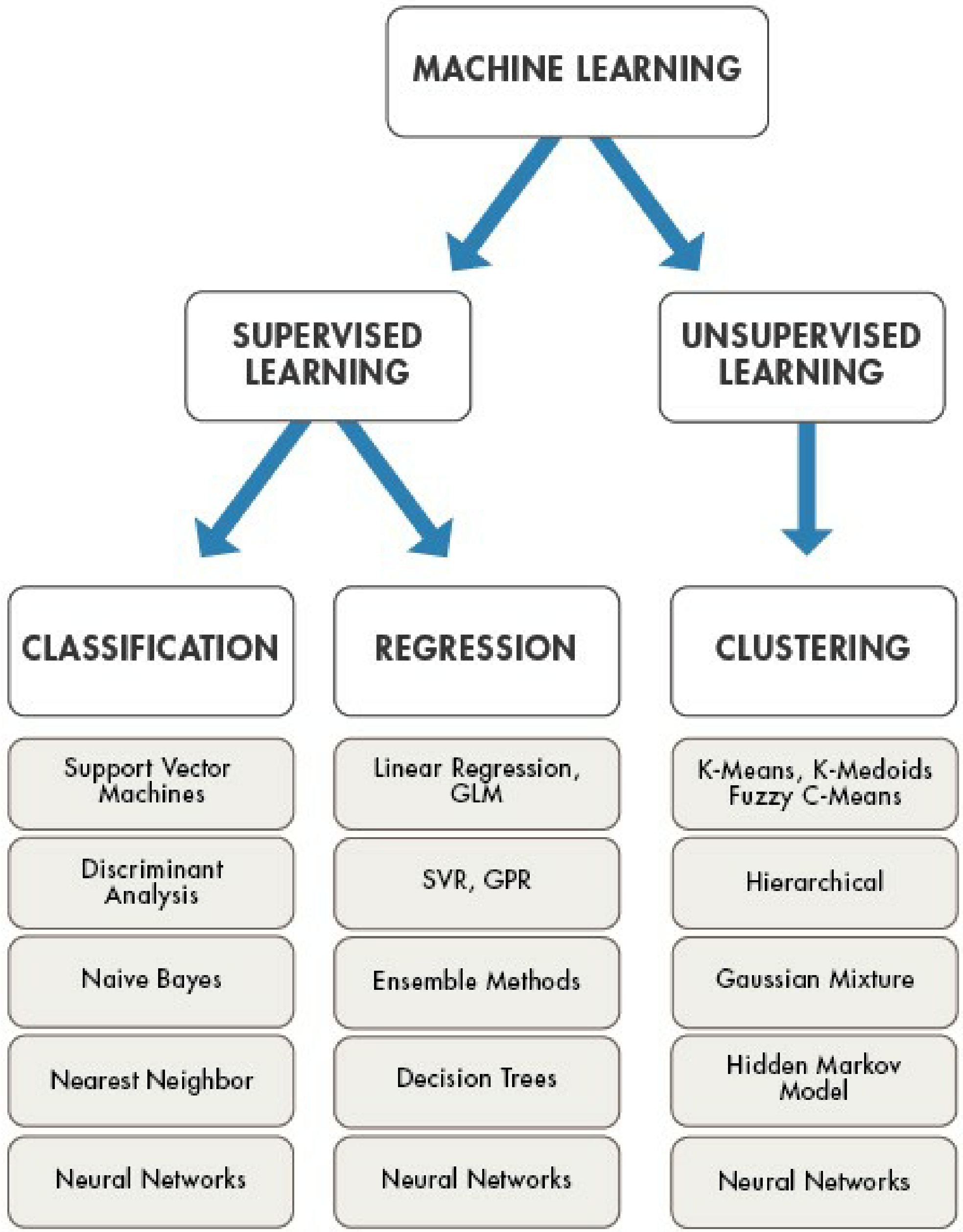
Text Analysis

AMAZON FINE FOOD REVIEWS


amazon.com


MACHINE LEARNING
SUPERVISED LEARNING
UNSUPERVISED LEARNING
CLASSIFICATION
REGRESSION
CLUSTERING
Support Vector Machines
Discriminant Analysis
Naive Bayes
Nearest Neighbor
Neural Networks
Linear Regression, GLM
SVR, GPR
Ensemble Methods
Decision Trees
Neural Networks
K-Means, K-Medoids Fuzzy C-Means
Hierarchical
Gaussian Mixture
Hidden Markov Model
Neural Networks

Attribute Information

- ID
- Product id
- User id
- Profile Name
- Helpfulness Numerator
- Helpfulness Denominator
- Score - 1 to 5
- Time
- summary
- Text

