Modul 3

Supervised Learning 2

Data Science Program

Purwadhika
Startup and Coding School

Agenda

Session 1: KNN

Session 2: Ensemble model

Session 2: Ensemble model (cont)

Session 3: Exercise

Objective

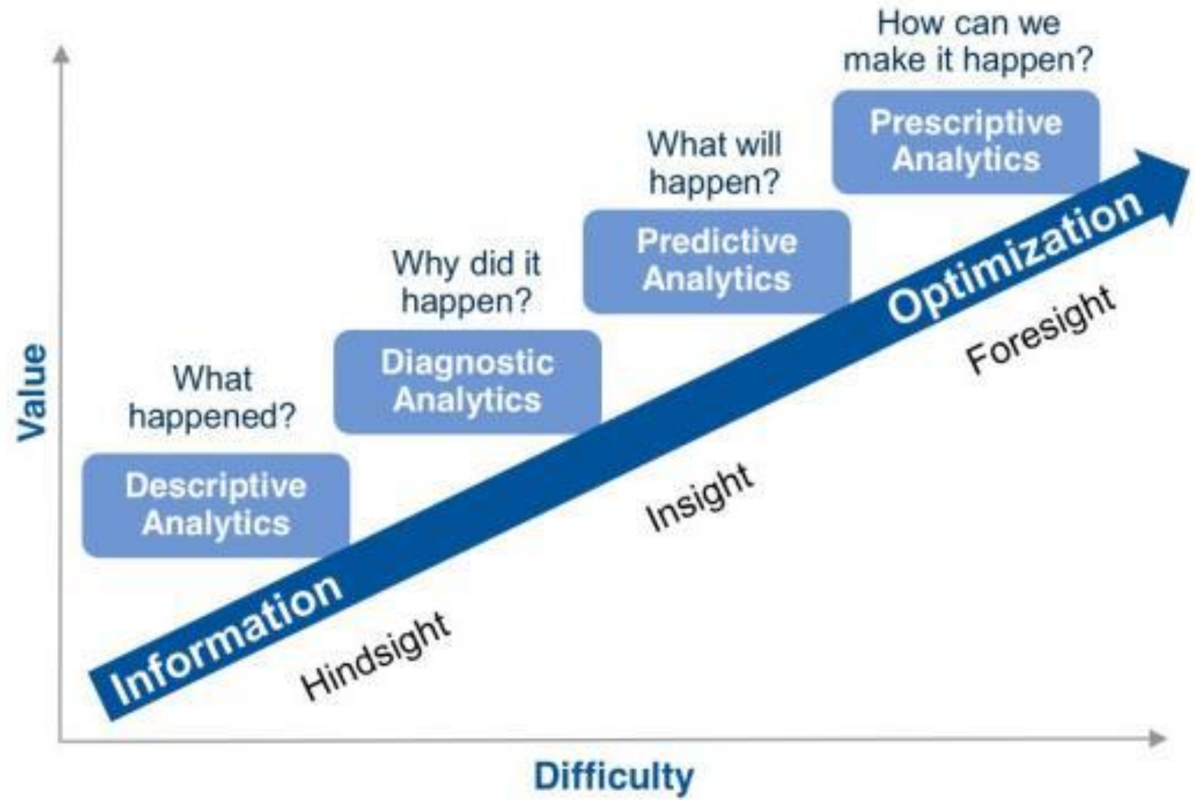
Understand concept of KNN

Understand concept of Ensemble Model

Able to create and evaluate model

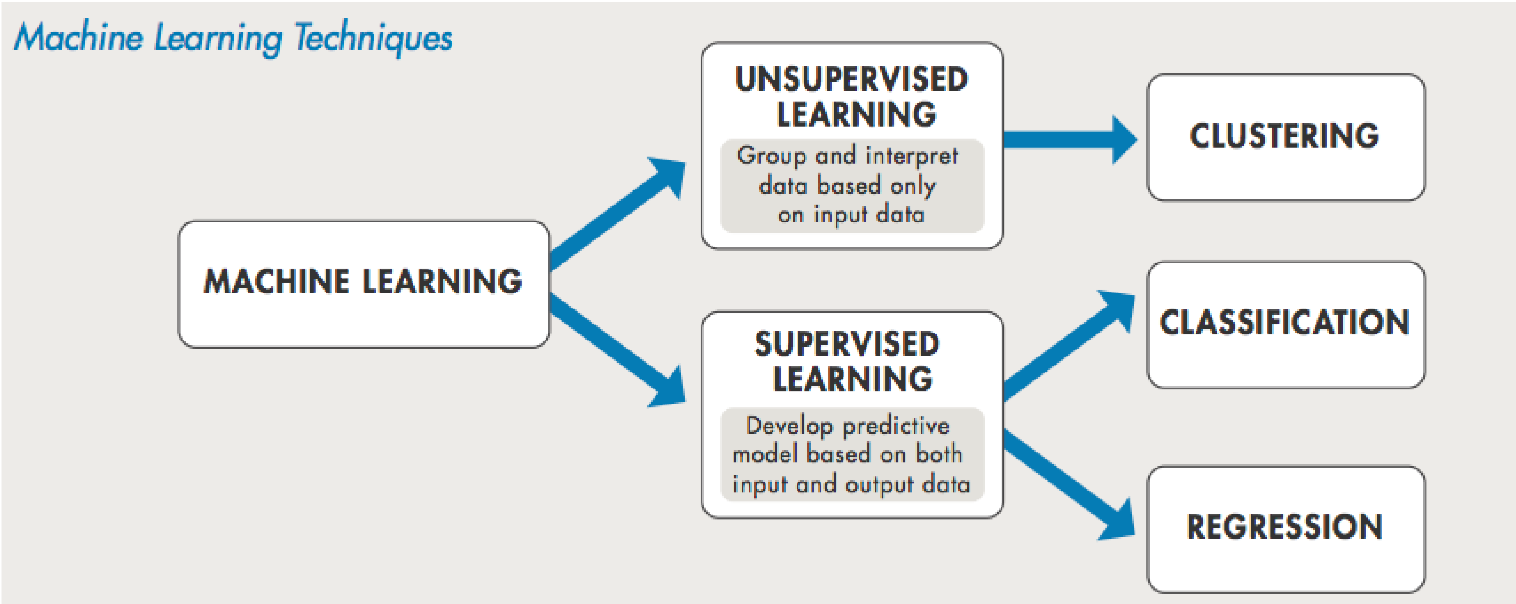
Exposed to Imbalanced Proportion situation

