synchrony
FINANCIAL

Income Classification Using Support Vector Machines

OMAR CHOUDHRY

22ND FEBRUARY 2018

Project Overview

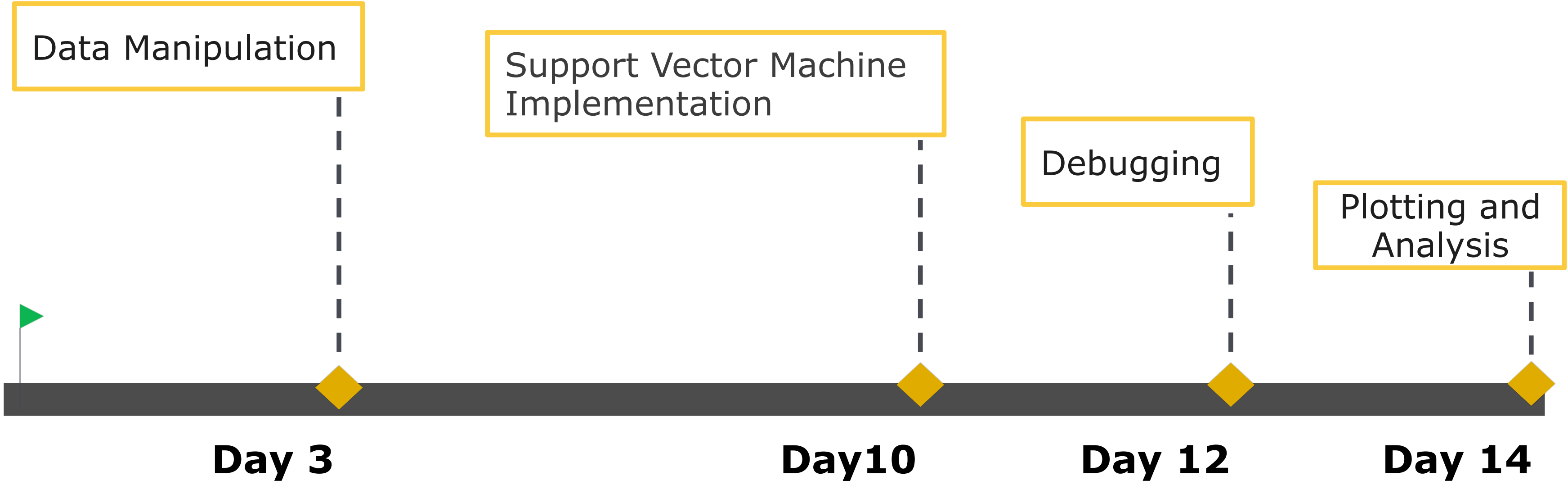
Project

Predict whether a household’s income is greater than or less than 50K based on some relevant attributes

