upGrad

#LifeKoKaroLift

PG Diploma in ML

upGrad

Course : PG Diploma in ML

Lecture On : Live Session on Regularisation in Logistic Regression

Instructor : Dr. Reena Duggal

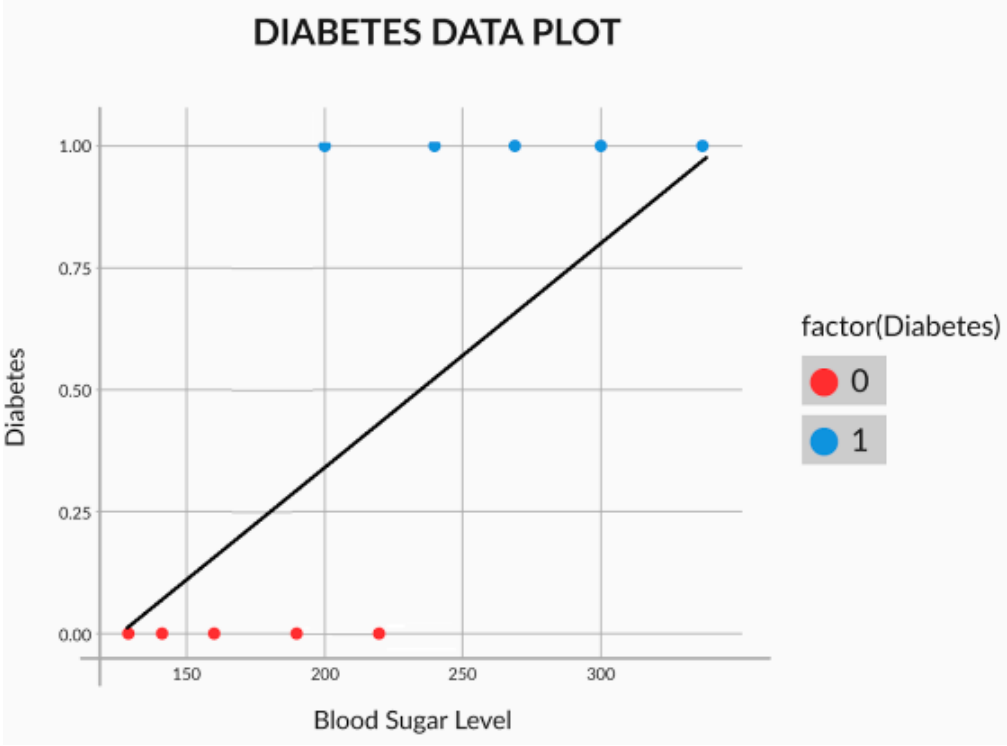
upGrad

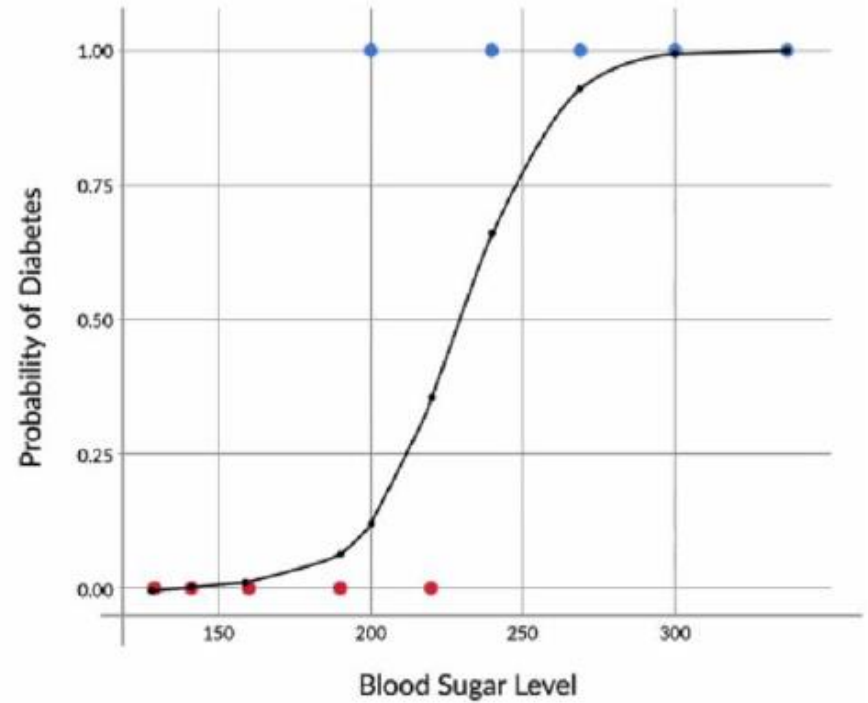
Today's Agenda

1. Logistic Regression quick recap
2. Regularisation in Logistic Regression
3. Lasso based Regularisation
4. Ridge based Regularisation
5. Hands-on in Python
6. Doubt resolution and discussion