Data Analysis Journey



Purwadhika
Startup and Coding School

Machine Learning Techniques



Purwadhika
Startup and Coding School

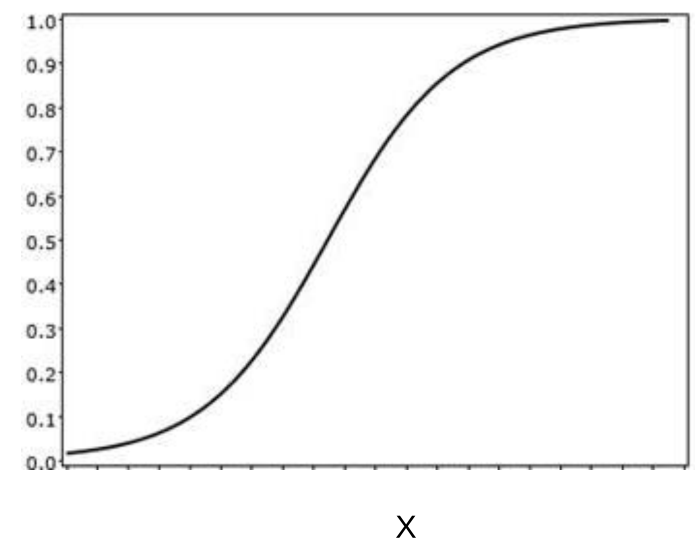
Logistic Regression

$$\text{logit}[\pi(x)] = \log\left[\frac{\pi(x)}{1-\pi(x)}\right] = \alpha + \beta x$$

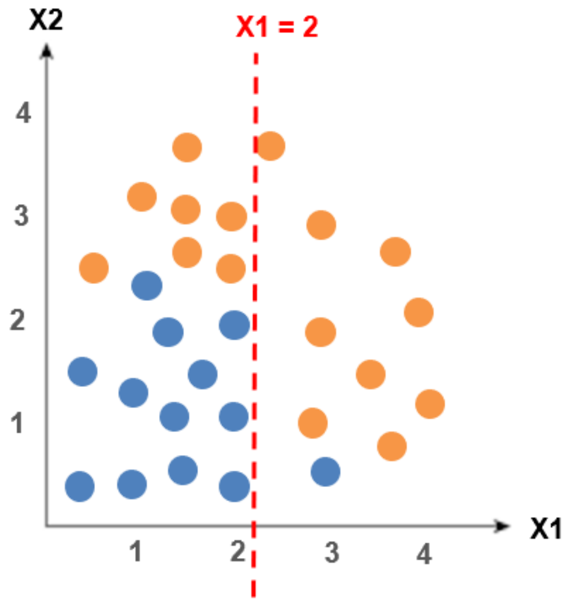
$$\pi(x) = \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta x)}$$

P('Success'|X)

$$\pi(x) = \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta x)}$$

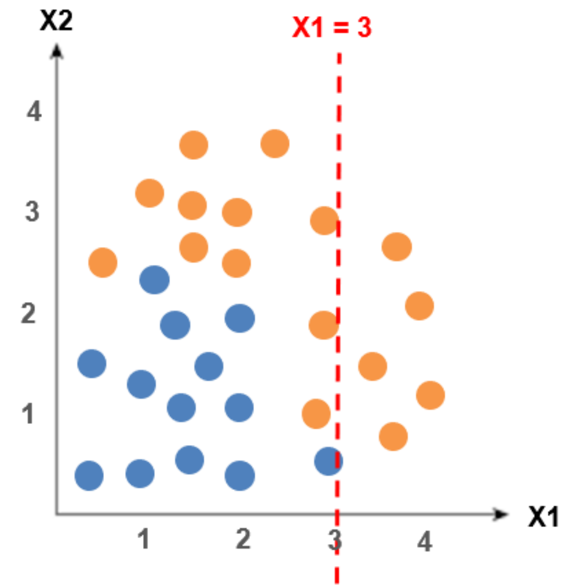


Decision Tree



16 obs.
13 obs.

