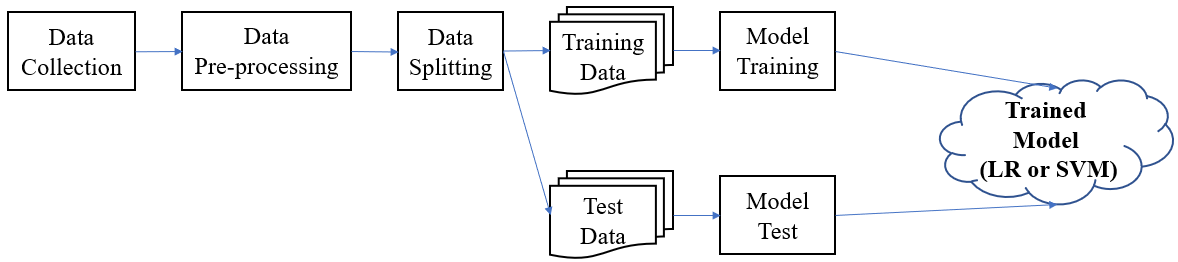
Census Income Prediction (CIP)

Huijun Hong

CS 767

## **1. SUMMARY**

My overall purpose of this project, i.e., CIP, is to predict whether a person earns more than 50,000 dollars a year based on the census information.



The workflow of CIP is presented above, which follows traditional machine learning designs, i.e., collect data first, and pre-process the data (including eliminating noise, filling the missing values and normalization et al.), and then split the data into training and test part. The training set is for training the target model. I choose logistic regression and support vector machine for my models. Once the model finish training, I can then employ the model to predict other unsees samples. Before that, test the performance of the trained model with the previous split test set is a common strategy and good practice.

## **2. Dataset**

UCI Adult, also known as “Census income” Dataset, which is used to predict whether income exceeds $50K/year based on census data. The dataset contains 32,561 samples, each consists of 14 attributes. My dataset is downloaded from UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/Adult).

**I/O Examples:**

Example 1: Input 28 years old, doctor’s degree and work in a high-tech company … Output YES with confidence 95%.

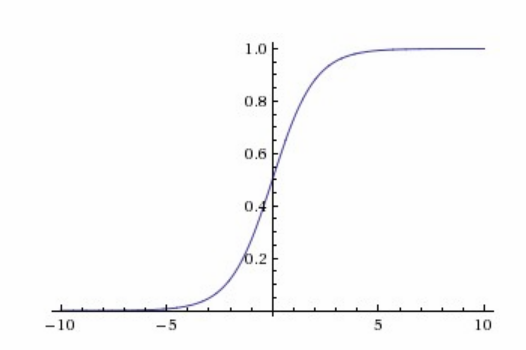
Example 2: Input 18 years old, high-school student and not employed … Output NO with confidence 90%.

## **3. Approaches**

I use logistic regression (LR) [1] and support vector machine (SVM) [2].

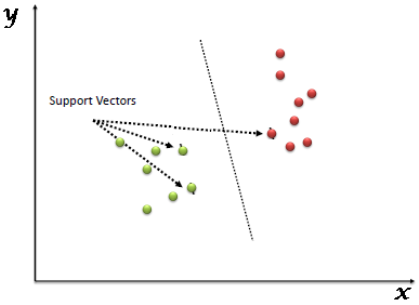
Logistic regression is one of the most popular, widely used, easy to understand and implement machine learning algorithms, especially when confronting with binary classification tasks. The basic formular of LR is a sigmoid function and is presented below:

Where and are model parameters that need to be solved via training and is the feature vector, i.e., the census data in my project. The shape of a sigmoid curve is shown below:



Notably, LR outputs real values between 0 and 1. The larger the output value, the more likely that the associate sample belong the positive class. Normally, if LR predicts a value larger than 0.5, we will think that the associate sample is in the positive class.

Support vector machine is another widely used and powerful machine learning algorithm in solving binary classification tasks. Imagine that data samples are essentially points in n-dimensional space, where n is the number of features. SVM tends to find the hyper-plane that can perfectly differentiate points that are in different classes. In other words, SVM can divide the points of the same class to the same side of the hyper-plane, as the figure below shows. SVM is more difficult to understand, but in many cases, outperforms LR in binary classification tasks with a great number of features.