CONTENTS

ABSTRACT

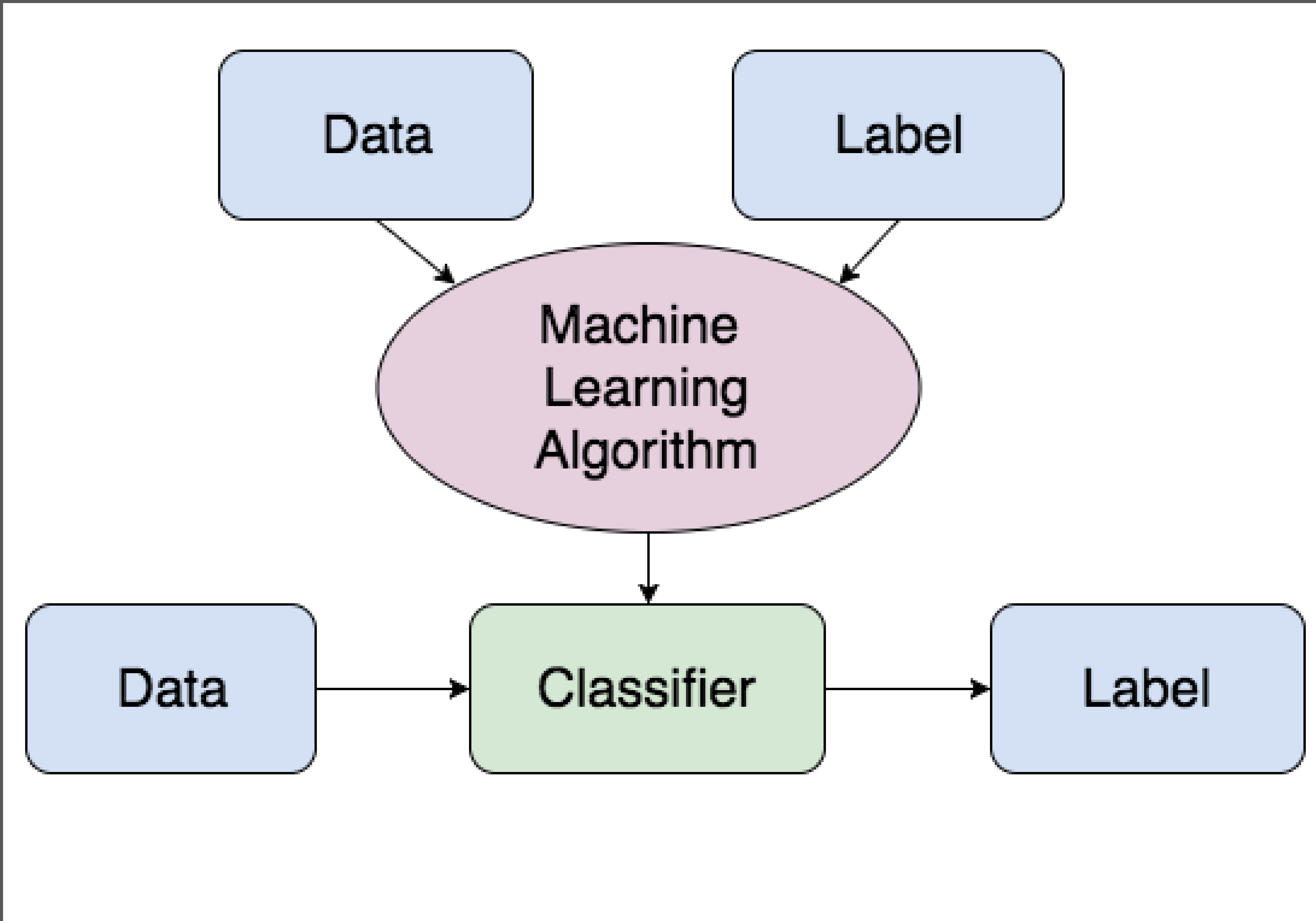
- This is an Machine Learning project which can be used in E-commerce websites. The development of the Internet changed the way people eat and responding for food. Amazon is a biggest website where users can easily purchase all kind of food they need with only a mouse-click. Here our aim is to create a classifier that classifies the reviews in to either positive or negative based on the data given such as the time at which the review was written, rating given by the customer. We can analyze the using various models like Naive Bayes, KNN, Logistic Regression etc.

Eventually we want to build an acceptable model which helps us better understand how customers rate and review the food they purchased.