E = 0.99



7 obs.
12 obs.

9 obs.
1 obs.

E = 0.95

E = 0.47

IG = 0.21

11 obs.
13 obs.

5 obs.
0 obs.

E = 0.99

E = 0

IG = 0.17

Purwadhika
Startup and Coding School

The KNN

Works both for Classification and Regression. But widely used for classification

Non-parametric method.

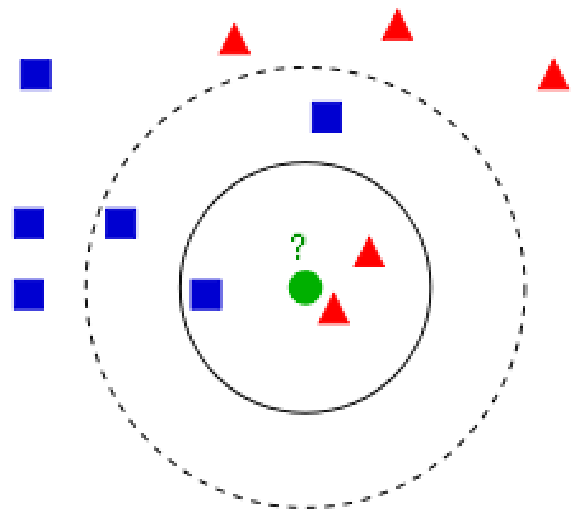
- **When to consider?**
  - Less than 20 attributes per instance
  - Lots of training data

- **Advantage**
  - Easy to program
  - Training is fast
  - Do not lose information

- **Disadvantage**
  - Sensitive to irrelevant/redundant features. All features contribute to the similarity and thus to the classification

Basic Idea

- Store/keep training data
- Classify new observation based on similarity to the observation in training data
- Chosen class is class of k-observation(s) with the closest distance
- Use majority decision rule to classify the records



■ ▲ Training data

● Test data (New observation)

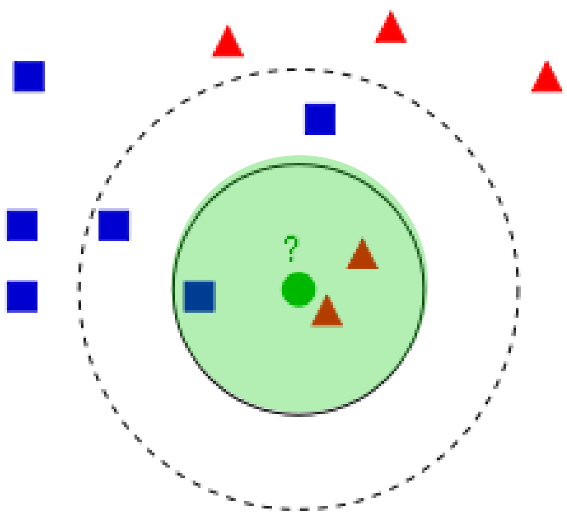
Test data will be classified as ■ or ▲ ?

(Ignore circle line for now)