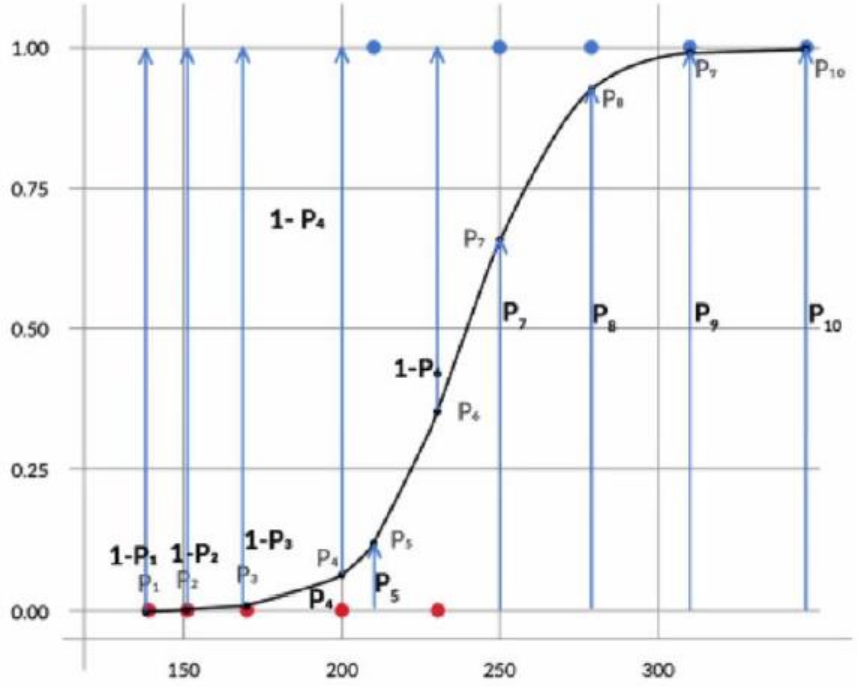
upGrad

Logistic Regression is a supervised classification model. It allows you to make predictions from labelled data, if the target (output) variable is categorical.



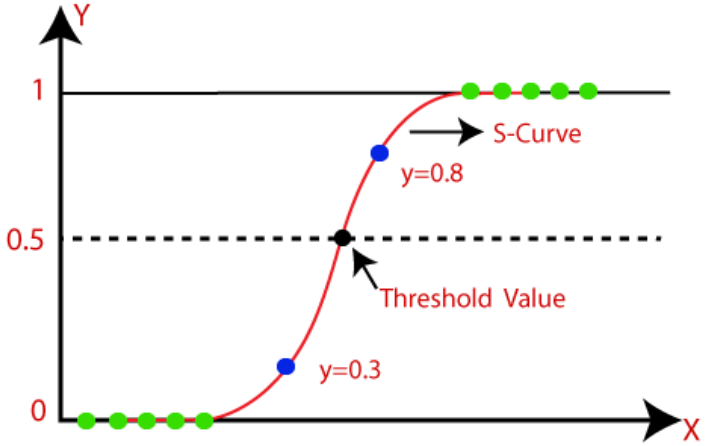


upGrad



$$P(\text{Diabetes}) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \qquad \ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x$$

$$P = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\beta_3 x_3+\cdots)}}$$