Two-week group project for Applied Machine Learning

Support Vector Machine

Timeline



Data Manipulation

Support Vector Machine Implementation

Debugging

Plotting and Analysis

Day 3

Day10

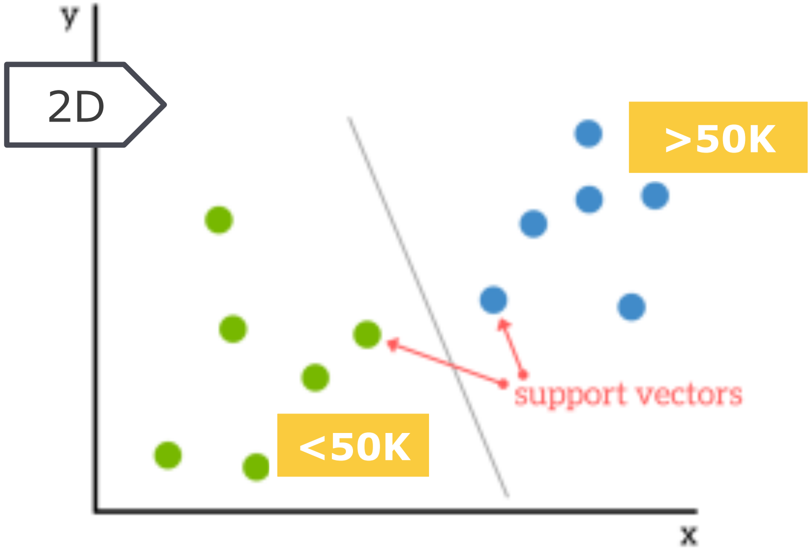
Day 12

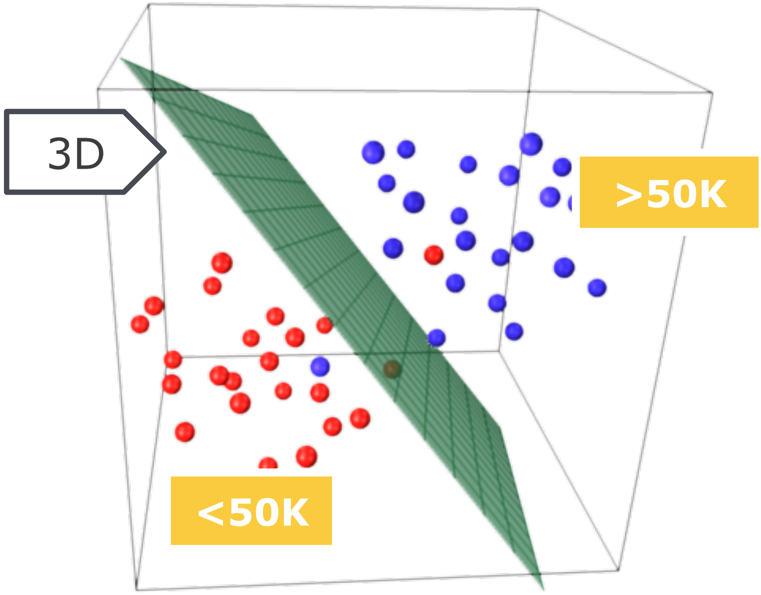
Day 14

Support Vector Machine

Supervised machine learning model that is used to classify a binary variable

SVM Examples



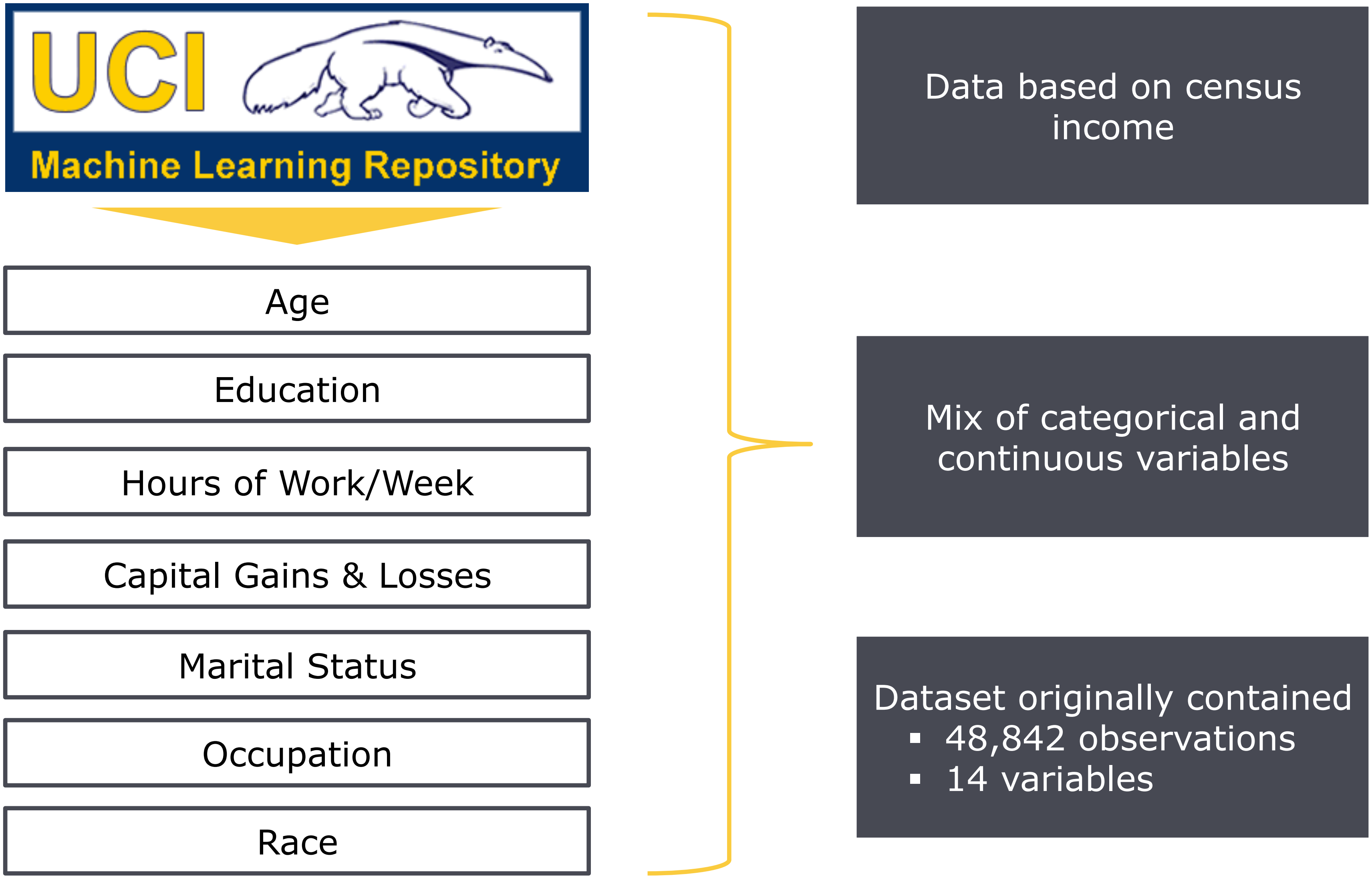


6 Predictive Attributes

Vectors in a 6D Space

Searching for the best hyperplane that separates them into the 2 groups based on what label they carry

Data Description



UCI Machine Learning Repository

- Age
- Education
- Hours of Work/Week
- Capital Gains & Losses
- Marital Status
- Occupation
- Race

Data based on census income

Mix of categorical and continuous variables

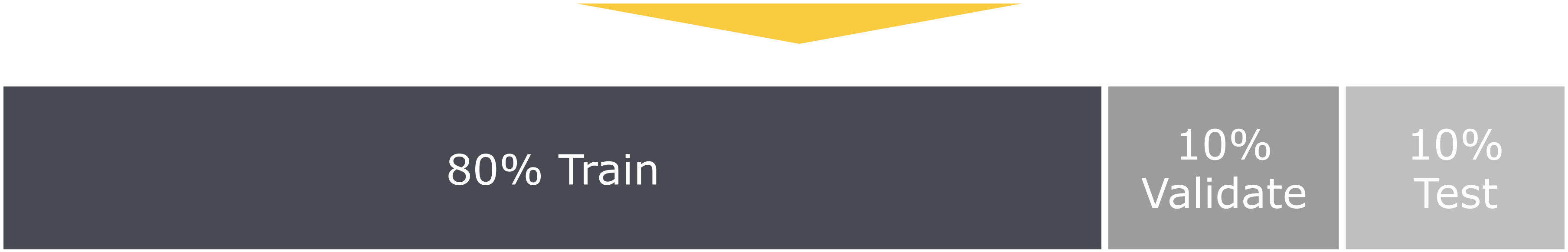
Dataset originally contained
- 48,842 observations
- 14 variables

Data Manipulation

Data Cleaning

1. Drop rows with missing data
2. Drop columns with categorical data
3. Scaled to unit variance and 0 mean