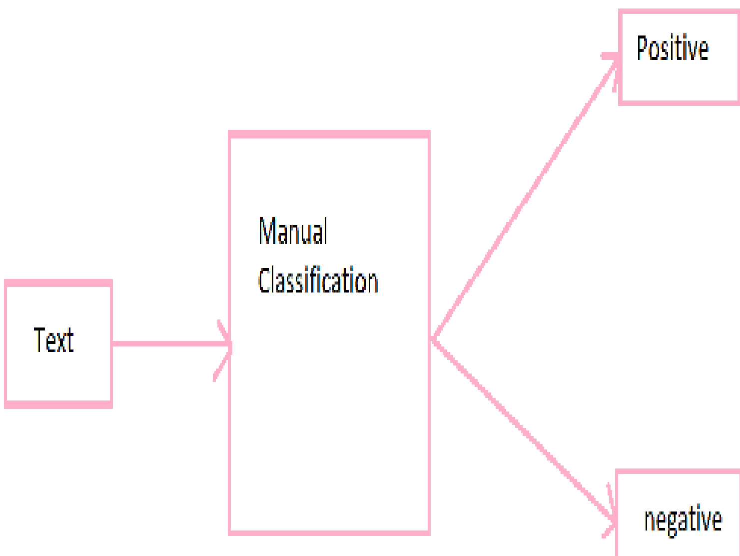
Here, we classified the reviews into 2 categories

- 1) Positive: which indicates the customer is satisfied with the product
- 2) Negative: which indicates the customer is not satisfied with the product

The main propaganda of this project is, it can be used in case of e-commerce websites where the review plays a vital role based on which the interest of the new customers depends.


Data
Label
Machine
Learning
Algorithm
Data
Classifier
Label

EXISTING SYSTEM



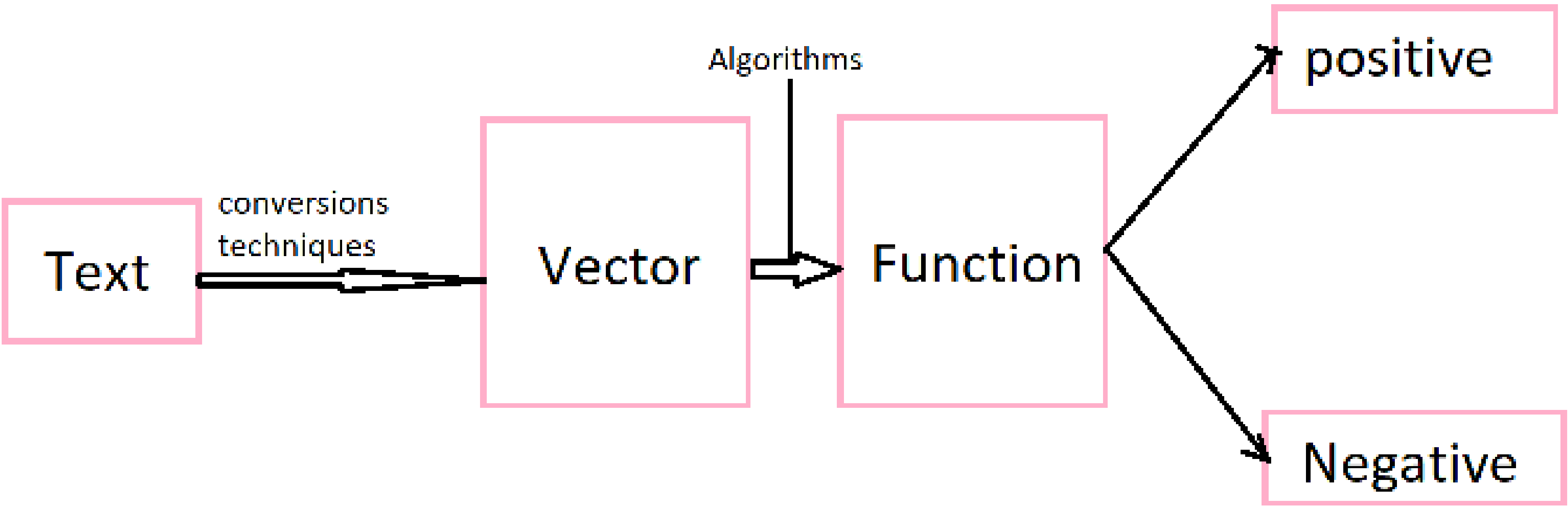
In early days review are classified manually but

