Case Study 3

TEAM 12

MADE BY::

GUANXIONG LIU

JIANKUN BI

NAVEEN POTHAYATH

ABHISHEK EASWARAN

SUCHITHRA BALAKRISHNAN

Data

- 2000 samples are used.
  - 1000 Positive reviews
  - 1000 Negative review
- Split into training and testing data.
  - Training data-75%
  - Test data=25%

Sklearn Tutorial Problem

```
0 params - {'vect__ngram_range': (1, 1)}; mean - 0.82; std - 0.02
1 params - {'vect__ngram_range': (1, 2)}; mean - 0.83; std - 0.01
```

```
             precision    recall  f1-score   support

        neg       0.90      0.86      0.88       248
        pos       0.87      0.90      0.89       252

avg / total       0.88      0.88      0.88       500

[[213  35]
 [ 24 228]]
```

TFIDVECTORIZER

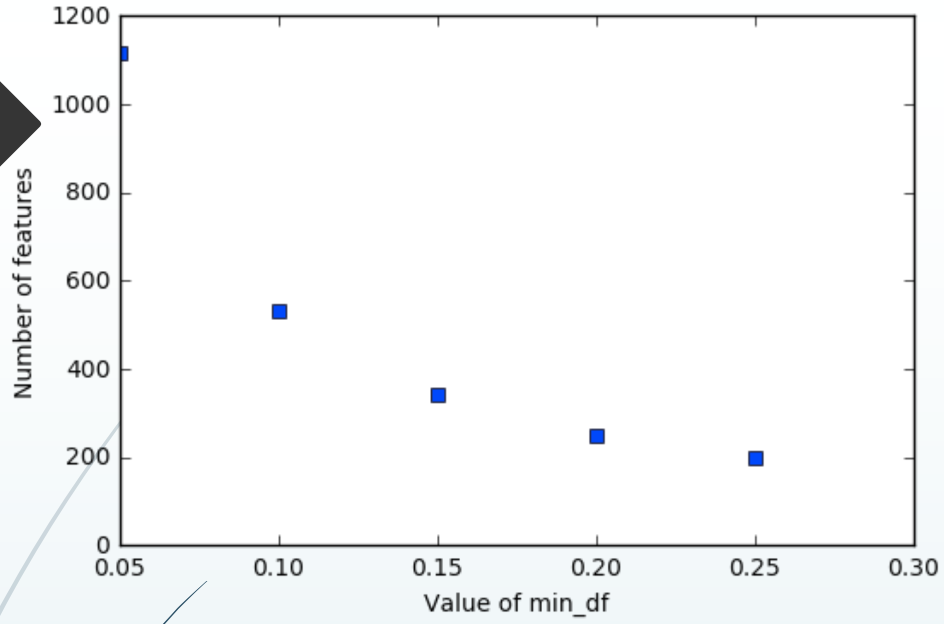
In information retrieval, tf–idf, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.[1]:8 It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

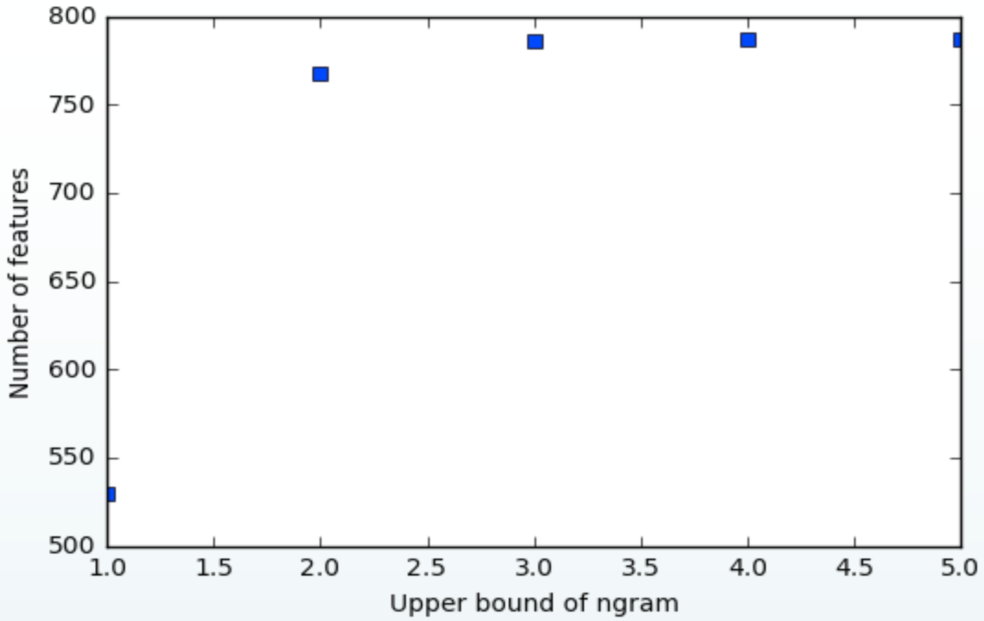
$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