Basic Idea

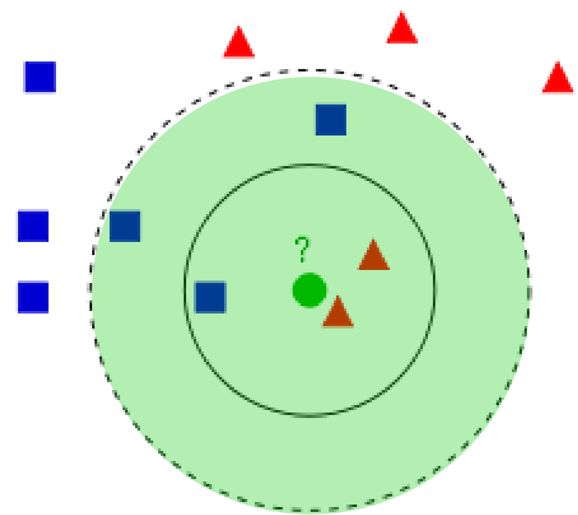
With **k=3** nearest neighbors

Test data classified into ▲



With **k=5** nearest neighbors

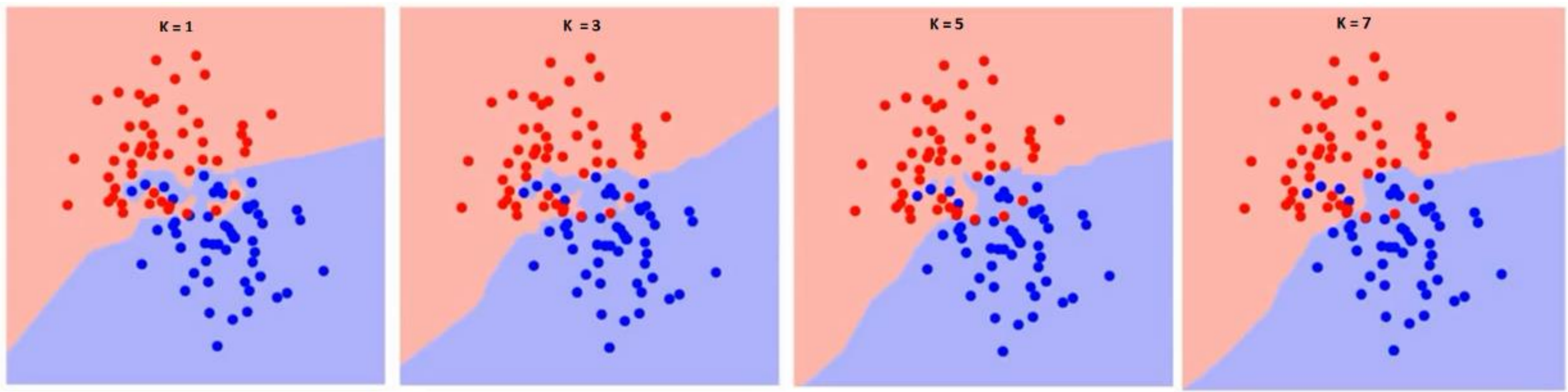
Test data classified into ■



Purwadhika
Startup and Coding School

How do we choose factor K?

k=1 k=3 k=5 k=7

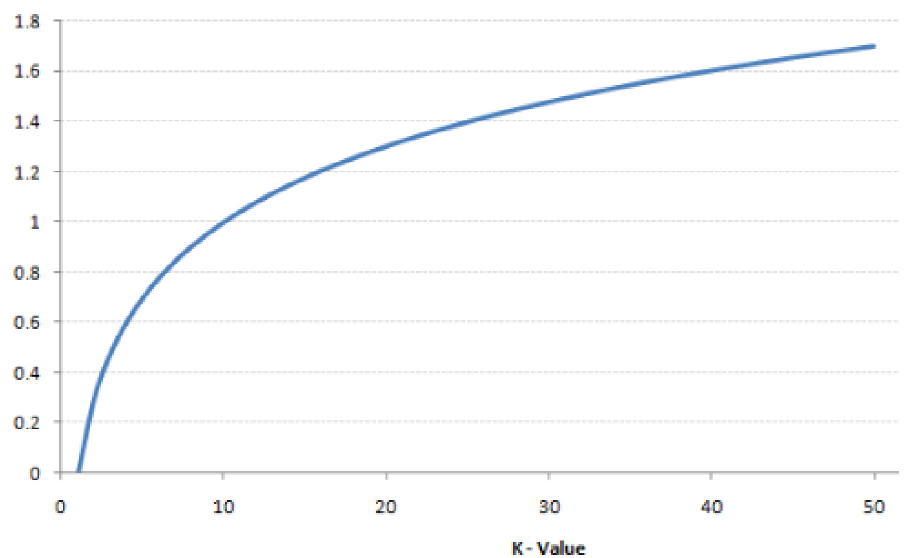


- Boundary becomes smoother with increase value of K
- With K increases to inf, finally becomes all-blue/all-red depending on total majority

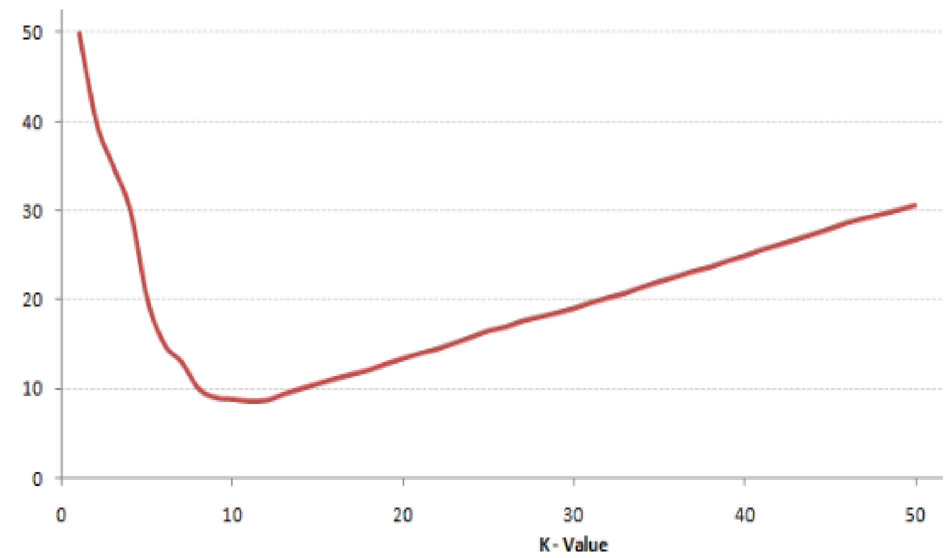
Purwadhika
Startup and Coding School

How do we choose factor K?