at present the as review has been increased

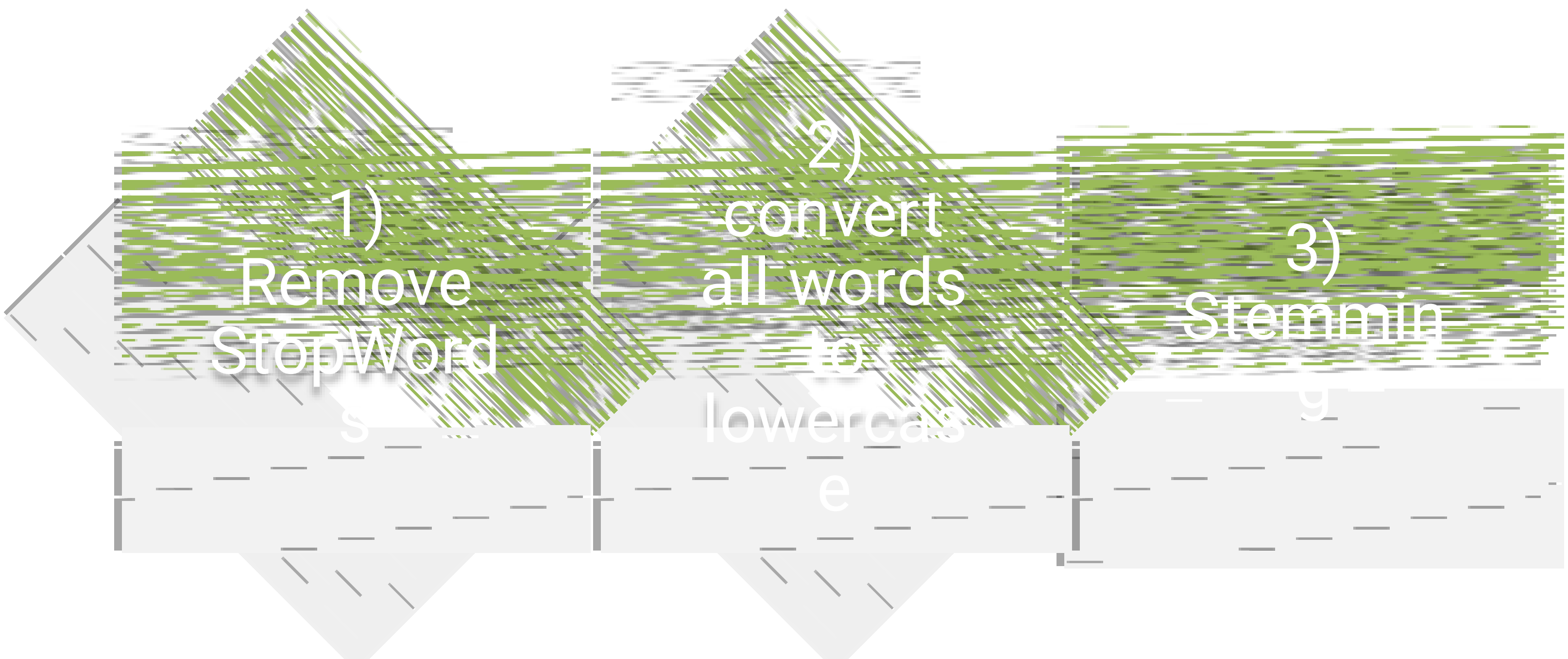
exponentially, it is impossible to do manually

so automation is required.

PROPOSED SYSTEM:



Text Pre_Processing



Text toVector techniques

1

1. BOW(Bag Of Words)

2

2.TF-IDF(Term frequency Inverse Document Frequency

3

3.W2V (word to vector)

Bag Of Words

- T1: the dog is on the table
- T2: now the cats are on the table

Limitation example

- T3: This is a jntu college
- T4: This is not a jntu college

TF-IDF

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Formulas

1)TF(W,R)=No of times w occur in R/Total no of word in R

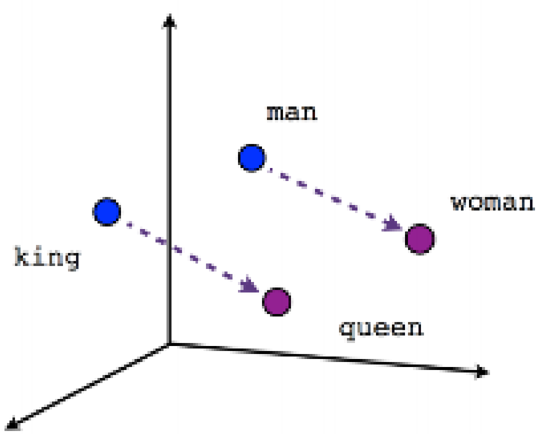
0<=TF(W,R)<=1

2)IDF(W,D)= Log(N/ni)

