Data Application Lab

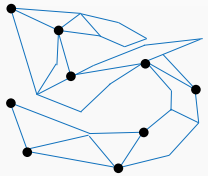
# Default Credit Card Default Detection

1 Introduction

2 Data Analysis

3 Proposed Solution

4 Result and Discussion

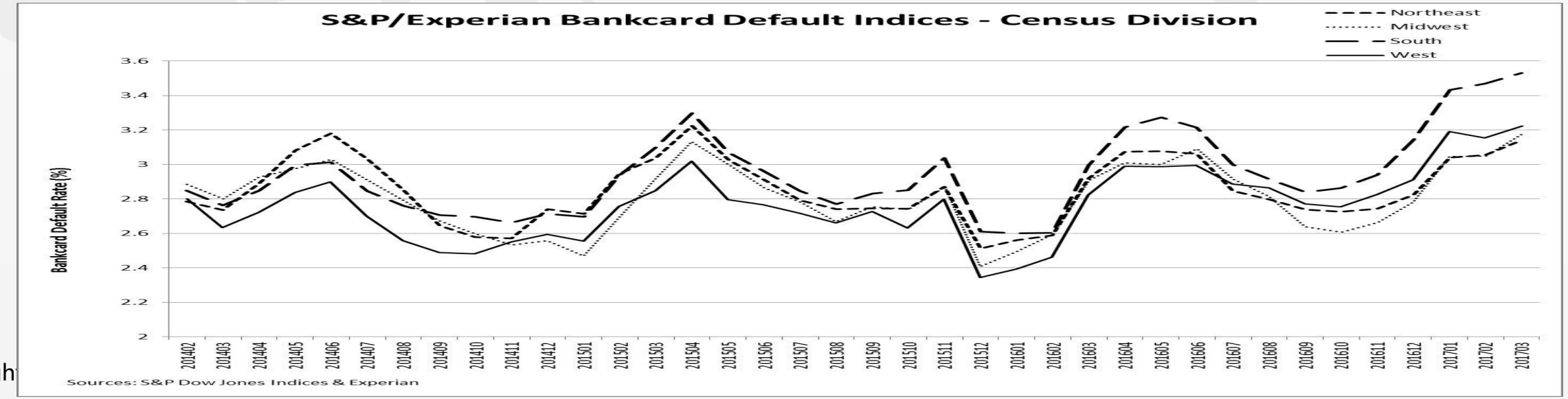


# Introduction

1

# Credit Card Default

- Default is to fail to make a payment on a debt by the due date.
- If someone miss the minimum card payment six months in a row, their credit card will be an default.
- If this happens with a credit card, creditors might raise interest rates to the default (or penalty rate) or decrease the line of credit.

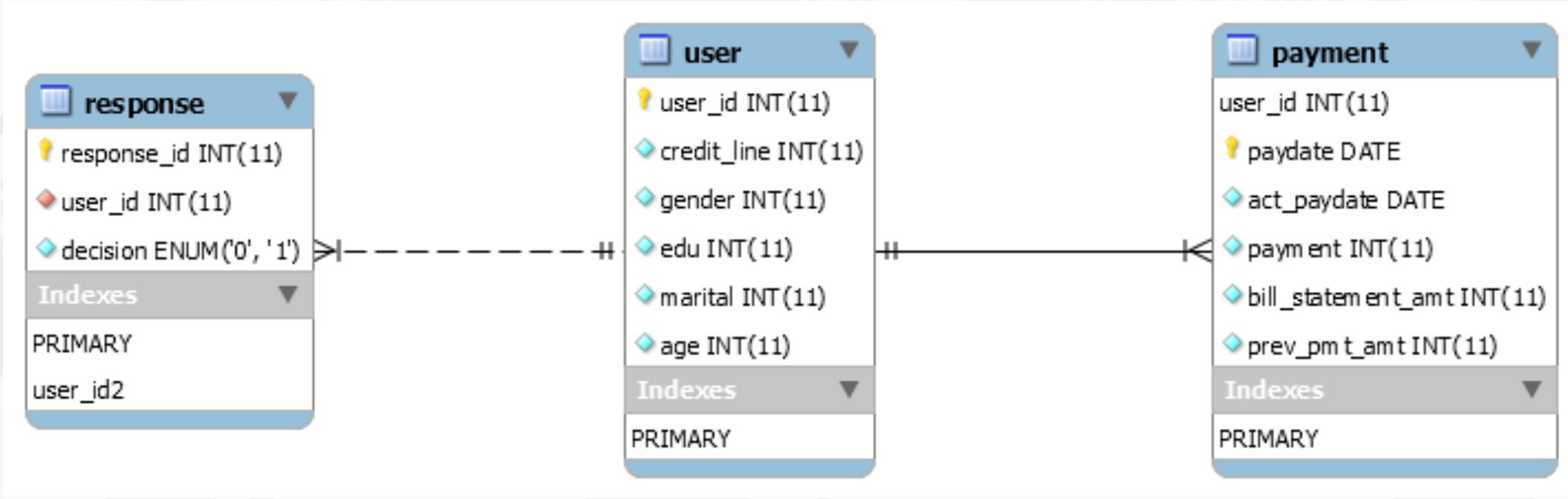


# Our Goal

Is to develop a system that detect client is most likely not able to pay in the next billing cycle.

# Data Analysis

# Original tables



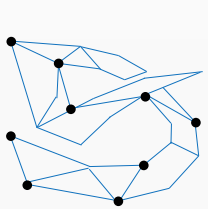
# Data Set Overview

| | id | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | ... | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 200000 | 2 | 3 | 1 | 53 | 0 | 0 | 0 | 0 | ... | 133207 | 136159 | 138741 | 6500 | 5000 | 5000 | 5100 | 5000 | 5400 | 0 |
| **1** | 2 | 130000 | 2 | 3 | 2 | 39 | 0 | 0 | 0 | 2 | ... | 130814 | 130758 | 123468 | 7500 | 10000 | 0 | 4500 | 4500 | 4179 | 0 |
| **2** | 3 | 350000 | 2 | 1 | 2 | 41 | 0 | 0 | 0 | 0 | ... | 119947 | 117328 | 118400 | 6000 | 5900 | 5800 | 4100 | 4500 | 5000 | 0 |
| **3** | 4 | 240000 | 2 | 2 | 1 | 43 | 1 | -2 | -2 | -1 | ... | 12700 | 12500 | 26225 | 0 | 0 | 12700 | 0 | 13725 | 0 | 0 |
| **4** | 5 | 180000 | 1 | 2 | 2 | 28 | -1 | -1 | -1 | -1 | ... | 332 | 416 | 416 | 0 | 416 | 332 | 500 | 3500 | 832 | 0 |

- Id = A unique Id field which represents a customer
- X1 = Credit line
- X2 = Gender (1 = male; 2 = female).