Training Error Rate



Validation Error



- Error rate at K=1 always zero for training sample, closest point to any data point is itself

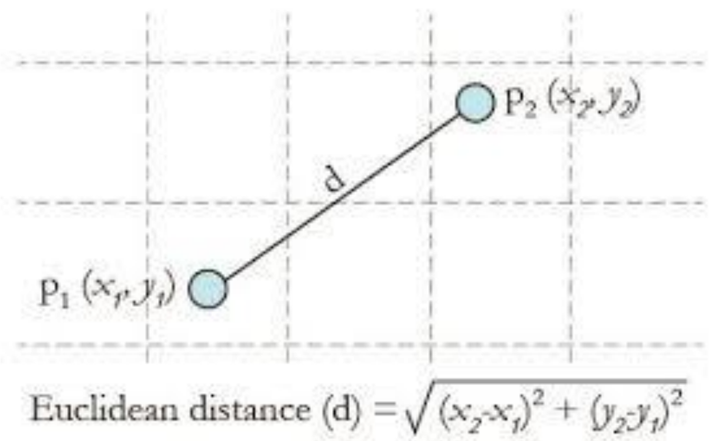
- Overfitting the boundaries at K=1
- Error rate decreases with increases K until reach its minima

- Tips 1: Use odd number of K
- Tips 2: Evaluate using validation or cross validation

Purwadhika
Startup and Coding School

Measuring distance

- Closest neighbor is identified based on the distance
- There are several method to measure distance. The popular one is Euclidean.



Illustration

(x1, y1) = (1, 2)

(x2, y2) = (5, 4)

Euclidean distance

$= \sqrt{(5-1)^2 + (4-2)^2}$

$= \sqrt{(4)^2 + (2)^2}$

$= \sqrt{20}$

= 4.47

Purwadhika
Startup and Coding School

Issue with distance

- Given X is Area with unit of hectare
- Given Y is Corn Production with unit of kg.

Illustration

(x1, y1) = (3, 2000)

(x2, y2) = (5, 4000)

Euclidean distance

$$= \sqrt{(5-3)^2 + (4000-2000)^2}$$

$$= \sqrt{(2)^2 + (2000)^2}$$

$$= \sqrt{4\,000\,004}$$

$$= 2000$$

Distance contribution from X is surpassed by contribution from Y due to different scale.

Variable with large scale will have larger effect on the distance.

Purwadhika
Startup and Coding School

Issue with distance

Solution to solve scale issue is **Normalization**.

- **Min-Max Scaling**

  Uses *MinMaxScaler*

  Transform to defined range

$$y = \frac{x - \min x_i}{\max x_i - \min x_i}$$

- **Standardization**

  Uses *StandardScaler*

  Transform to mean=0, sd=0

$$y = \frac{x - \bar{x}}{s}$$

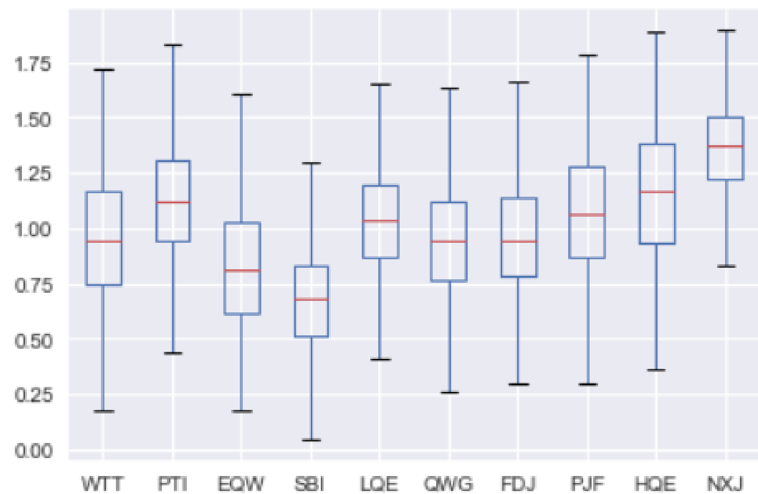
Where

$\bar{x}$ = mean

s = Standard deviation

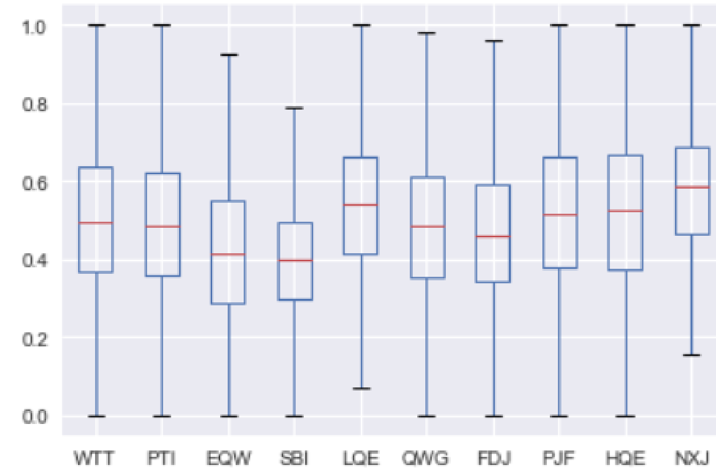
Purwadhika
Startup and Coding School

Illustration



Original Data

MinMaxScaler



StandardScaler

Illustration

Objective: Predict class of new data point using KNN

Data: Provided data from a company, but due to its confidentiality, variable name is masked.

```
df.shape
```

```
(1000, 11)
```

```
df.head()
```