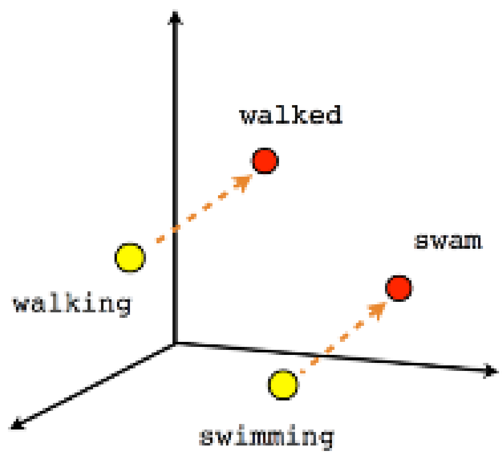
N---> Total no of Documents

ni--->No of documents contain w

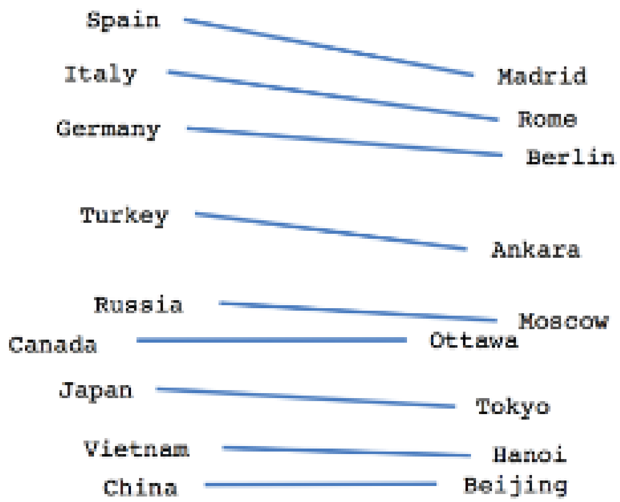
W2V(Word to Vector)



Male-Female



Verb tense



Country-Capital

Algorithms

- KNN(k_nearest-neighbors)
- Logistic Regression
- Decision Tree
- Random Forest
- Naive Bayes