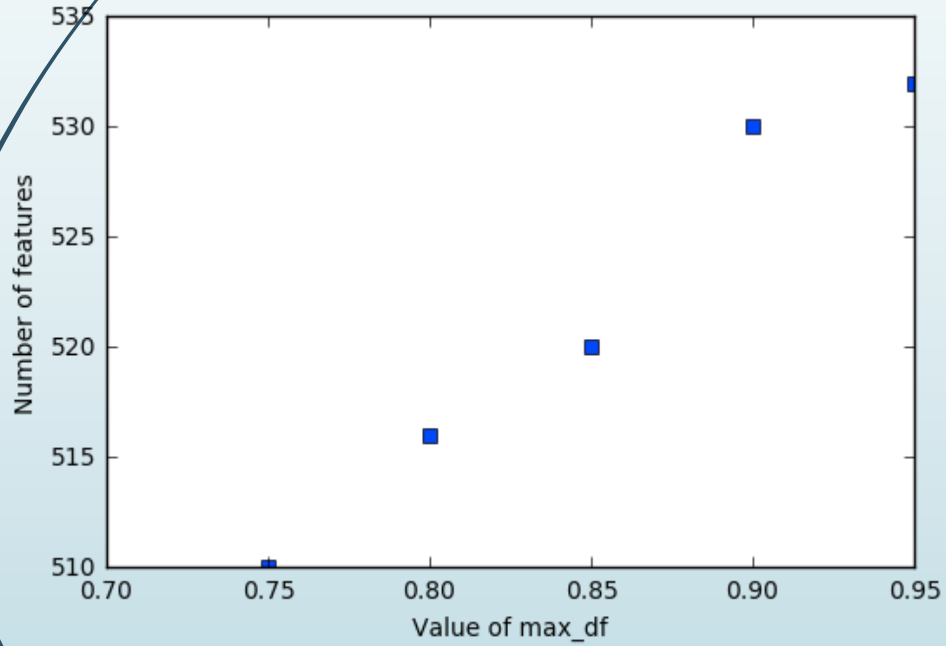
$tf_{i,j}$ = number of occurrences of i in j

