Linear Discriminant Analysis

1.Theory

From Bayes Rules, the condition probability can be formulated as:

$$\mathrm{p(Y = k|X = x)} = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

Where $f_k(x)$ is the conditional density function of X|Y=k, and $\pi_k = P(Y = k$ is the prior probability.

The best prediction is picking the one that maximizing the posterior:

$$\arg\max_k \ \pi_k f_k(x)$$

LDA model $f_k(x)$ as a normal distribution. Suppose we model each class density as multivariate Gaussian N(u_k , $\sum_k$), and assume that the covariance matrices are the same across all k, i.e., $\sum_k=\sum$

Then, the probability function for class k is:

$$f_k(x) = \frac{1}{(2\pi)^{\frac{p}{2}} \sum^{\frac{1}{2}}} \exp[-\frac{1}{2}(x - u_k)^T \sum^{-1}(x - u_k)]$$

The log-likelihood function for the conditional distribution is:

$$log f_k(x) = -\log((2\pi)^{\frac{p}{2}} \sum^{\frac{1}{2}}) - \frac{1}{2}(x - u_k)^T \sum^{-1}(x - u_k)]$$

$$= -\tfrac{1}{2}(x - u_k)^T \sum^{-1}(x - u_k) + constant$$

Hence, we just need to select the category that attains the highest posterior density:

$$Y_{pred} = \arg\max_k log(\pi_k f_k(x))$$

$$= \arg\max_k -\tfrac{1}{2}(x - u_k)^T \sum^{-1}(x - u_k) + log(\pi_k)$$

Noticing that quadratic term can be simplified to:

$$-\frac{1}{2}(x - u_k)^T \sum^{-1}(x - u_k)$$

$$= \mathrm{x}^T \sum^{-1} u_k - \tfrac{1}{2} {u_k}^T \sum^{-1} u_k + irrelevant\ things$$

Then the discriminant function is defined as:

$$\delta_k(x) = x^T \Sigma^{-1} u_k - \frac{1}{2} u_k{}^T \Sigma^{-1} u_k + log(\pi_k)$$

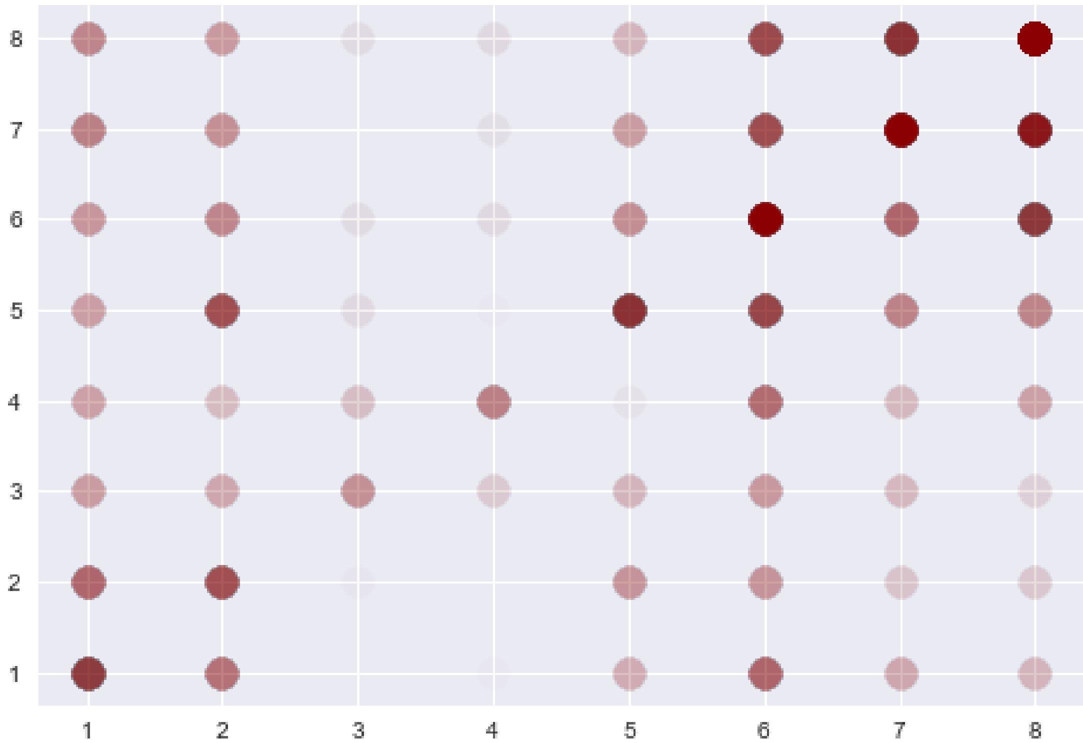
$$= w_k^T x + b_k$$

We can just calculate w_k and b_k for each class k.

2.Result

In LDA.py, I did the preprocessing and wrote the function for LDA. Then I implemented LDA (select all the attributes), it only cost 0.16 seconds. I got the prediction in variable “pred_kaggle” and I got score 0.51 on Kaggle.

Here is accuracy plot (by splitting train data into train and test):



Plot: Actual(X) vs Predict(Y)