Paramete r Tuning

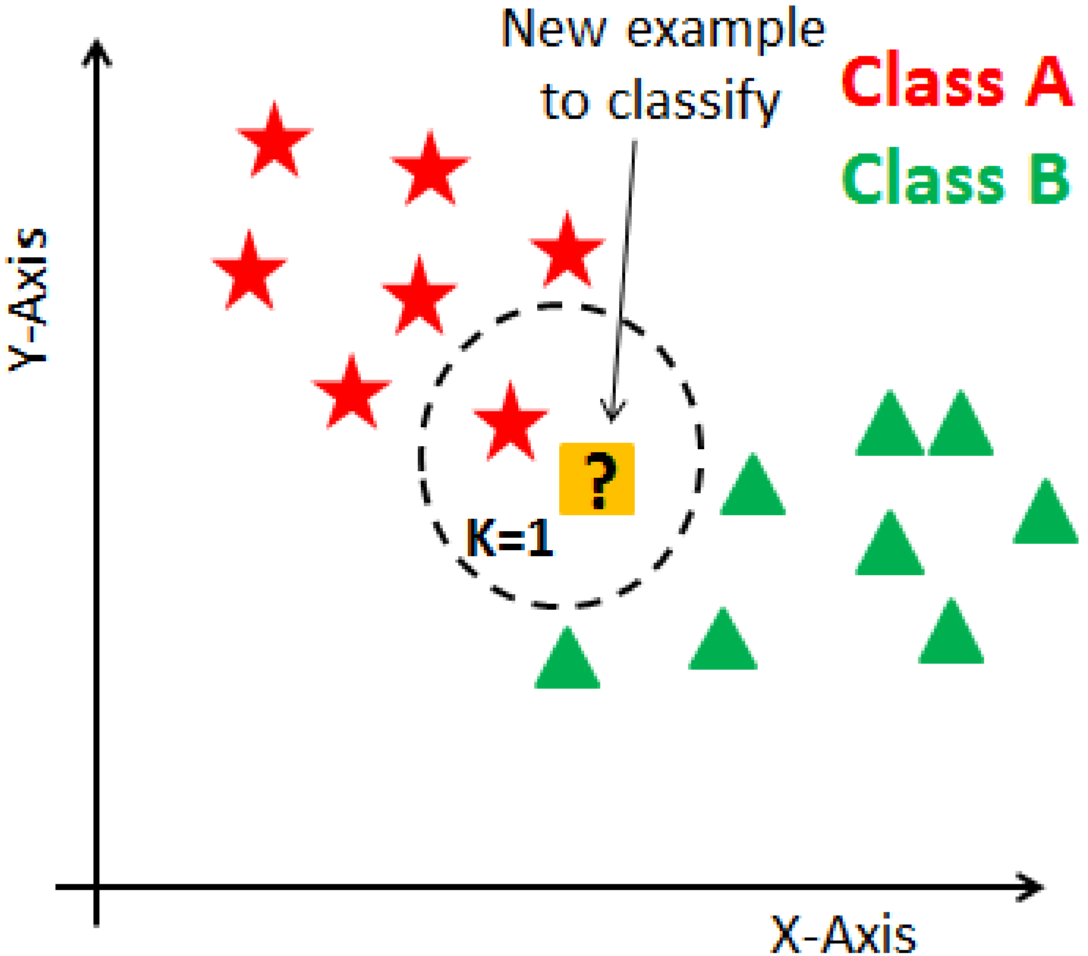
- GridSearchCV
- RandomizedSearch

fit KNN model to data after w2v

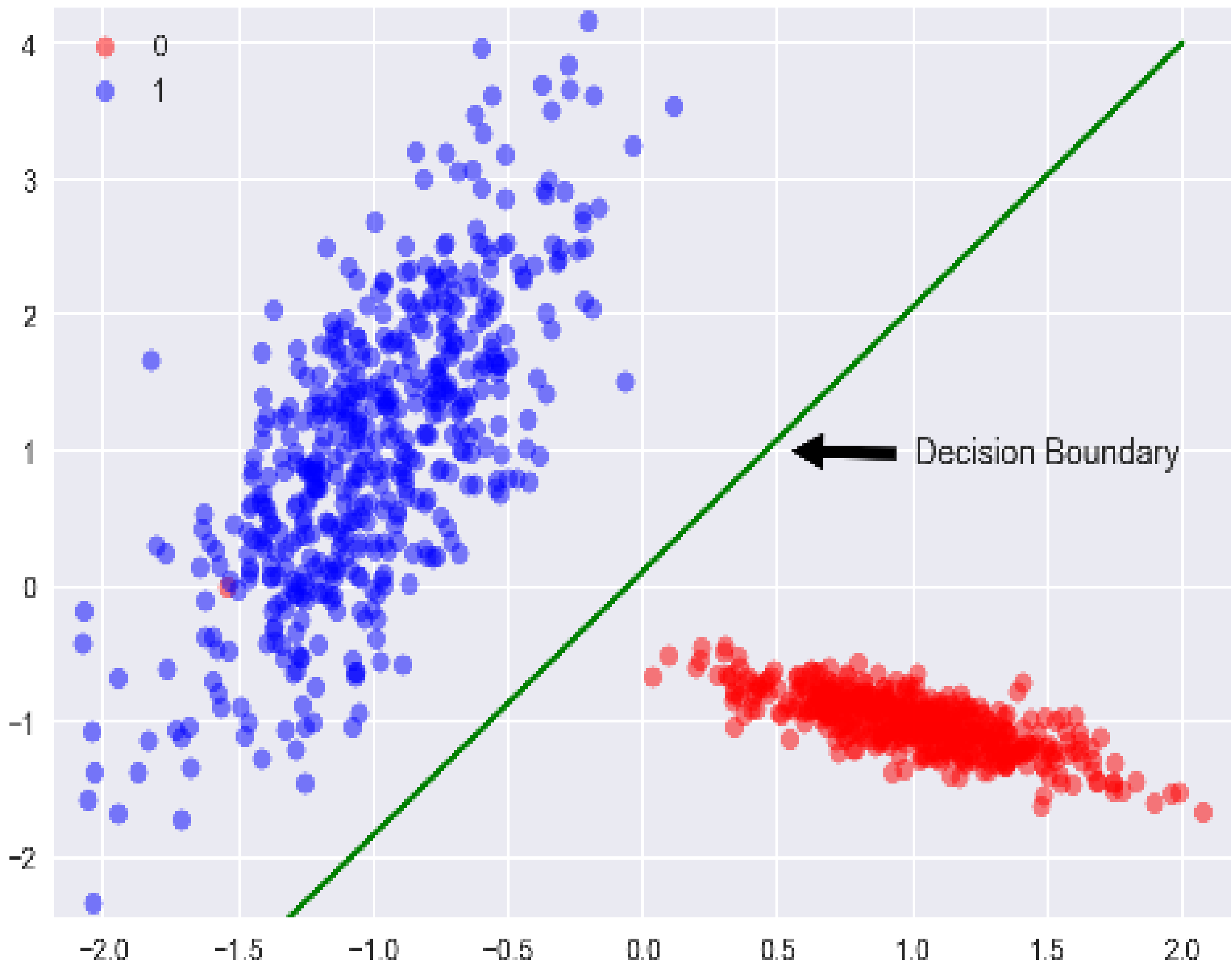
```
In [69]: from sklearn.model_selection import GridSearchCV
         from sklearn.neighbors import KNeighborsClassifier
         parameters = {'n_neighbors':[1,2,3,4,5,6,7,8]}
         neigh = KNeighborsClassifier()
         clf = GridSearchCV(neigh, parameters, cv=5, scoring="accuracy")
         clf.fit(x2_train,y2_train)
         clf.best_params_

Out[69]: {'n_neighbors': 7}
```