	WTT	PTI	EQW	SBI	LQE	QWG	FDJ	PJF	HQE	NXJ	TARGET CLASS
0	0.913917	1.162073	0.567946	0.755464	0.780862	0.352608	0.759697	0.643798	0.879422	1.231409	1
1	0.635632	1.003722	0.535342	0.825645	0.924109	0.648450	0.675334	1.013546	0.621552	1.492702	0
2	0.721360	1.201493	0.921990	0.855595	1.526629	0.720781	1.626351	1.154483	0.957877	1.285597	0
3	1.234204	1.386726	0.653046	0.825624	1.142504	0.875128	1.409708	1.380003	1.522692	1.153093	1
4	1.279491	0.949750	0.627280	0.668976	1.232537	0.703727	1.115596	0.646691	1.463812	1.419167	1

Purwadhika
Startup and Coding School

Illustration

Data Preparation: Standardize the variables. In this case using StandardScaler

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
```

```
scaler.fit(df.drop('TARGET CLASS',axis=1))
```

```
StandardScaler(copy=True, with_mean=True, with_std=True)
```

```
scaled_features = scaler.transform(df.drop('TARGET CLASS',axis=1))
```

```
df_feat = pd.DataFrame(scaled_features,columns=df.columns[:-1])
df_feat.head()
```

Before normalization

After normalization

adhika Coding School

Illustration

Modeling:

Train-Test Split

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(scaled_features,df['TARGET CLASS'], test_size=0.30)
```

```
print('x_train ',x_train.shape)
print('y_train ',y_train.shape)
print('x_test ',x_test.shape)
print('y_test ',y_test.shape)
```

```
x_train  (700, 10)
y_train  (700,)
x_test  (300, 10)
y_test  (300,)
```

Purwadhika
Startup and Coding School

Illustration

Modeling:

Using KNN, start with k=1

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn = KNeighborsClassifier(n_neighbors=1)
```

```
knn.fit(x_train,y_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=1, p=2,
           weights='uniform')
```

```
pred = knn.predict(x_test)
```

Prediction and Evaluation

```
from sklearn.metrics import classification_report,confusion_matrix
```

```
print(confusion_matrix(y_test,pred))
```

```
[[132  15]
 [ 10 143]]
```

```
print(classification_report(y_test,pred))
```

```
             precision    recall  f1-score   support

          0       0.93      0.90      0.91       147
          1       0.91      0.93      0.92       153

avg / total       0.92      0.92      0.92       300
```

Purwadhika
Startup and Coding School

Illustration

Modeling:

Choosing K-Value. Choose K=19?

```
error_rate = []

# Will take some time
for i in range(1,30):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_train,y_train)
    pred_i = knn.predict(x_test)
    error_rate.append(np.mean(pred_i != y_test))
```

```
plt.figure(figsize=(10,6))
plt.plot(range(1,30),error_rate,color='blue',
         linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
```

Purwadhika
Startup and Coding School

Illustration

Modeling:

Comparison K=1 and K=19

```
# FIRST A QUICK COMPARISON TO OUR ORIGINAL K=1
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(x_train,y_train)
pred = knn.predict(x_test)

print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
[[136  13]
 [ 14 137]]
             precision    recall  f1-score   support

          0       0.91      0.91      0.91       149
          1       0.91      0.91      0.91       151

avg / total       0.91      0.91      0.91       300
```

```
# NOW WITH K=21
knn = KNeighborsClassifier(n_neighbors=19)
knn.fit(x_train,y_train)
pred = knn.predict(x_test)

print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
[[139  10]
 [  5 146]]
             precision    recall  f1-score   support

          0       0.97      0.93      0.95       149
          1       0.94      0.97      0.95       151