df_i = number of documents containing i

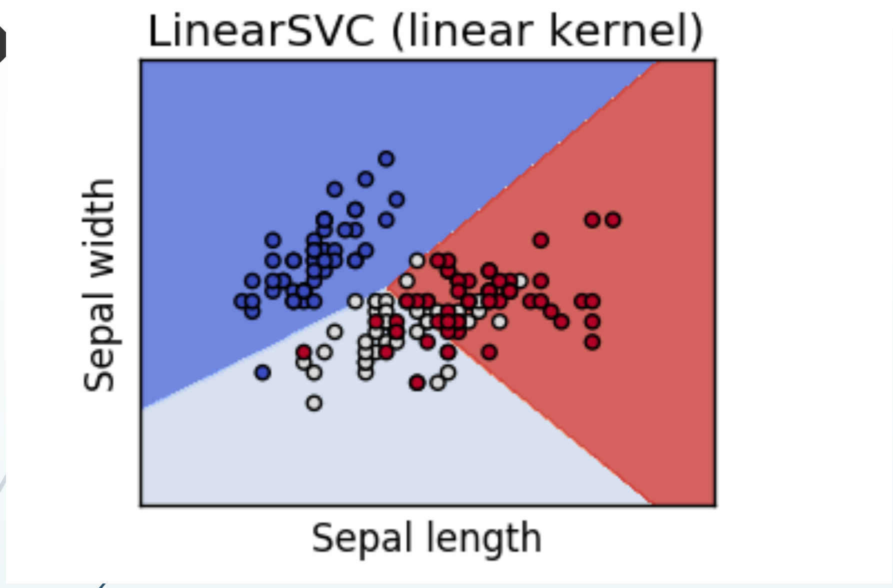
N = total number of documents





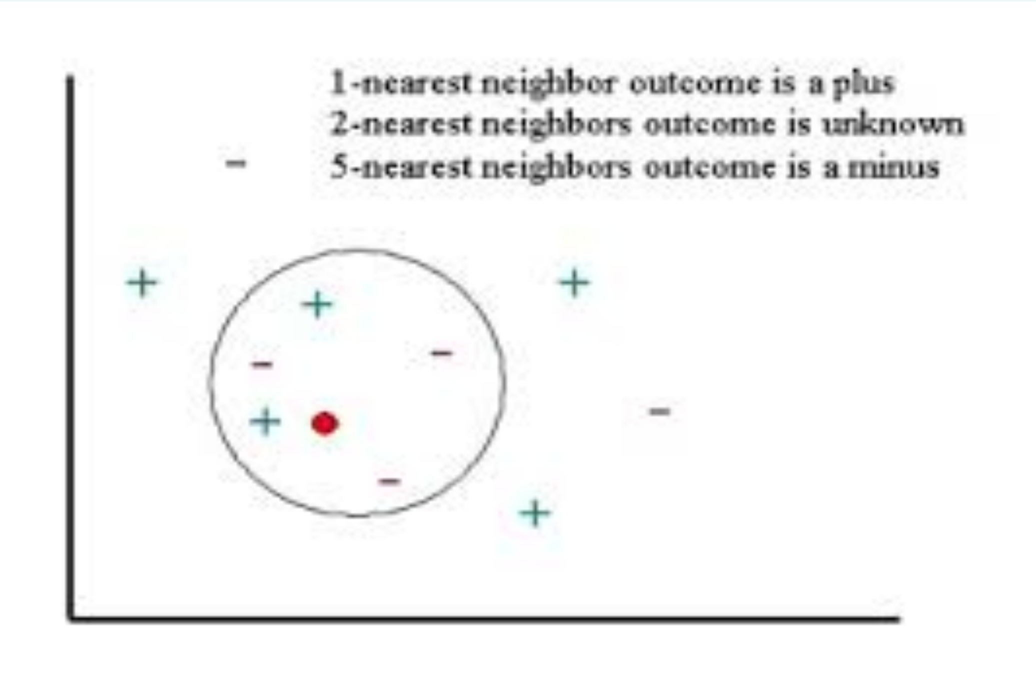


Impact of min_df, max_df and ngram upper bound on the number of extracted features



In [machine learning](), support vector machines (SVMs, also support vector networks[1]) are [supervised learning]() models with associated learning [algorithms]() that analyze data used for [classification]() and [regression analysis]().

In [pattern recognition](), the k-Nearest Neighbors algorithm (or k-NN for short) is a [non-parametric]() method used for [classification]() and [regression]().[1] In both cases, the input consists of the k closest training examples in the [feature space]().