Separated the data set into training, validation, and testing sets


80% Train

10% Validate

10% Test

SVM Implementation

synchrony FINANCIAL

Initial Condition

a = weight vector → Set to 6 1’s

b = intercept (bias) → Set to 0

This represents our initial guess for the best hyperplane which is obviously not accurate

Passed these two parameters to a function that accepts a random row (x_k) and the corresponding label (y_k) from the training dataset

x 26,000

$$\mathbf{a}^{(n+1)} = \mathbf{a}^{(n)} - \eta \begin{cases} \lambda \mathbf{a} & \text{if } y_k\left(\mathbf{a}^T \mathbf{x}_k + b\right) \geq 1 \\ \lambda \mathbf{a} - y_k \mathbf{x} & \text{otherwise} \end{cases}$$

$$b^{(n+1)} = b^{(n)} - \eta \begin{cases} 0 & \text{if } y_k\left(\mathbf{a}^T \mathbf{x}_k + b\right) \geq 1 \\ -y_k & \text{otherwise} \end{cases} .$$

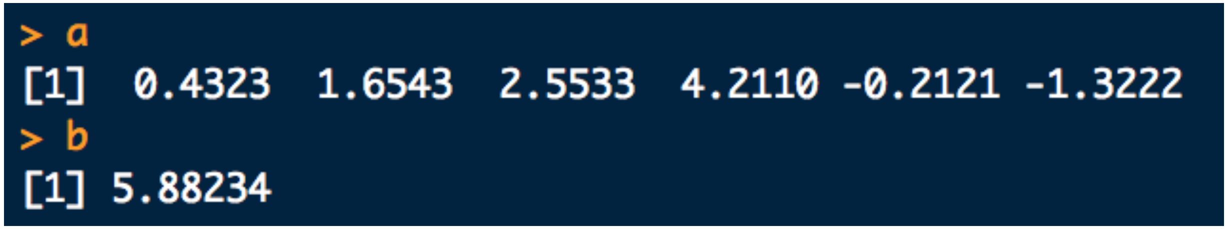
λ = regularization constant

η = learning rate

Returns best estimate for **a** and **b**

Results

Sample



```
> a
[1]  0.4323  1.6543  2.5533  4.2110 -0.2121 -1.3222
> b
[1] 5.88234
```