avg / total       0.95      0.95      0.95       300
```

Purwadhika
Startup and Coding School

Exercise

Use Case: Predict Default customer

Data: Debtor Dataset

Col Name	Description
NAMA	Debtor Name
T	Event Time
Y	Upselling Status
X1	Sex
X2	Age
X3	Marital status
X4	# Dependents
X5	Home ownership
X6	Education

Col Name	Description
X7	Area
X8	Occupation
X9	Length Of Work
X10	Income
X11	Debt Burden Ratio
X12	Term of Credit (mo)
X13	Credit Limit
X14	Credit Quality
class	Classification

Predictor:

X4, X9, X11, X12, X13

Response

Class (1=Not Default, 0=Default)

Ensemble Model Idea

- Combining two or more algorithms (base learners) in order to make more robust system.
- Suppose we want to predict customer default just like we did yesterday. We can use different model such as Logistic Regression, KNN, Decision Tree, etc.

ID	RegLog	DT	KNN	Actual
1	1	1	1	1
2	1	0	0	1
3	1	0	0	0
4	0	0	1	0
5	1	1	1	1
….				

For example, accuracy are:

Reglog : 82%

DT : 79%

KNN: 70%

Which algorithm to choose?

Purwadhika
Startup and Coding School

Type of Ensemble

Type of ensemble:

- **Averaging**

Model1	Model2	Model3	AveragePrediction
45	40	65	50

- **Majority Vote**

Model1	Model2	Model3	VotingPrediction
1	0	1	1

- **Weighted average**

	Model1	Model2	Model3	WeightAveragePrediction
Weight	0.4	0.3	0.3	
Prediction	45	40	60	48

Purwadhika
Startup and Coding School

Ensemble Model Idea

- There's no algorithm that always accurate
- Each algorithm use different Algorithm, Hyperparameter, Training set, Hypothesis

Imagine a meeting room with experts from different background discussing company stock. Everyone could contribute opinion or suggestion based on their individual point of view. The opinion will be varied.

Taking account the opinion as input will result more robust final decision, more accurate and less likely to be biased.

- Ensemble model gives the global picture

Challenge

Challenge in developing ensemble models:

- Not to obtain highly accurate base model, but rather to obtain base model which make different kind of errors

- High accuracy can be accomplished if differential base model misclassify different training examples, even if the base classifier accuracy is low.

Purwadhika
Startup and Coding School

Adv-Disadvantage

Advantages:

- Proven method to improve accuracy of the model
- Model more robust and stable in most scenarios
- For those who love competition, this ingredient wins almost all competition

Disadvantages:

- Reduce model interpretability
- Time consuming
- Base model to ensemble is hard to master

Purwadhika
Startup and Coding School

BAGGING

- Bagging : Bootstrap Aggregating
- Bootstrap : Sampling technique in which we choose ‘n’ rows out from ‘n’ rows original dataset with replacement.

Data	Bootstraped Sample
Row 1	
Row 2	
Row 3	

Data	Bootstraped Sample
Row 1	Row 2
Row 2	
Row 3	

Iteration 1

Data	Bootstraped Sample
Row 1	Row 2
Row 2	Row 1
Row 3	