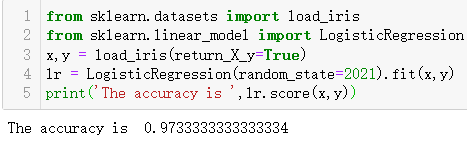


The logistical regression and support vector machine are two different machine learning algorithms. The LR is easy to implement and understand and is essentially a sigmoid function.

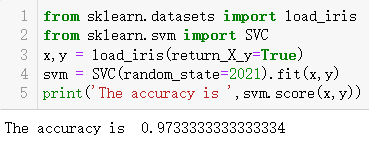
It takes a feature vector and transform it linearly with the linear equation , then it calculates a value (probability) on to perform the classification. Generally, larger than 0.5 will be considered as positive. SVM treats data samples as points in n-dimensional space, where n is the number of features. It tends to find the hyper-plane that can perfectly differentiate points that are in different classes. In other words, SVM can divide the points of the same class to the same side of the hyper-plane.

I choose LR and SVM for this task, because they are two of the most widely used algorithms in binary classification, and usually can achieve remarkable performances. I run some test examples of the two algorithms from the Scikit-learn, and the results are shown below.

LR code snippets and results:



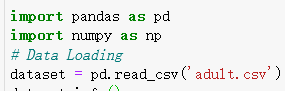
SVM code snippets and results:



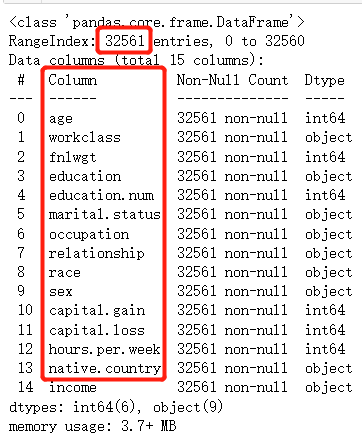
Surprisingly, the two algorithms achieve exactly the same classification accuracy on classifying the iris flower dataset.

## **4. Results**

## **4.1 Data Loading**



As the picture below show, the data set consists of 14 features, contains 32,561 data records in total.



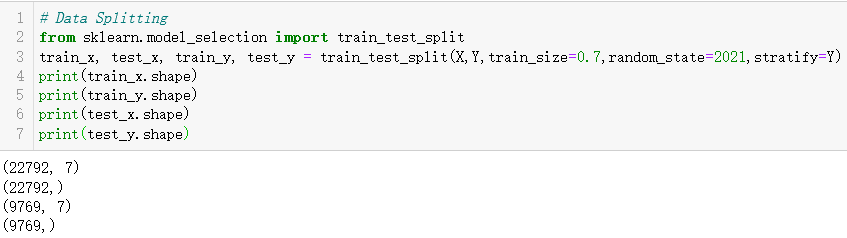
## **4.2 Data Pre-processing**

For leveraging algorithms implemented in the scikit-learn package, I only select features of numeric type and discard those string-type features. Then, I fill the missing values with default values and encode the binary features.



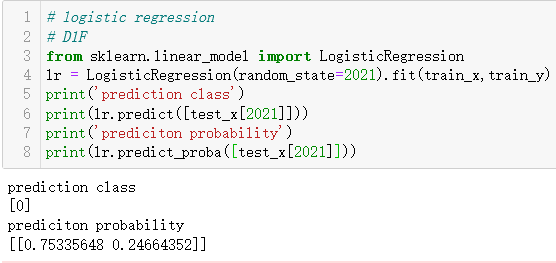
## **4.3 Data Splitting**

We split the processed data into training and test set with the ratio of 7:3, thus, get a training set with 22,792 data records and a test set with 9,769 data records.

