lasso_reg.score(test_X, test_y)

Out[20]: 0.6636111369404488

In [21]: lasso_reg.score(train_X, train_y)

Out[21]: 0.6766985624766824

In [22]: from sklearn.linear_model import Ridge

         ridge_reg= Ridge(alpha=50, max_iter=100, tol=0.1)
         ridge_reg.fit(train_X, train_y)

Out[22]: Ridge(alpha=50, max_iter=100, tol=0.1)

In [23]: ridge_reg.score(test_X, test_y)

Out[23]: 0.6670848945194959

In [24]: ridge_reg.score(train_X, train_y)

Out[24]: 0.6622376739684328
```

Recent Reference Papers

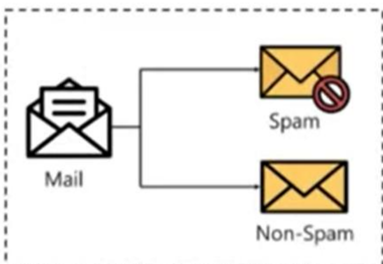
- Looking for interesting machine learning papers to read for the break or the new year? Here is a nicely curated list by Louis-François Bouchard.

- https://github.com/louisfb01/Best_AI_paper_2020

-

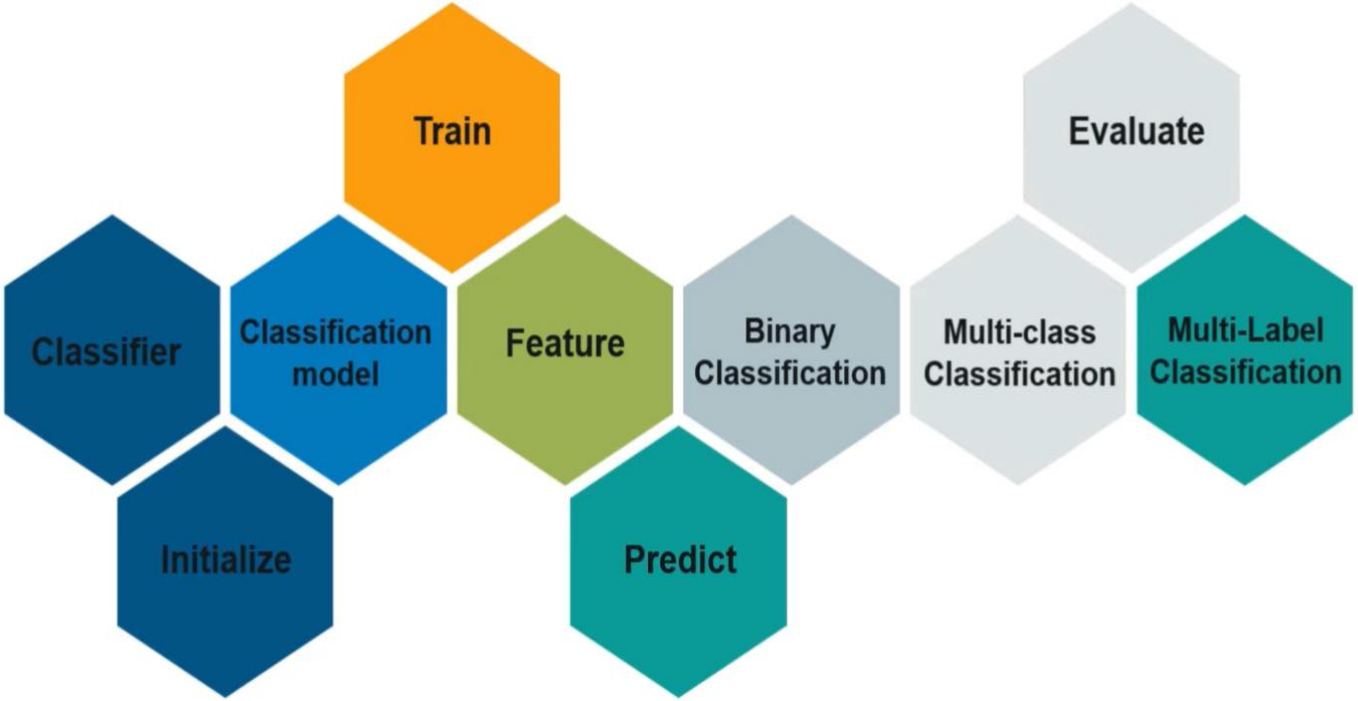
Thank You

What is Classification In Machine Learning?



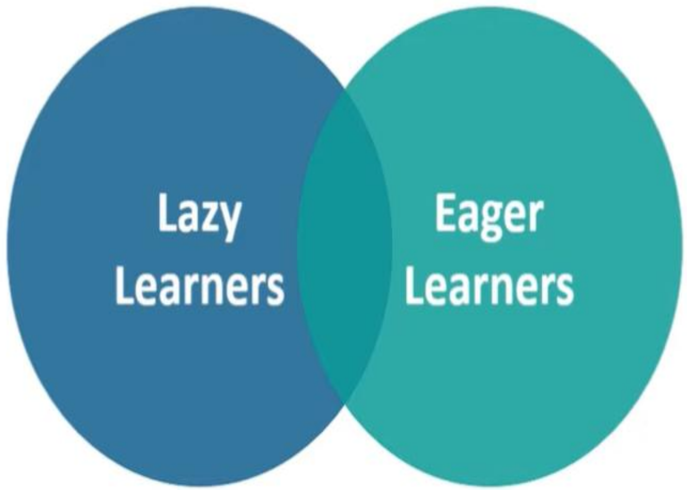
Classification is a process of categorizing a given set of data into classes, It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points. The classes are often referred to as target, label or categories.

Classification Terminologies



Types Of Learners In Classification

Lazy learners simply store the training data and wait until a testing data appears.



Eager learners construct a classification model based on the given training data before getting data for predictions