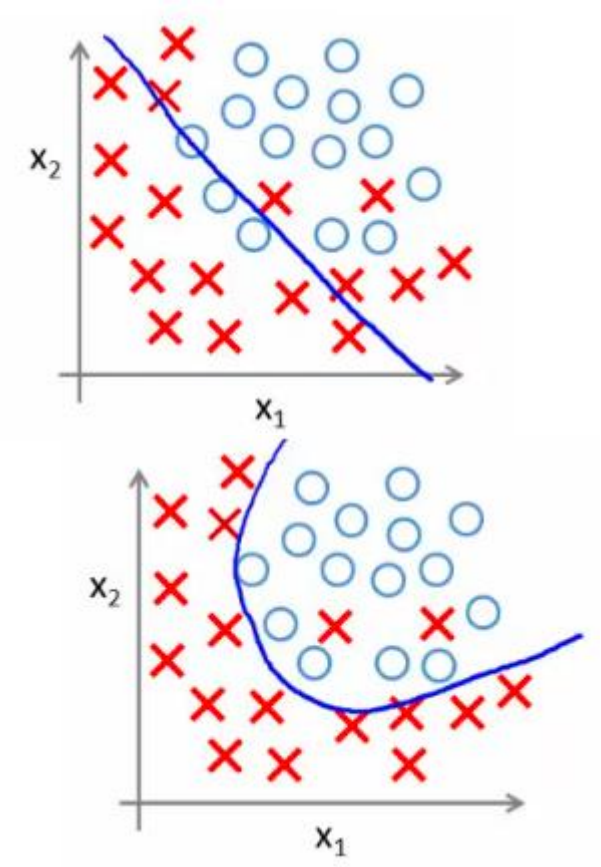
Likelihood = $(1-P_1)(1-P_2)(1-P_3)(1-P_4)(P_5)(1-P_6)(P_7)(P_8)(P_9)(P_{10})$

Generally, it is the product of -

[$(1-P_i)(1-P_i)$ ------ for all non-diabetics --------] X [$(P_i)(P_i)$ -------- for all diabetics -------]

Solve Non-linearity

upGrad



$$\ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots$$

You can define the degree of polynomial that will build the decision boundary.

Need to choose Coefficients which maximize Log-Likelihood(objective function)
or minimize the negative of Log-Likelihood

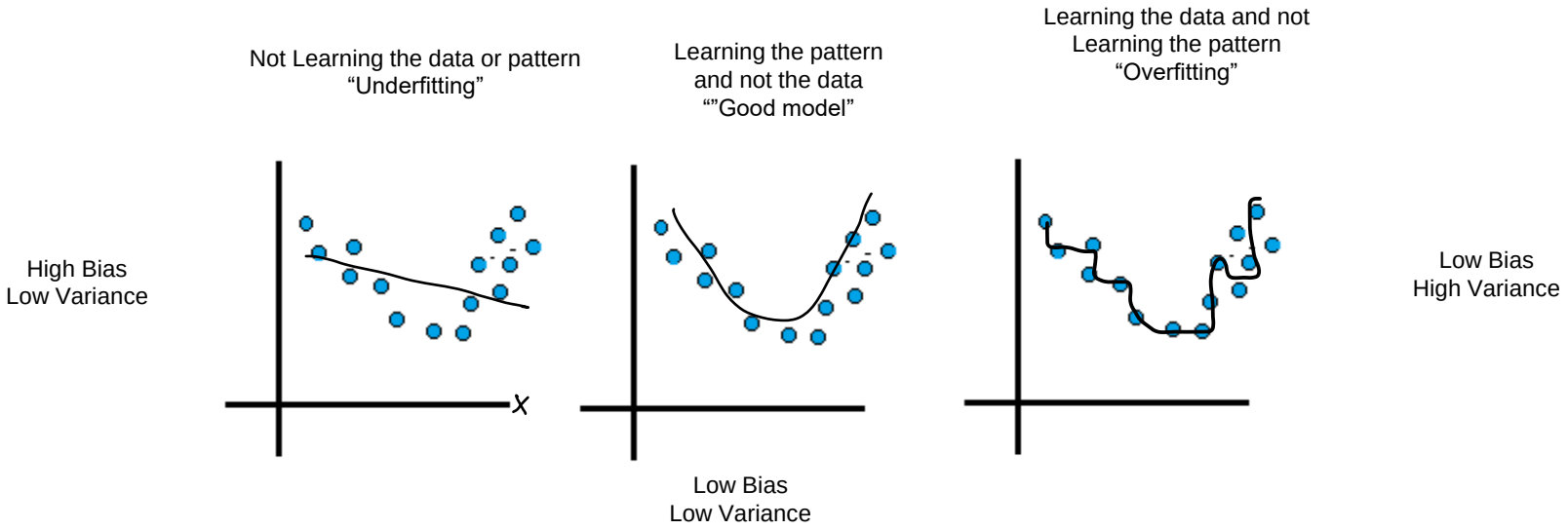
$$\hat{\boldsymbol{\beta}} = \min_{\beta} \; -LL(\boldsymbol{\beta}; \boldsymbol{y}, \boldsymbol{X})$$