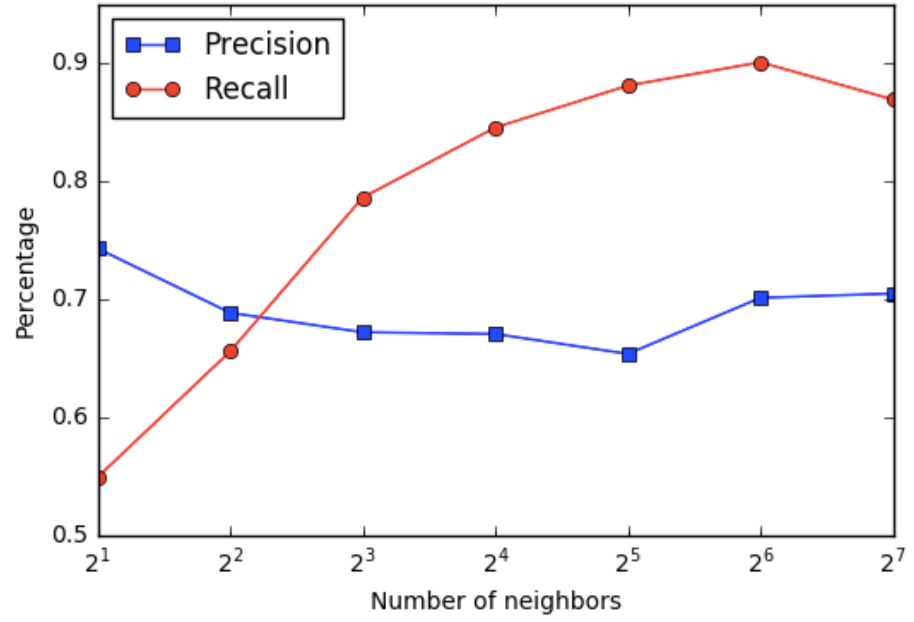


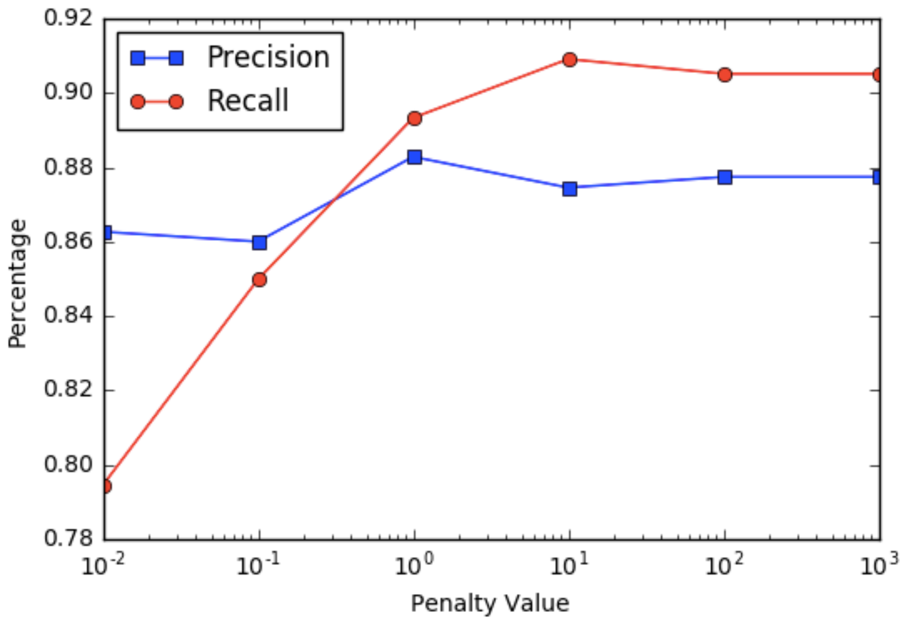
Evaluation Metrics

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

		Predicted condition	
	Total population	Predicted Condition positive	Predicted Condition negative
True condition	condition positive	**True positive**	**False Negative** (Type II error)
	condition negative	**False Positive** (Type I error)	**True negative**





	precision	recall	f1-score	support
neg	0.87	0.85	0.86	252
pos	0.85	0.87	0.86	248
avg / total	0.86	0.86	0.86	500

```
[[215  37]
 [ 33 215]]
```