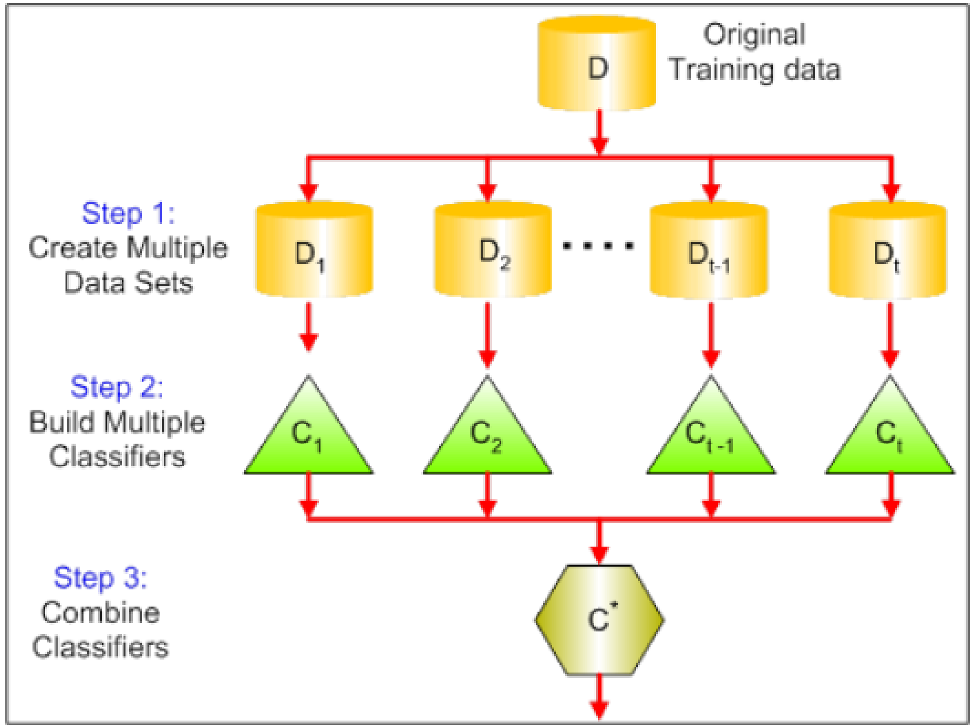
Iteration 2

Data	Bootstraped Sample
Row 1	Row 2
Row 2	Row 1
Row 3	Row 1

Iteration 3

- Aggregate individual learners
- Use Voting

Bootstrap Aggregating



Purwadhika
Startup and Coding School

Random Forest

- Just like Bagging, Random Forest use multiple trees.
- Each tree gives classification. We say the tree ‘votes’ for that class
- The forest choose the classification having the most votes over all trees in forest

Purwadhika Startup and Coding School

How RF works

- Assume number of cases in the training set is N. Sample of these N cases is taken at random but *with replacement*. This sample will be the training set.

- If there are M input variables, a number m<M is specified such that at *each node*, m variables are selected at random out of the M. The best split on these m is used to split the node. The value of m is held constant while we grow the forest.

- Each tree is grown to the largest extent possible and there is no pruning.

- Predict new data by aggregating the predictions of the ntree trees (i.e., majority votes for classification, average for regression)

Purwadhika
Startup and Coding School

Adv-Disadvantage

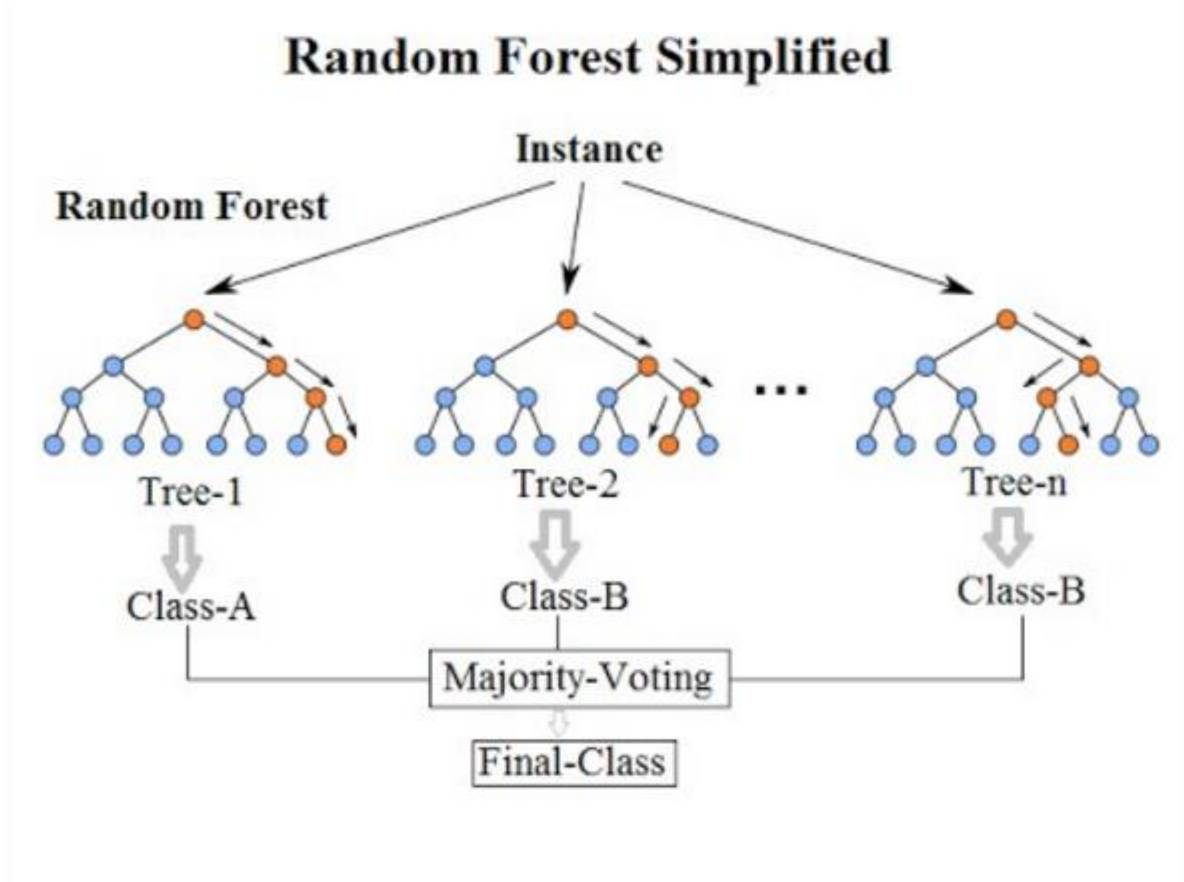
Advantages:

- Can solve both classification and regression case
- Handle high dimensionality data and results Importance of variable, a handy features
- Has methods of balancing error in data sets where classes are imbalanced
- Data not used for training during bootstrapping is used for testing data, called out-of-bag samples. Error estimated on these data is accurate.

Disadvantages:

- It works better for classification, but not for regression
- Feel like black-box approach. Very little control to what the software does.

How RF works



https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

Exercise

Banks play a crucial role in market economies. **They decide who can get finance and on what terms and can make or break investment decisions.**

Credit scoring algorithms, which make a guess at the probability of default, are the method banks use to determine whether or not a loan should be granted. This competition **requires participants to improve on the state of the art in credit scoring, by predicting the probability that somebody will experience financial distress in the next two years**.

Exercise

Challenge

Different to the exercise before, this dataset has imbalanced proportion of response variable.
Proportion of Default / Not-Default = 0.06 / 0.94

Use several algorithm and compare the result.
Compare the model evaluation metrics (Accuracy, Specificity, Sensitivity, etc).
Do you see high accuracy and low sensitivity?
What do you think causes this situation?

It’s a wrap

- KNN concept
- Distance calculation
- Variable normalization
- Define best factor K
- Ensemble model
- Bootstrap concept
- Bagging concept
- Random Forest concept

Purwadhika
Startup and Coding School