The default nature of a model is that it does not account for model complexity - it only tries to maximize Log-Likelihood or reduce cost function, although if it may result in arbitrarily complex coefficients which result in **overfitting**.

Overfitting and Underfitting

upGrad

What is Bias and Variance

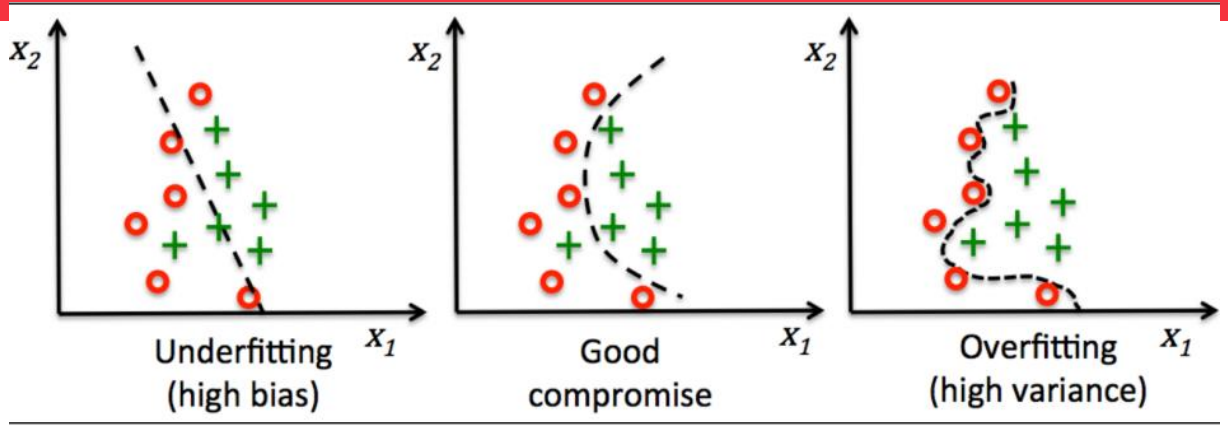


Bias - Amount of Error that the model is making on train data

Variance - Amount that the model(target variable) will change given different training data