Formula

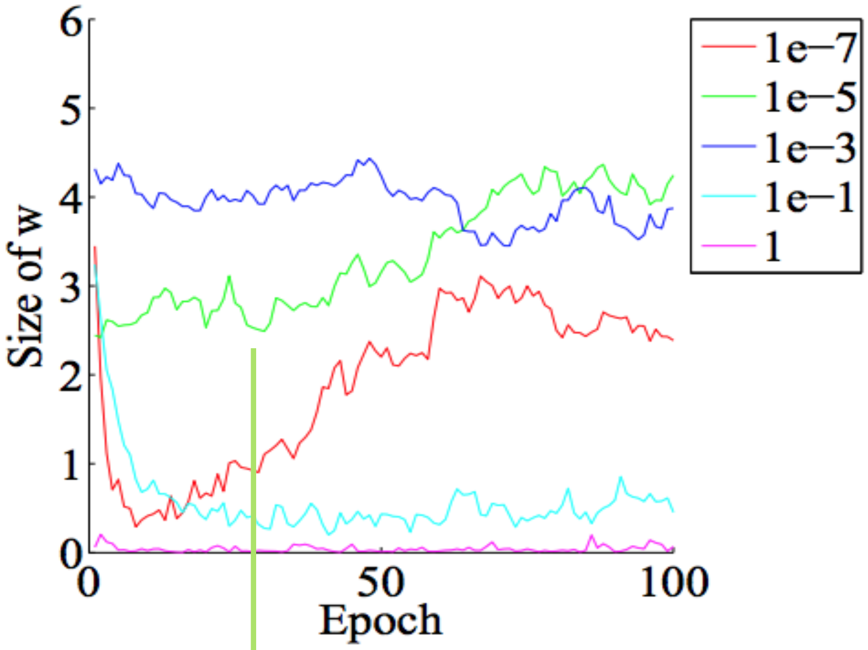
$$\text{sign}\left(\mathbf{a}^T\mathbf{x} + b\right) = \boxed{-1} \text{ OR } \boxed{1}$$

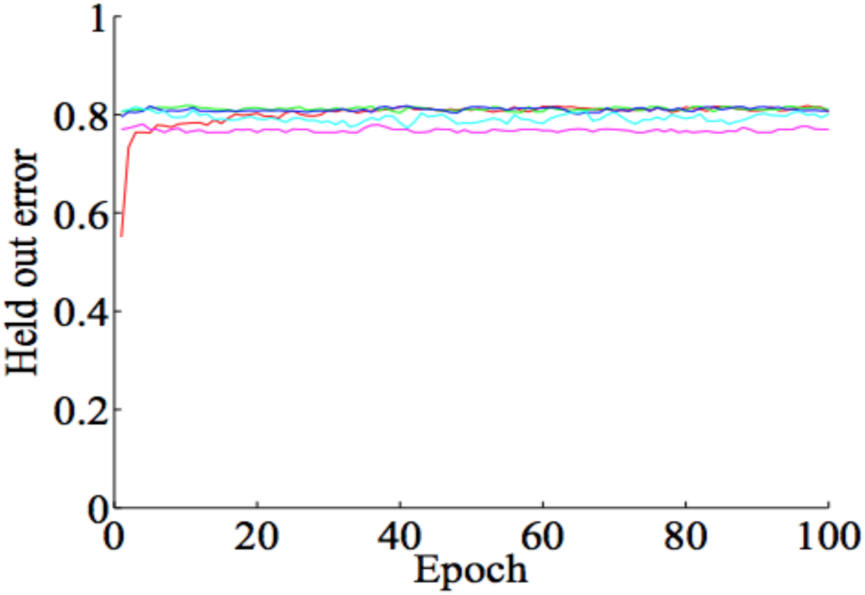
Row from Validation Set

The resulting sign indicates whether our model predicts above or below 50K

We then check the output against the actual value to assess accuracy

Results





Best Model

After 100 epochs using the regularization constant of 0.00001

This maximizes the magnitude of the weight vector with high accuracy

Conclusion

Project Applicability

Targeting customers based on household income

>50K

<50K

- Loan eligibility
- Financial advising
- Loyalty brands

Personal Development

Hard Skills

- Training a Machine Learning model
- Analysis and debugging

Soft Skills

- Working in a team
- Constrained time period

APPENDIX