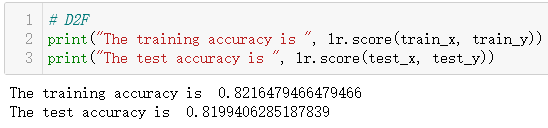


## **4.4 Model Training and Evaluation**

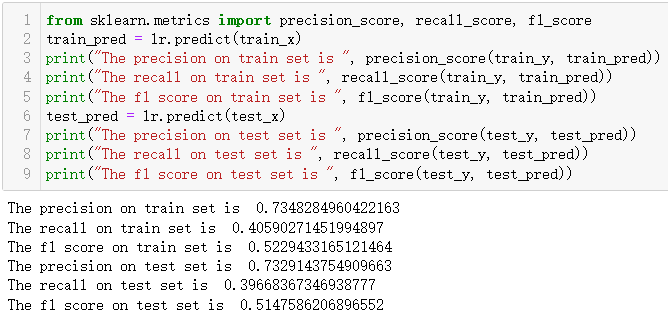
## **4.4.1 LR**



As presented, D1F is fulfilled. Given a data record, we can not only predict its class but also the associate probability.

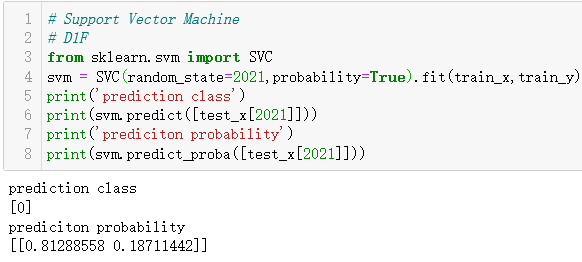


Sadly, I only get over 80% classification accuracy for both training and test set. This requirement is not accomplished with satisfaction.

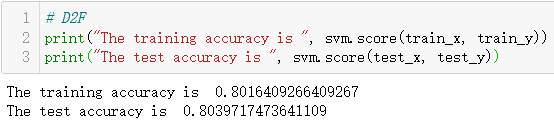


I learned to use various metrics to evaluate a machine learning model and found that classification accuracy is not enough for evaluation. For example, I achieve over 80% accuracy on the classification, but the recall and f1 score is astonishingly low.

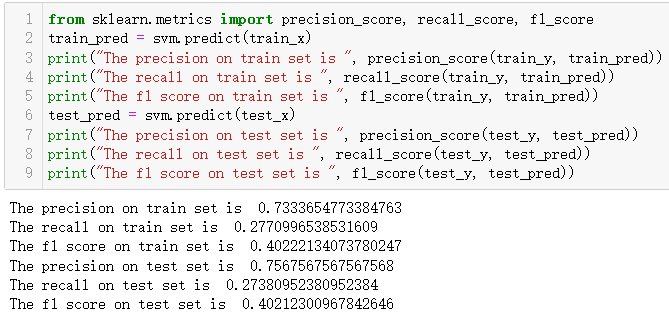
## **4.4.2 SVM**



As presented, D1F is fulfilled. Given a data record, we can not only predict its class but also the associate probability.



Sadly, I only get over 80% classification accuracy for both training and test set. This requirement is not accomplished with satisfaction.



Still, I learned to use various metrics to evaluate a machine learning model and found that classification accuracy is not enough for evaluation. For example, I achieve over 80% accuracy on the classification, but the recall and f1 score is astonishingly low.

## **5. Conclusion and Lesson Learned**

## **5.1 What did not work well**

The classification performances of logistic regression and support vector machine are both disappointing. I expect models with high classification accuracy (larger than 90%), but I only got over 80% accuracy. In reality, it certainly not good enough. I am new to machine learning, and always think machine learning is kind of a magic and magically have impressive performances. From this project, I realize that I was wrong. Machine learning can be quite powerful, but only when the overall procedure, such as data collection, pre-processing, model and hyper-parameter selection, is all accomplished appropriately. Those work can be really time consuming.

## **5.2 What did work well**

Through this project, the most significant aspects of my project that meet my desires are that I get familiar with the complete procedure of solving problems with machine learning techniques. I get to know that data processing or feature engineering is as important as, or even more important than model and hyper-parameter selection. I also realize that evaluation should be conducted with multiple metrics, a single or only a few metrics may give false impressions.