Problem of overfitting

upGrad



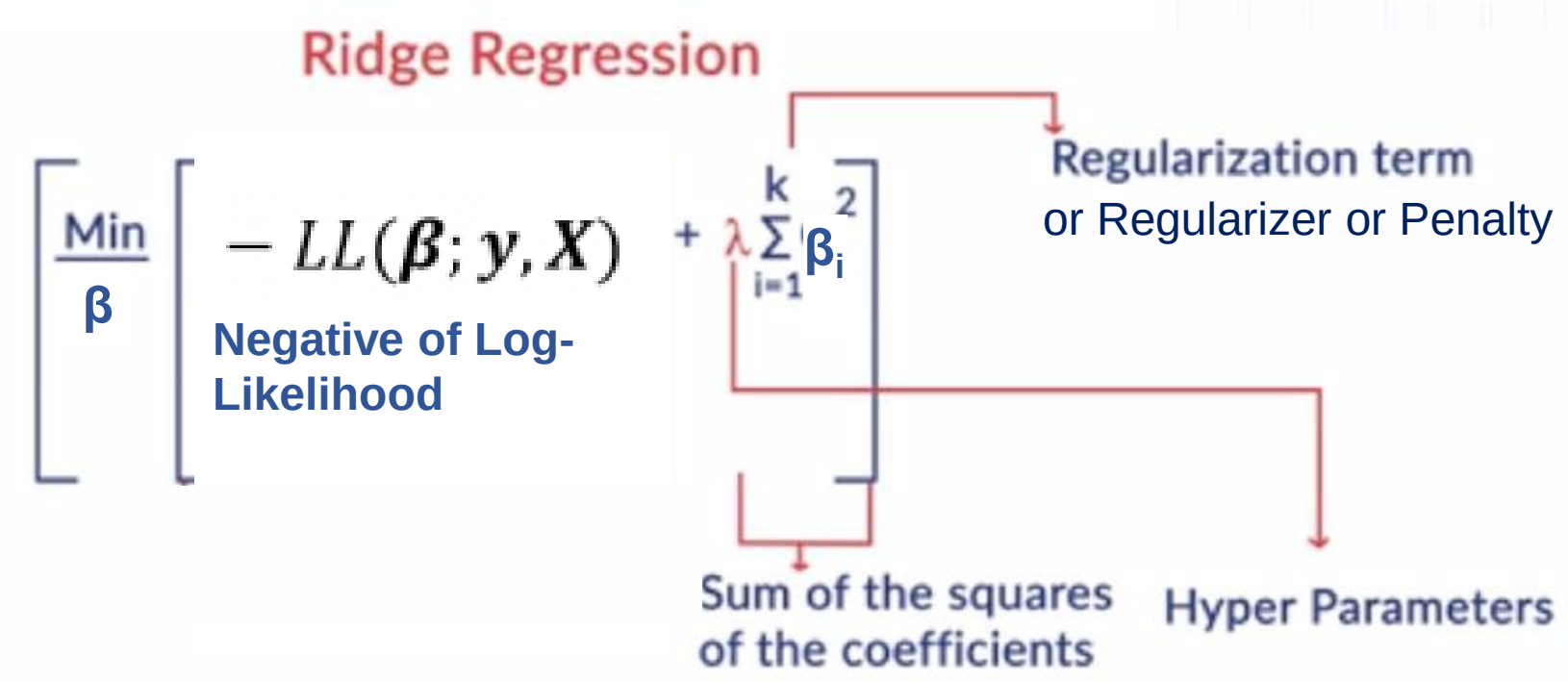
- **Regularization** can be used to avoid overfitting by creating an optimally complex model
- Regularization adds a penalty on the different parameters of the model to reduce the freedom of the model.
- Hence, the model will be less likely to fit the noise of the training data and will improve the generalization abilities of the model
- In regularized logistic egression, the objective function has two parts –
  the **Cost function(-ve of Log-likelihood)** and the **regularization term**.

$$\hat{\boldsymbol{\beta}} = \min_{\beta} \; -LL(\boldsymbol{\beta}; \boldsymbol{y}, \boldsymbol{X}) + \lambda R(\boldsymbol{\beta})$$

Ridge Regression (L2 regularization)

upGrad

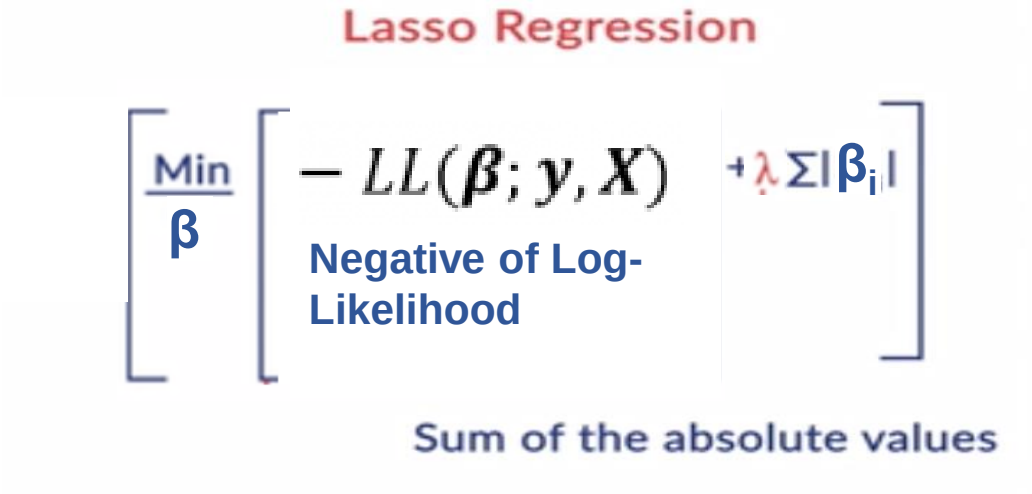
In ridge regression, an additional term of **"sum of the squares of the coefficients"** is added to the cost function along with the error term