KNN

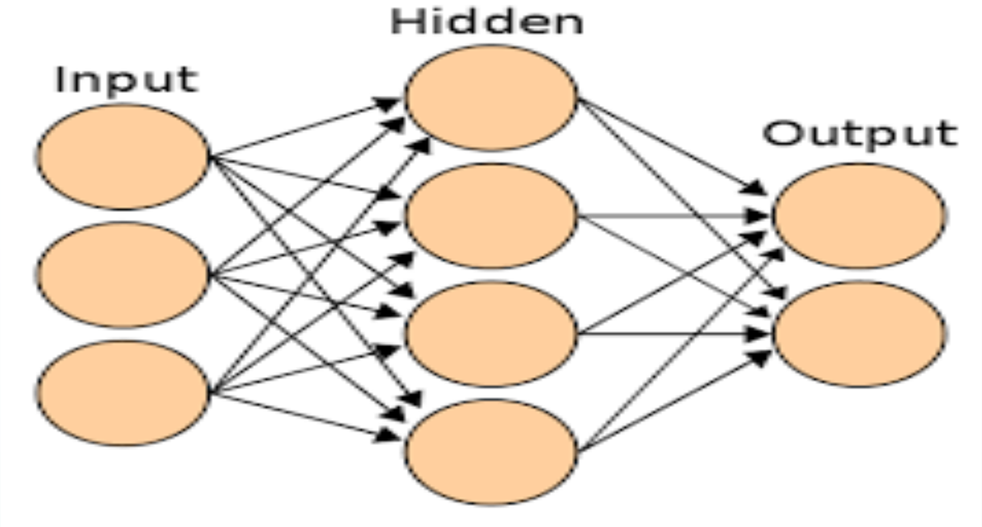
	precision	recall	f1-score	support
neg	0.61	0.81	0.70	247
pos	0.73	0.50	0.59	253
avg / total	0.67	0.65	0.65	500

```
[[201  46]
 [127 126]]
```

Linear SVC

	precision	recall	f1-score	support
neg	0.75	0.87	0.80	247
pos	0.85	0.71	0.77	253
avg / total	0.80	0.79	0.79	500

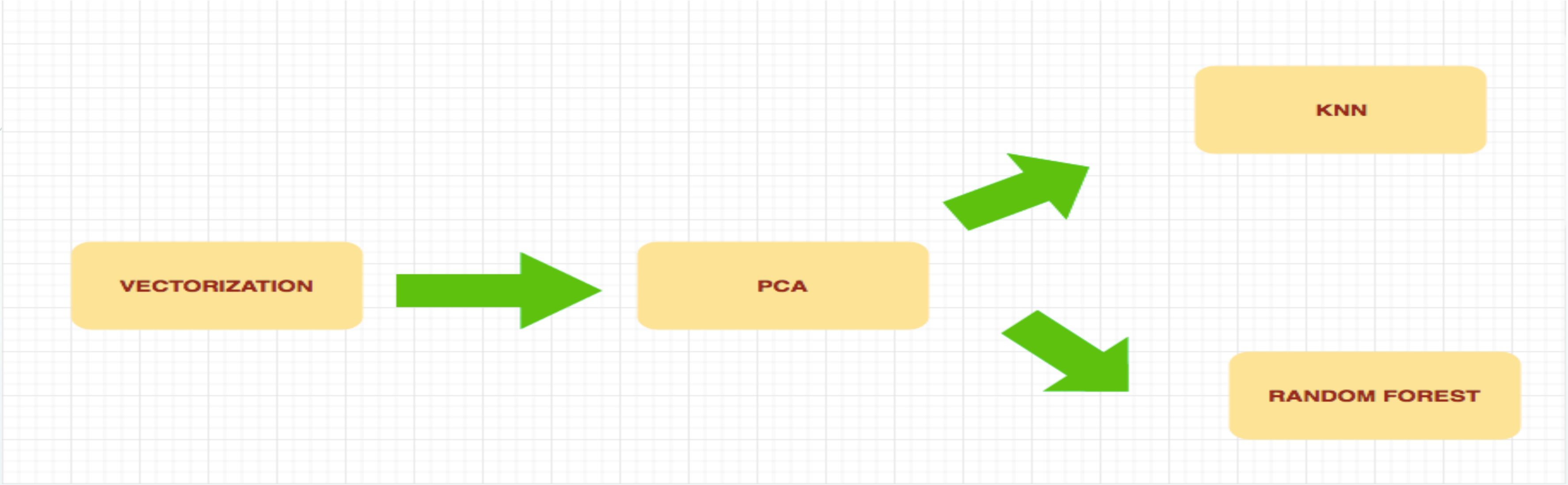
Random forests or random decision forests[1][2] are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.



	precision	recall	f1-score	support
neg	0.90	0.88	0.89	247
pos	0.89	0.91	0.90	253
avg / total	0.90	0.90	0.90	500

A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one.