# Data Set Overview

- X3 = Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
- X4 = Marital status (1 = married; 2 = single; 3 = others).
- X5 = Age
- X6-X11 = Past Payments History (4/2005 to 9/2005)
- X12-X17 = Bill Statements (4/2005 to 9/2005)
- X18-X23: Amount of previous payment (4/2005 to 9/2005)
- Y: Response ( If Default = 1 else 0)




# Jupyter notebook

```
train_data = pd.read_csv("train.csv")
test_data = pd.read_csv("test.csv")
print (train_data.shape)
print (test_data.shape)
```

```
(25000, 25)
(5000, 24)
```

```
train_data.columns
```

```
Index([u'id', u'X1', u'X2', u'X3', u'X4', u'X5', u'X6', u'X7', u'X8', u'X9',
       u'X10', u'X11', u'X12', u'X13', u'X14', u'X15', u'X16', u'X17', u'X18',
       u'X19', u'X20', u'X21', u'X22', u'X23', u'Y'],
      dtype='object')
```

# Data Processing

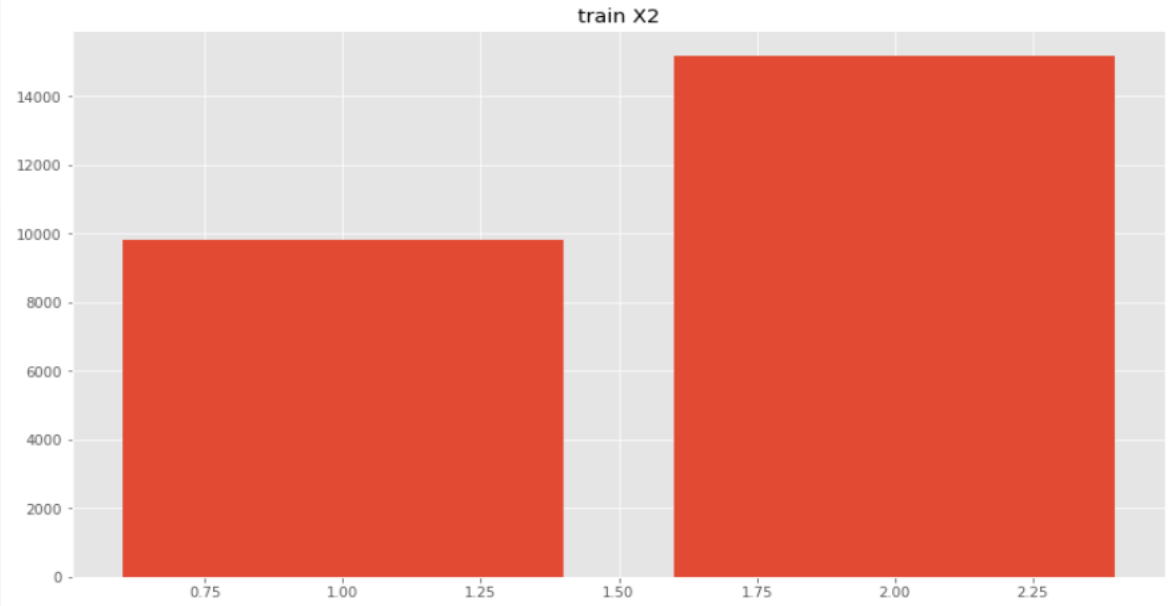
```
cat_v = []
con_v = []
for c in train_data.columns:
    if len(train_data[c].value_counts().index)<=15:
        cat_v.append(c)
    else:
        con_v.append(c)
cat_v.remove('Y')
target = ['Y']
```

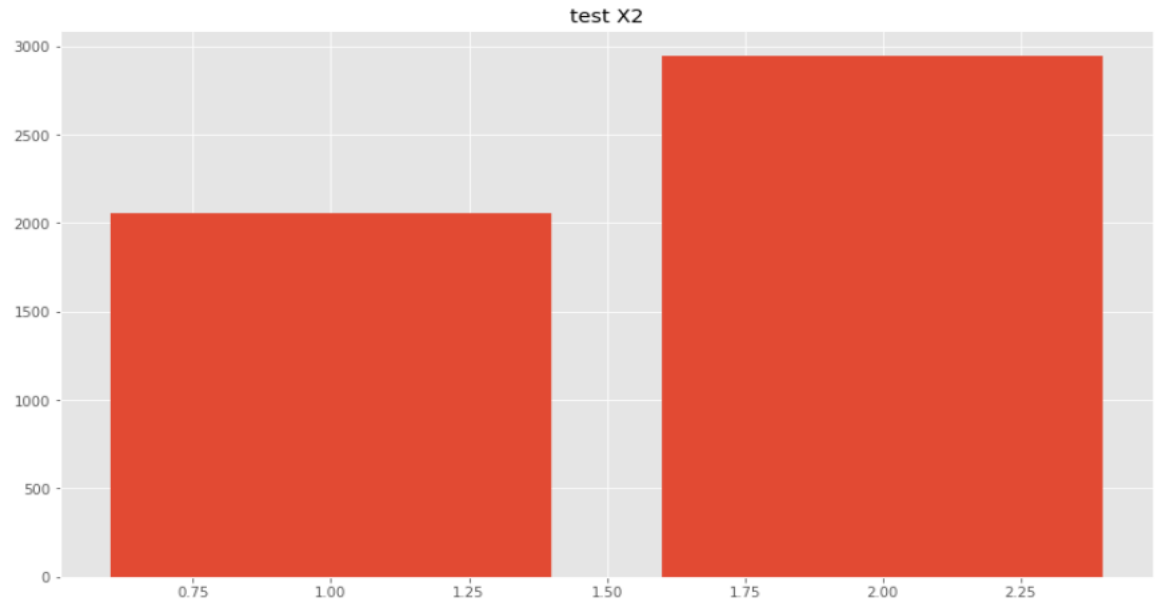
```
print("The continuous variables: ", con_v, "\n")
print("The categorical variables: ", cat_v)
```

```
('The continuous variables: ', ['id', 'X1', 'X5', 'X12', 'X13', 'X14', 'X15', 'X16', 'X17', 'X18', 'X19', 'X20', 'X21', 'X2
2', 'X23'], '\n')
('The categorical variables: ', ['X2', 'X3', 'X4', 'X6', 'X7', 'X8', 'X9', 'X10', 'X11'])
```