Lasso Regression (L1 regularization)

upGrad

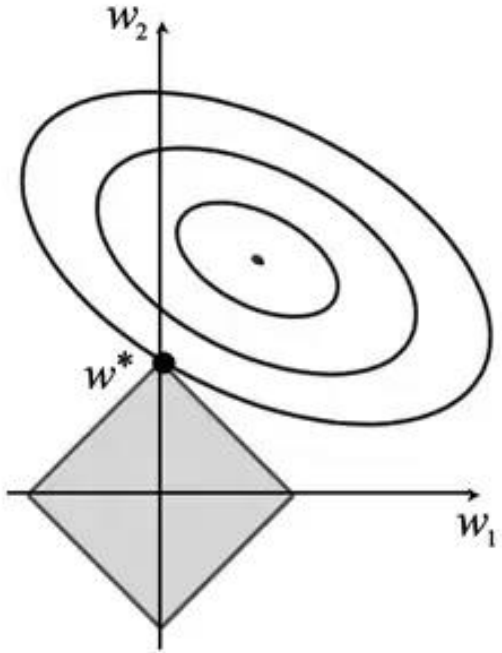
In case of lasso regression, a regularisation term of **"sum of the absolute value of the coefficients"** is added.



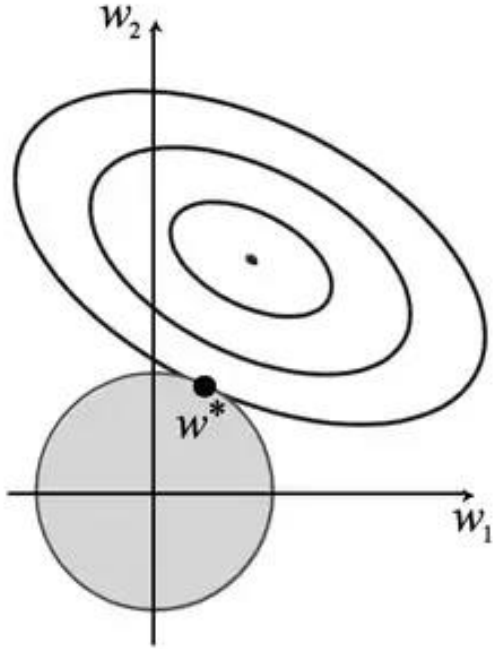
- Lasso regression not only helps in reducing over-fitting but it can help us in feature selection

How coefficients become zero for L1? (Contour Mapping)

upGrad



L1



L2

Let’s understand Lambda

upGrad

Cost function = minimize(-ve of Log-Likelihood + λ * Complexity)

λ (also called the regularization rate) is the tuning parameter that decides how much we want to penalize the flexibility of our model

I. λ = 0, no regularisation

II. λ is high, more regularisation. Model will be simple, but you run the risk of underfitting your data. Your model won't learn enough about the training data to make useful predictions.

III. λ is low, less regularisation. Model will be more complex, and you run the risk of overfitting your data. Your model will learn too much about the particularities of the training data, and won't be able to generalize to new data.

How to choose λ?

You’ll play around with different values.

Python sklearn

upGrad

- Other common names for λ:
  - *alpha* in sklearn.ridge() and sklearn.lasso()
  - C in many algorithms e.g. sklearn.LogisticRegression()
    - Usually C actually refers to the inverse regularization strength, $\frac{1}{\lambda}$
    - Interpretation:
      - C is high, less regularisation. Model is complex.
      - C is low, high regularisation. Model is simple.

- In sklearn.LogisticRegression(), L2 regularization(Ridge) is applied by default

Python Case Study

upGrad

#LifeKoKaroLift

Thank You!