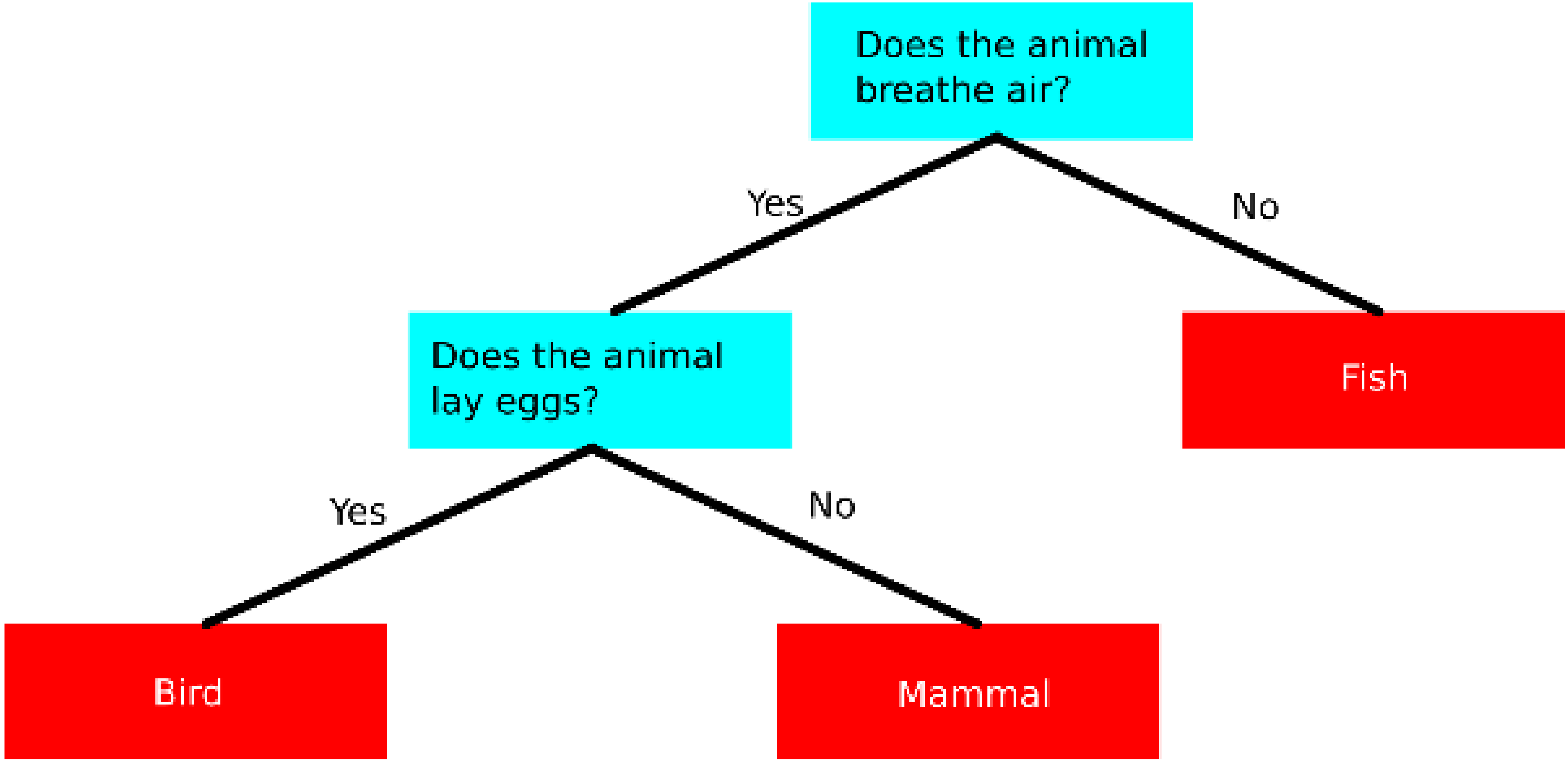
KNN

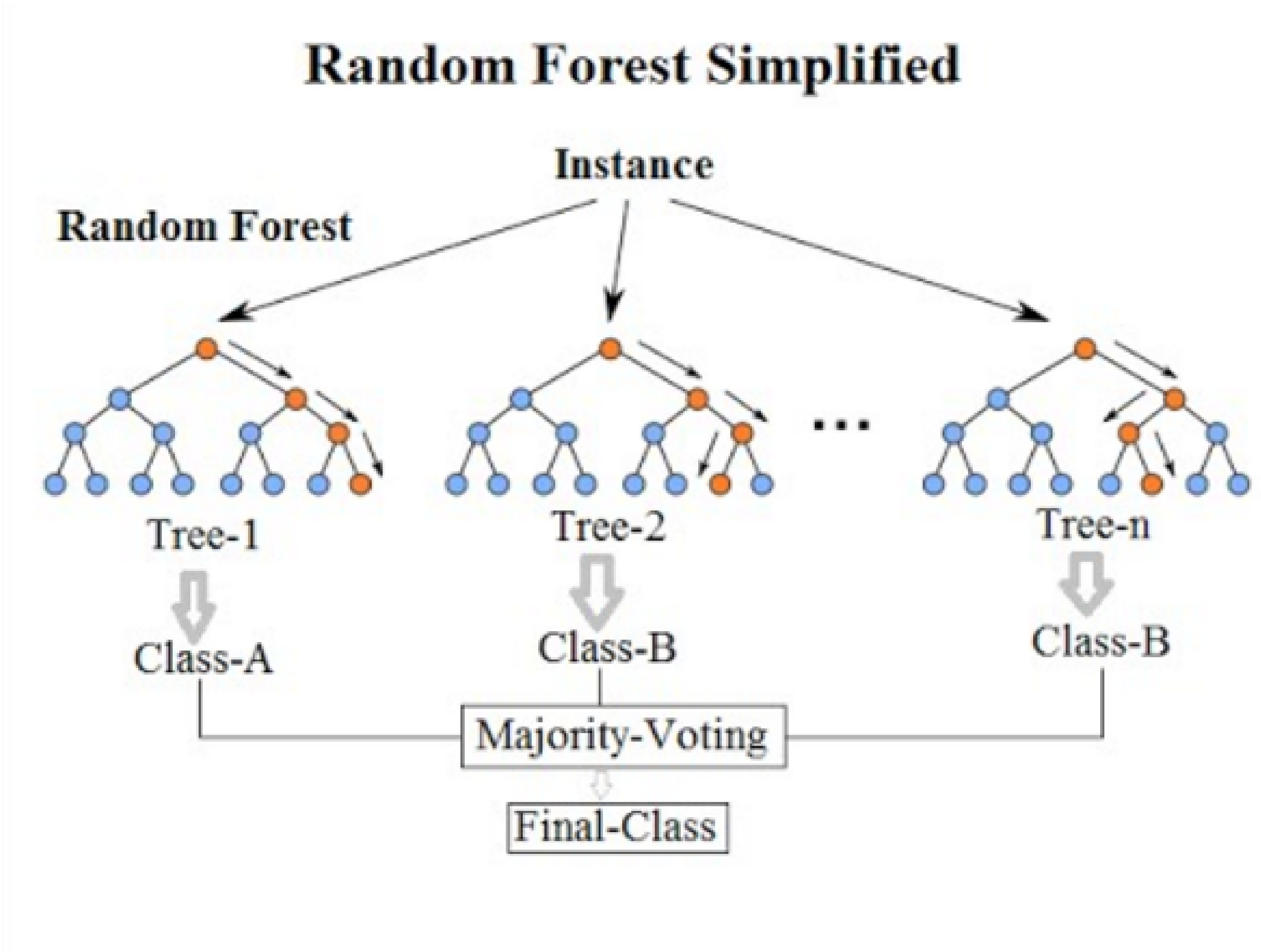


LogisticRegression



Decision Tree




Random Forest Simplified
Instance
Random Forest
...
Tree-1
Tree-2
Tree-n
Class-A
Class-B
Class-B
Majority-Voting
Final-Class

Naïve Bayes

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)\,P(c)}{P(x)}$$

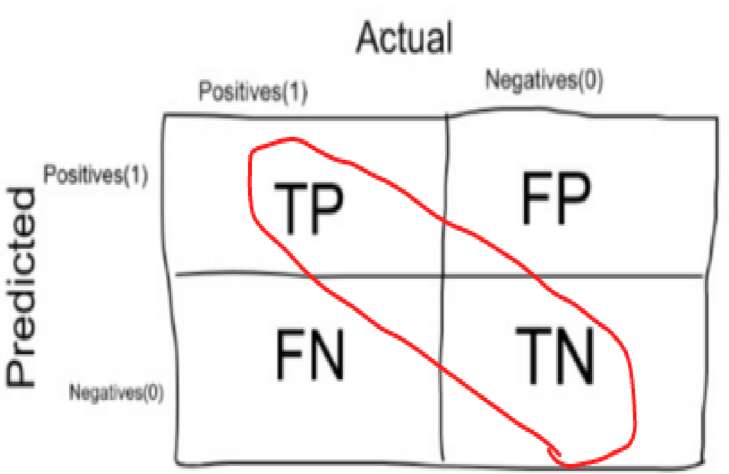
Posterior Probability

Predictor Prior Probability

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Performance Metrics

- ACCURACY
- F1_SCORE



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

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Results
Model Evaluation and Validation

Algorithms	BOW	TFIDF	W2V
naive Bayes	80(97)	88(100)	57(50)
KNN	82(90)	90(92)	69(74)
Logistic regression	90(100)	88(100)	57(85)
Decision Tree	83(85)	83(92)	71(73)
Random Forest	90(94)	88(95)	70(90)

Scope of Development

We may use some advanced machine learning techniques to improve the performance , like

Deep Learning

REQUIRMENTS

- OS: Windows,Mac OS,unix,linux, etc.
- RAM-1)Minimum(8gb)
- 2)Recommended(16gb)
- PROCESSOR: 1)Minimum(i5)
- 2)Recommended(i7)
- ROM:1)Minimum(512gb)
- 2)Recommended(1 TB)