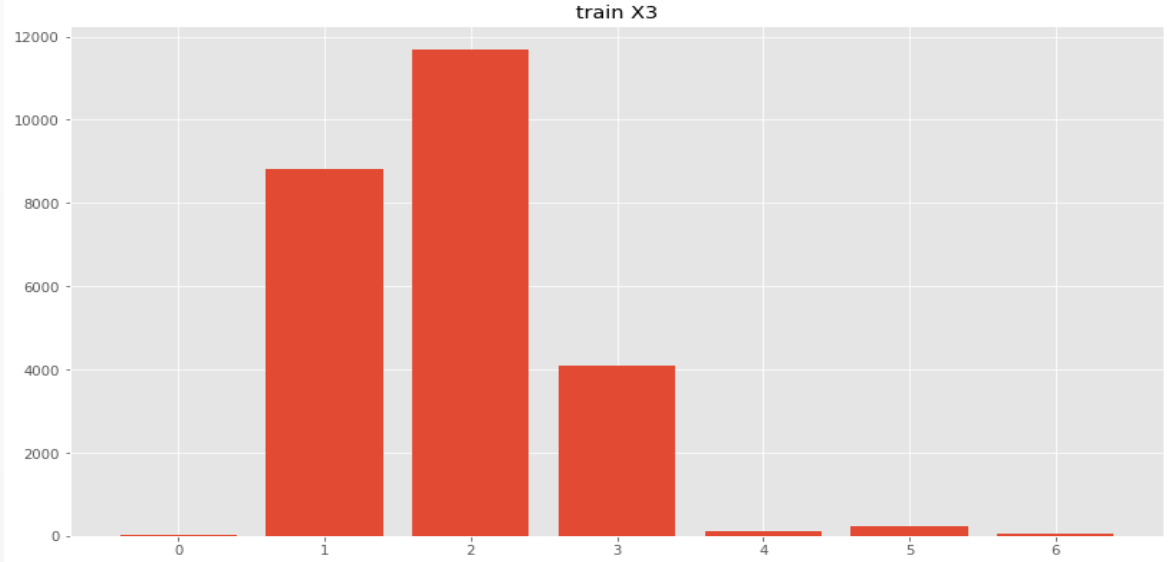
# Data Visualization

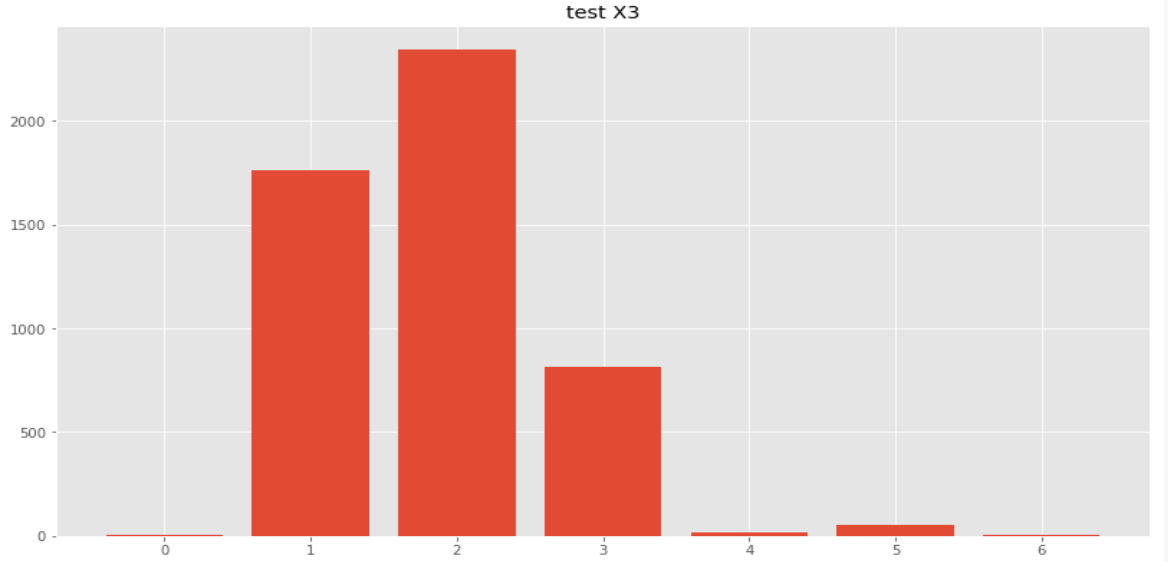
```
count=1
for i in range(len(cat_v)):
    fig = plt.figure(figsize=(30,80))
    plt.subplot(len(cat_v),2,count)
    plt.bar(train_data[cat_v[i]].value_counts().index, train_data[cat_v[i]].value_counts().values)
    plt.title("train "+cat_v[i])

    plt.subplot(len(cat_v),2,count+1)
    plt.bar(test_data[cat_v[i]].value_counts().index, test_data[cat_v[i]].value_counts().values)
    plt.title("test "+cat_v[i])
    count+=2
```

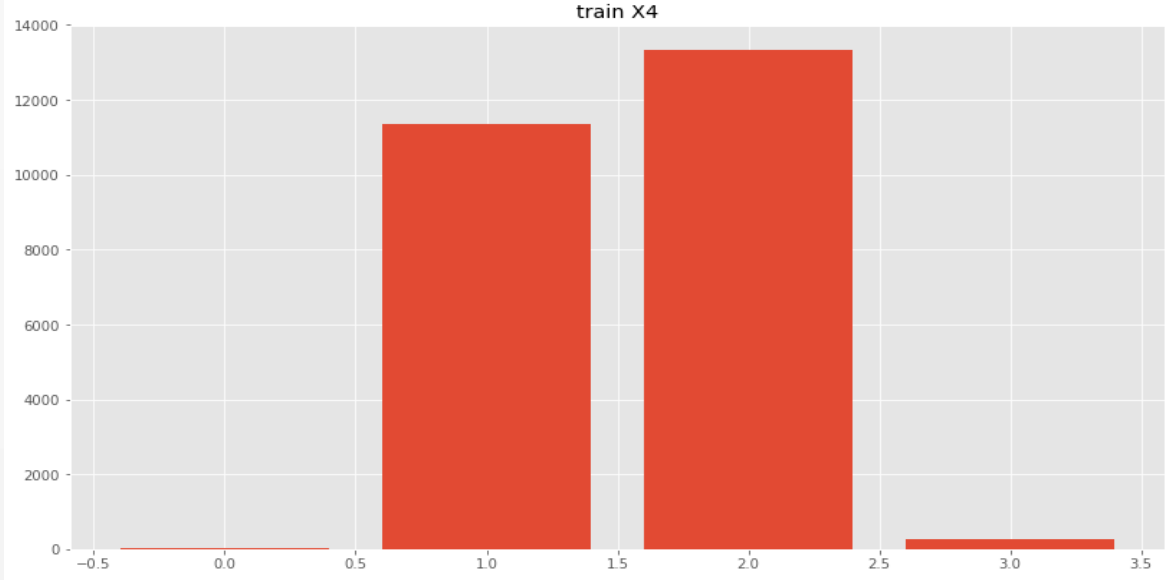


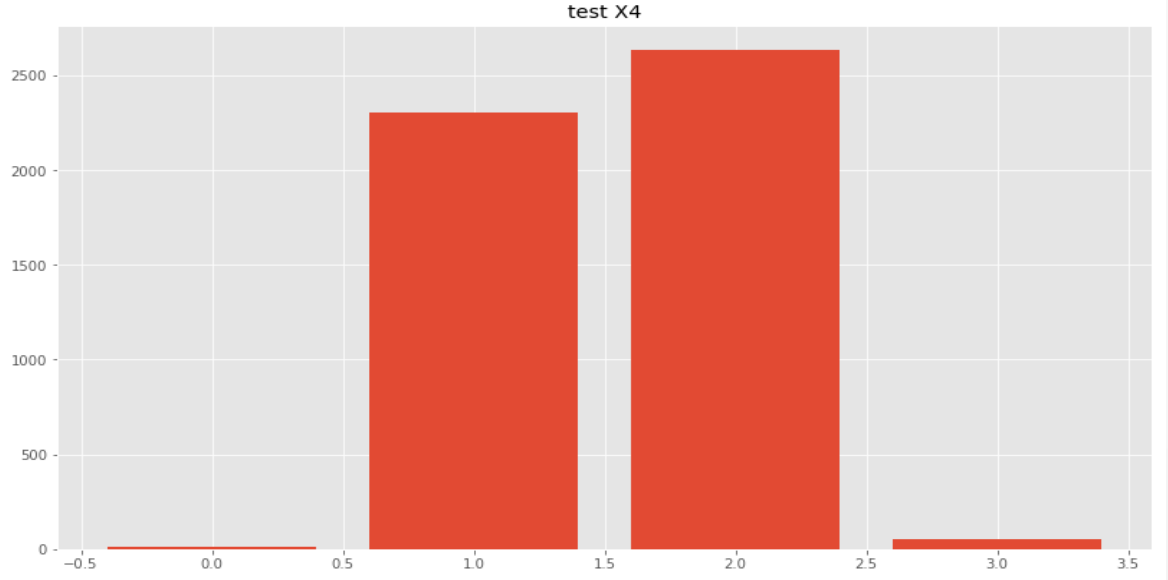



# Continue



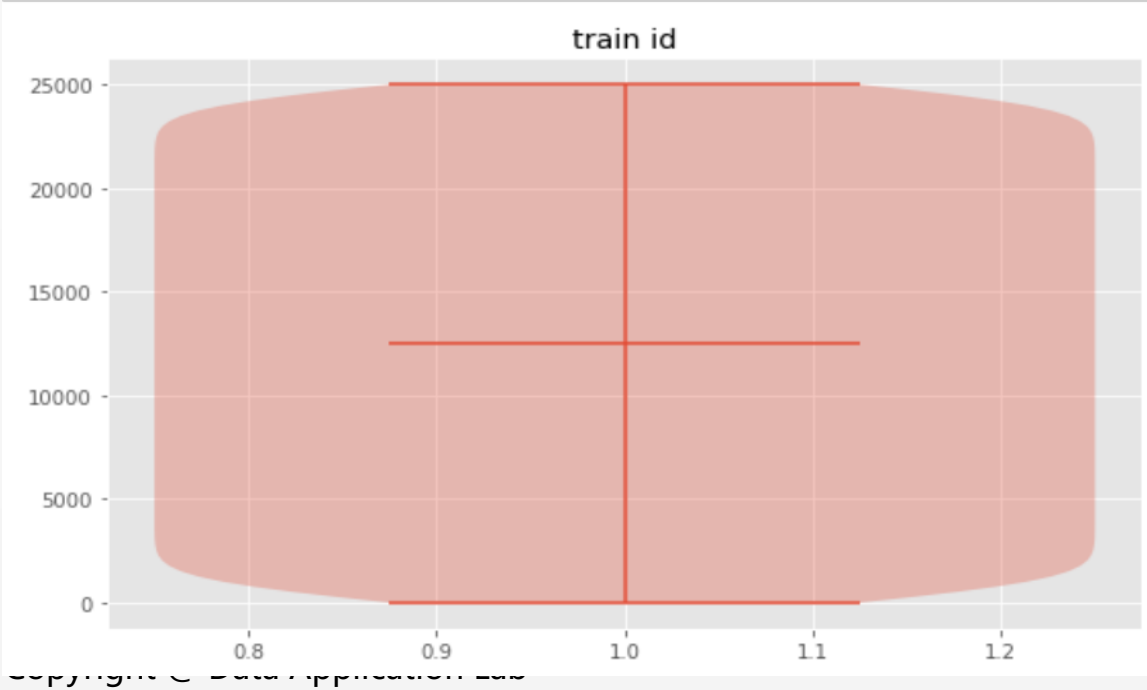



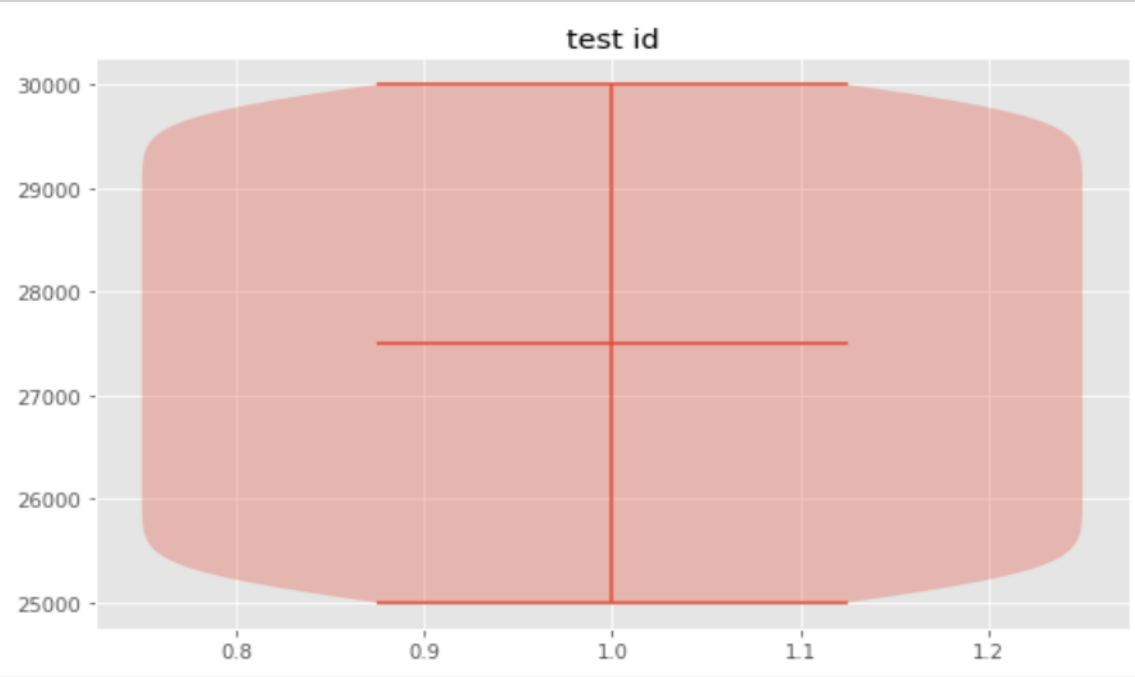



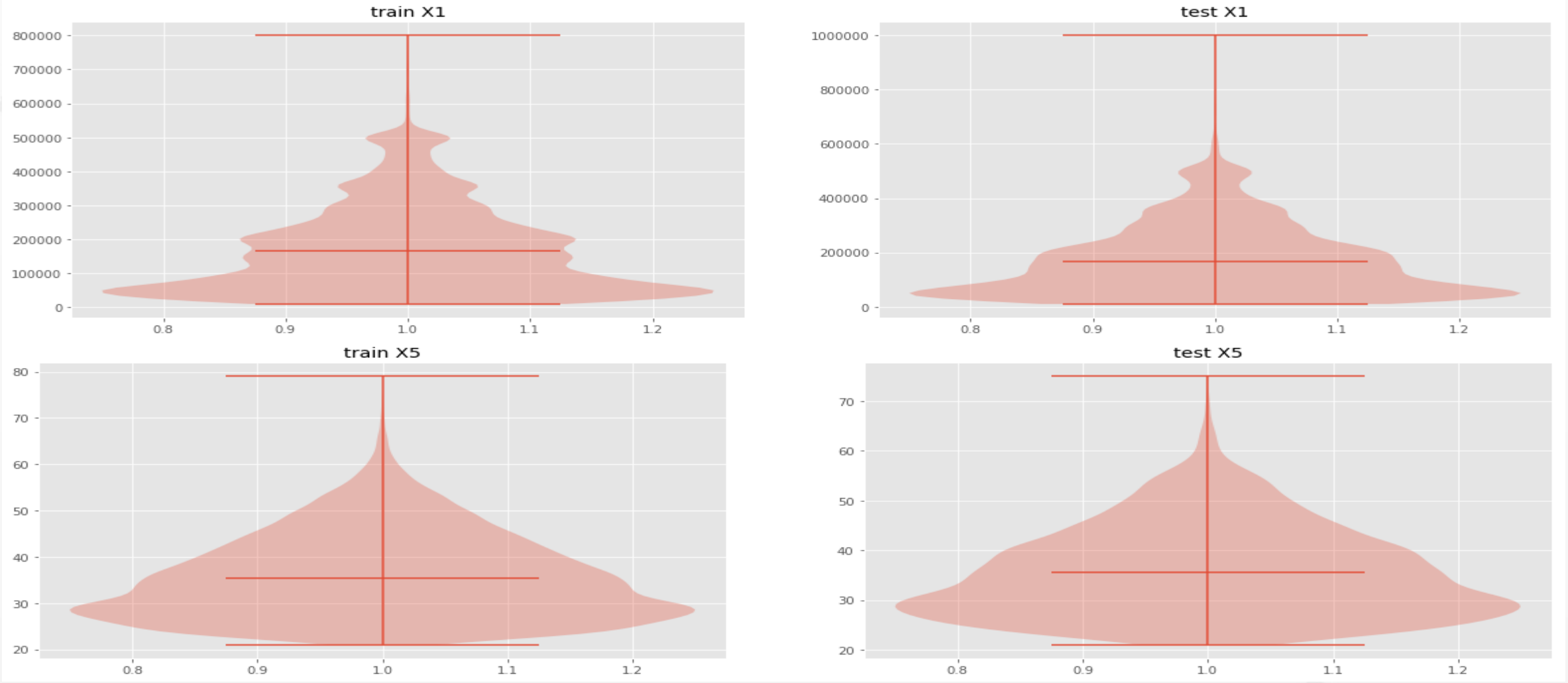
# Violin Plot

```
count=1
for i in range(len(con_v)):
    fig = plt.figure(figsize=(20,100))
    plt.subplot(len(con_v),2,count)
    plt.violinplot(train_data[con_v[i]],showmeans=True)
    plt.title("train "+con_v[i])

    plt.subplot(len(con_v),2,count+1)
    plt.violinplot(test_data[con_v[i]],showmeans=True)
    plt.title("test "+con_v[i])
    count+=2
```



# Continuous



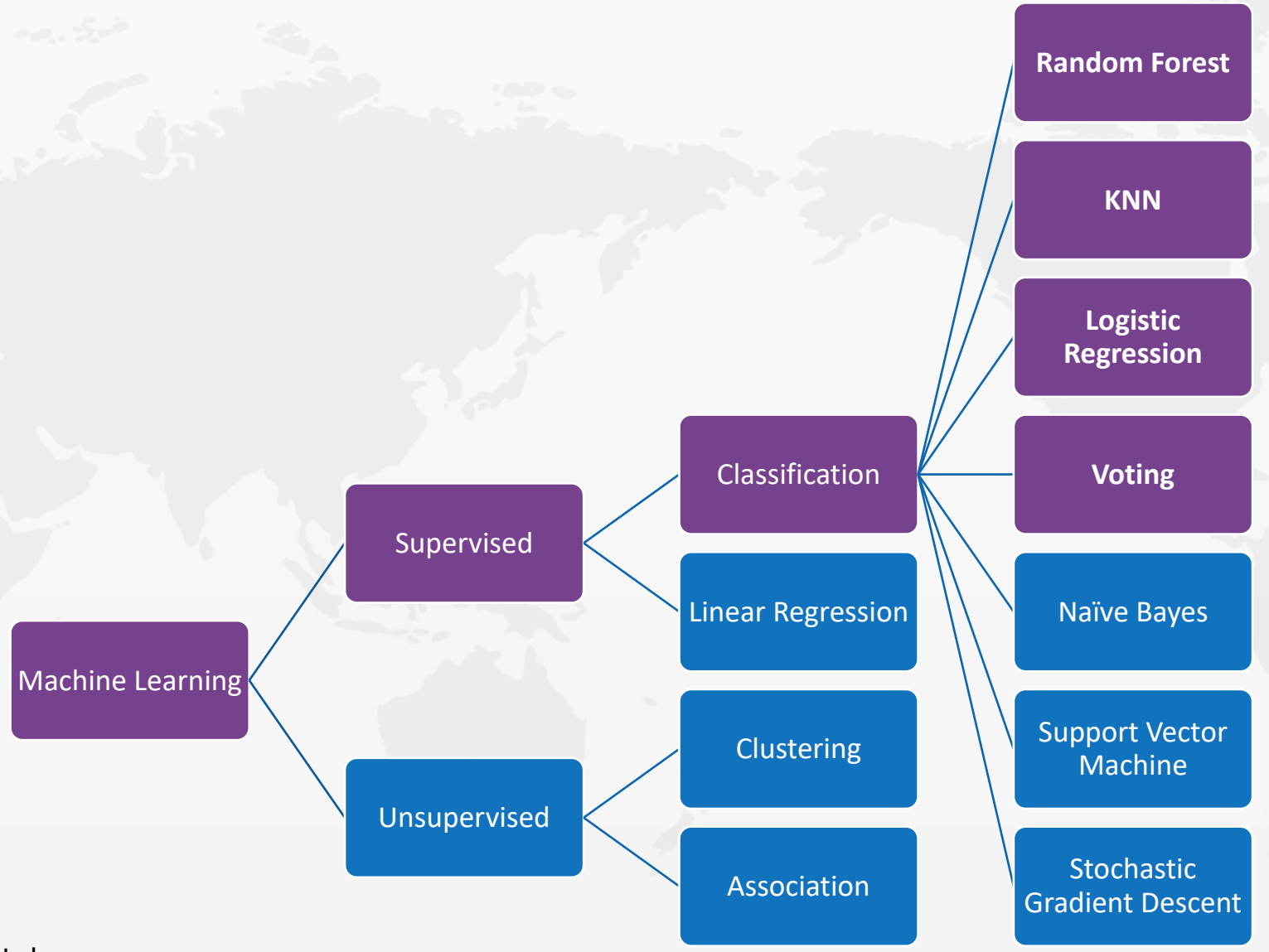
# Proposed Solution

3

# 1. Data Processing

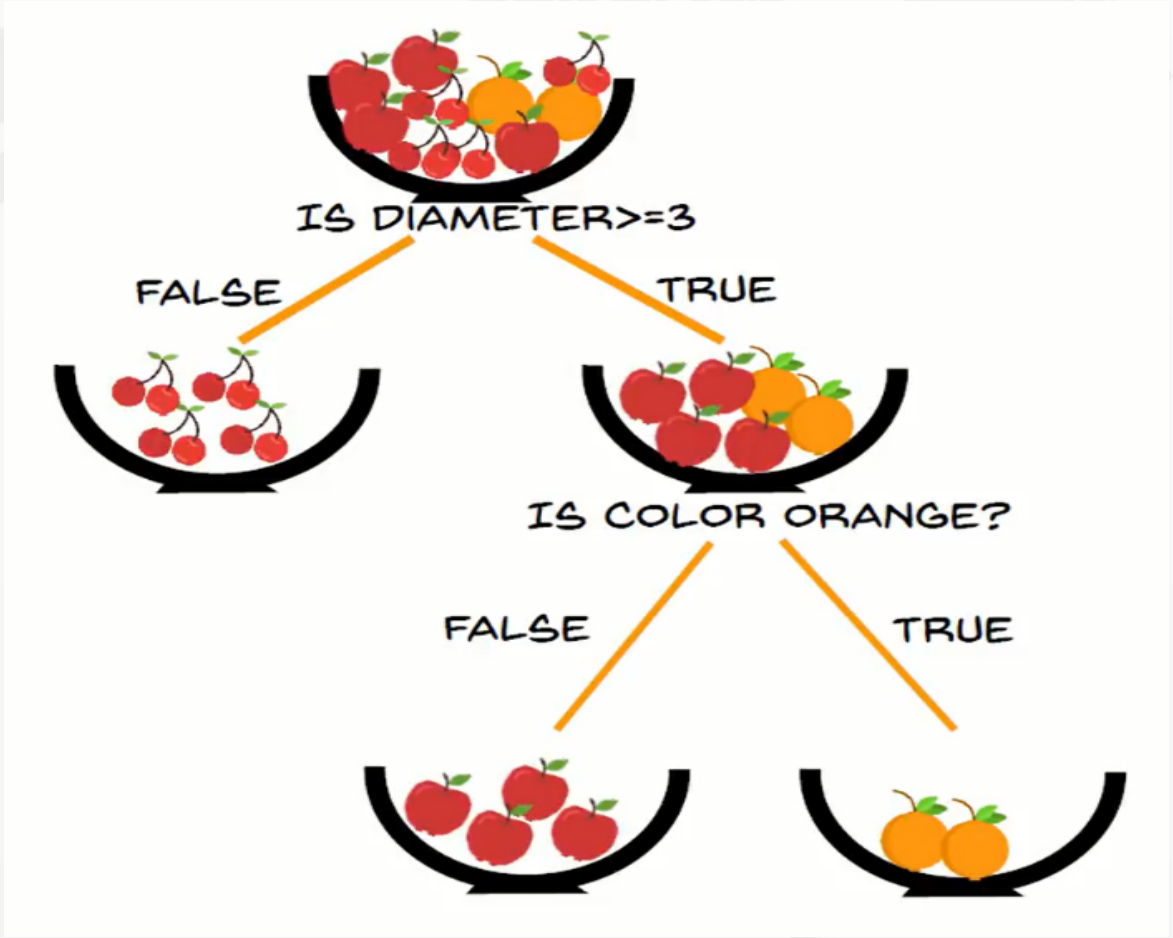
- **Formatting:** The data you have selected may not be in a format that is suitable for you to work with.
- **Cleaning:** Cleaning data is the removal or fixing of missing data. For example, replacing zero with the mean value.
- **Sampling:** There may be far more selected data available than you need to work with.

# 2. Machine Learning Algorithms



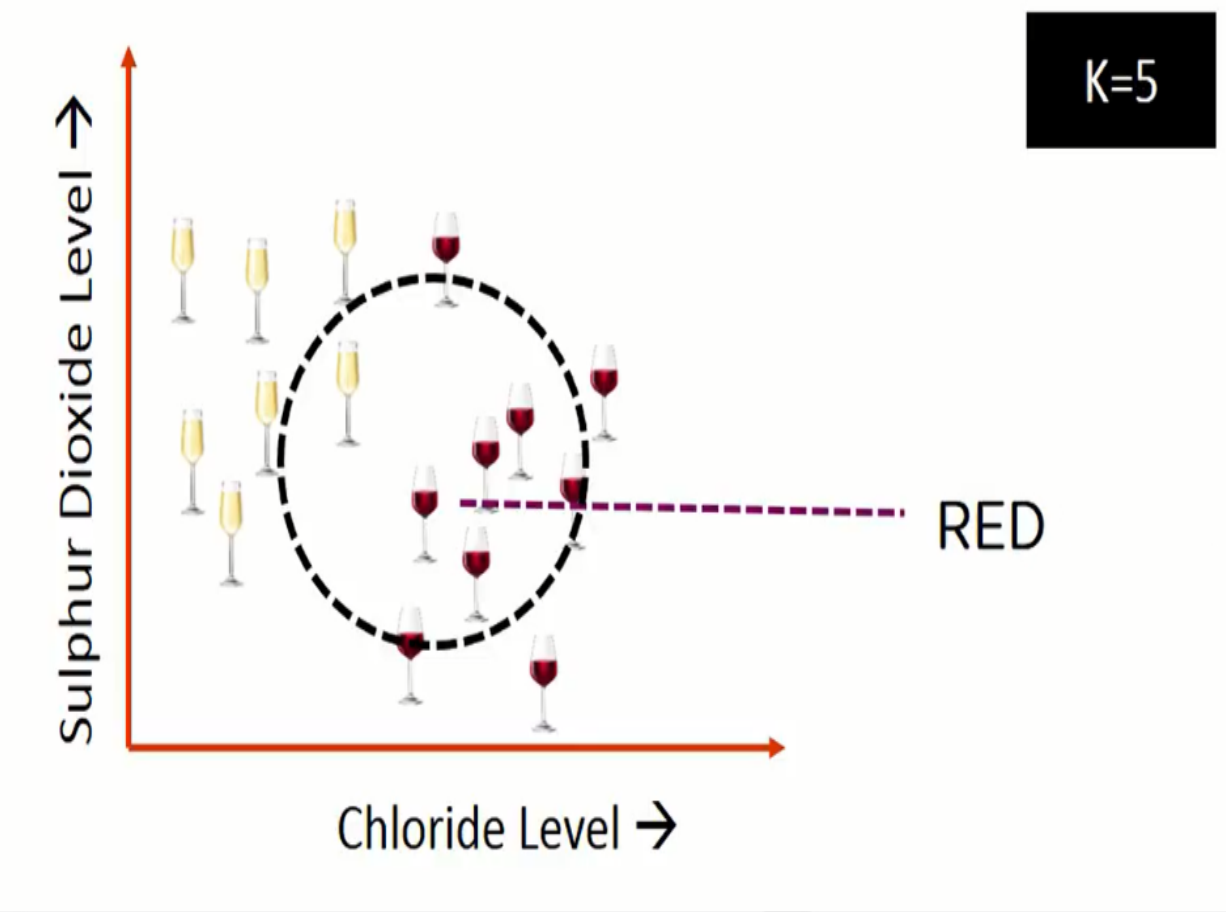
# Random Forest Classifier

- Constructs multiple decision trees and provides decision based on majority of the trees.
- For example, fruit classification.



# K Nearest Neighbor Classifier

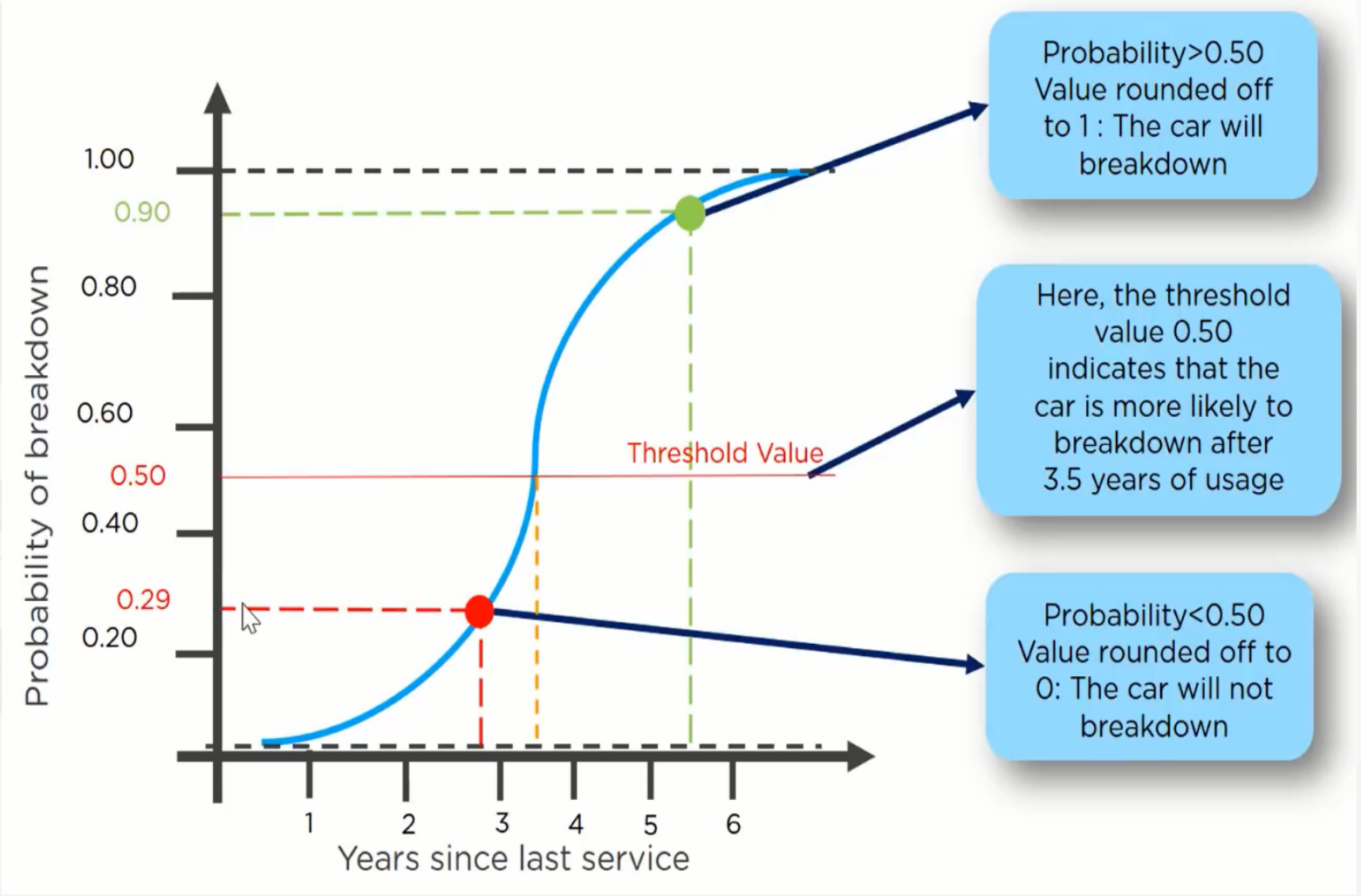
- Depending on value of K, KNN decides wine type as 4 out of 5 nearer wine are red. The algorithm will predict wine in that circle as red wine.



# Logistic Regression

- Regression model shows behavior based on previous data.
- Determine threshold value.

$$Y = \begin{cases} 1, P(X) \geq 0.5 \\ 0, P(X) < 0.5 \end{cases}$$



# Voting Classifier

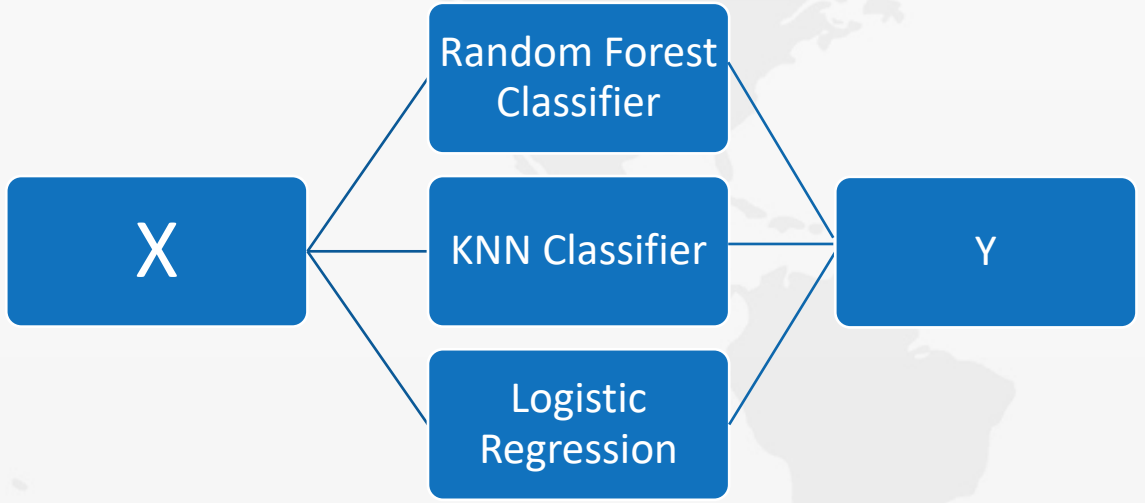
- **Hard Voting:** Is simply majority of voting.

  Y=Mean{X1,X2,X3} ={1,0,0}= 0

- **Soft Voting:** Predict probability based on average of other classifier's results.

  Y =(X1+X2+X3)/3= 1/3<threshold

  => Y=0



# Result and Discussion

4

# Evaluation Metrics

| | NOTDEFAULT PREDICTED | DEFAULT PREDICTED |
|---|---|---|
| NOTDEFAULT | TN | FP |
| DEFAULT | FN | TP |

**True Positive:  Actually Default and Predicted as a Default**

**True Negative: Actually Not-Default and predicted as a Not-Default**

**False Positive:  Actually Not-Default and predicted as a  Default**

**False Negative: Actually Default and predicted as a  Not-Default**

# 3. Evaluation

- **Accuracy**: Accuracy determines number of Not-Default and Default.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:** Amount of actual answers from positive prediction

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:** From all accounts that actually are default accounts.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-Score**: Harmonic mean of precision and recall.

$$\text{F1-Score} = 2 * \frac{P * R}{P + R}$$

- **Cross Validation:**

K Fold cross validation where K=5.

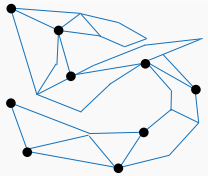
# Accuracy Table

| Classifier | Accuracy (%) |
|---|---|
| Random Forest | 79 |
| K Nearest Neighbor | 71 |
| Logistic Regression | 71 |
| Voting | 71 |

# 4. Analyses of The Best Model

## Random Forest Classifier.

| | Precision | Recall | F1- Score | Support |
|---|---|---|---|---|
| 0 | 0.83 | 0.95 | 0.88 | 3923 |
| 1 | 0.60 | 0.30 | 0.40 | 1077 |
| Avg/Total | 0.78 | 0.81 | 0.78 | 5000 |



# Thank You