SYSTEM DIAGRAM FOR RANDOM FOREST/KNN WITH PCA



CONFUSION MATRIX FOR KNN AFTER PCA

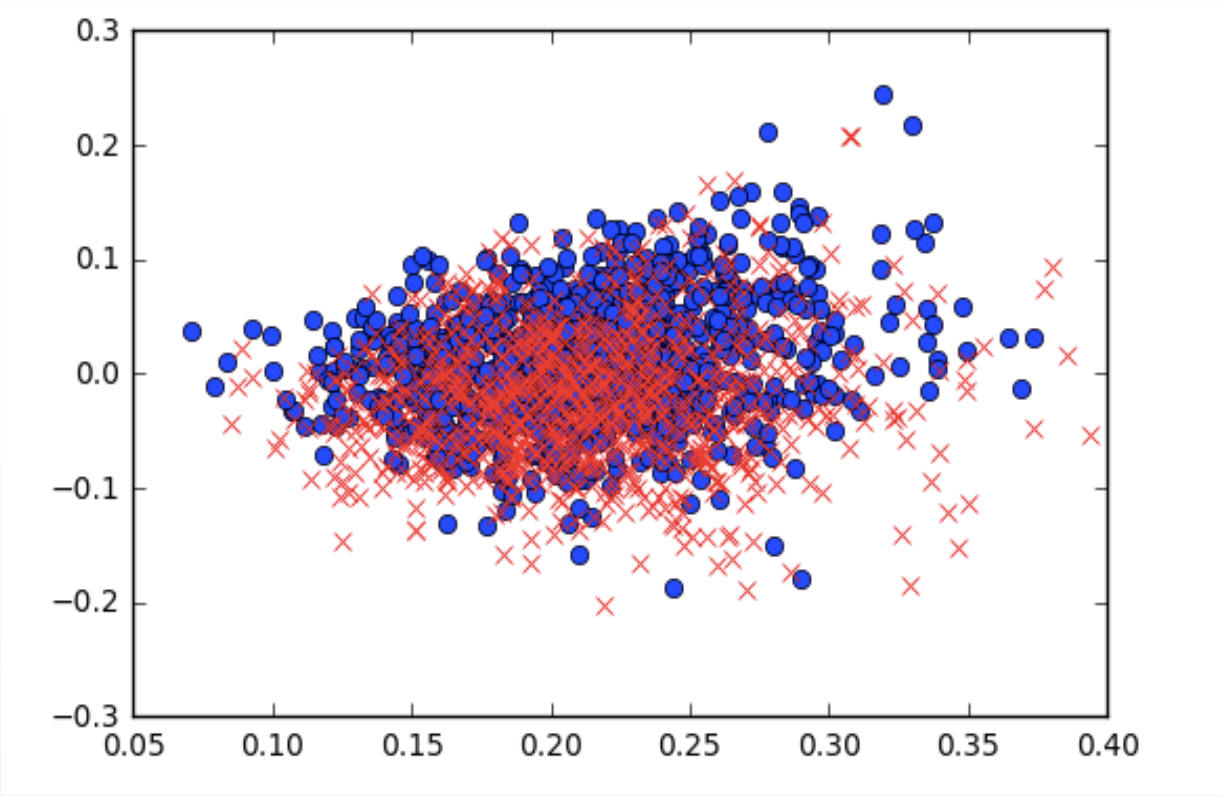
	precision	recall	f1-score	support
neg	0.61	0.81	0.70	247
pos	0.73	0.50	0.59	253
avg / total	0.67	0.65	0.65	500

```
[[201  46]
 [127 126]]
```

CONFUSION MATRIX FOR RANDOM FOREST AFTER PCA

	precision	recall	f1-score	support
neg	0.63	0.81	0.71	247
pos	0.74	0.55	0.63	253
avg / total	0.69	0.67	0.67	500

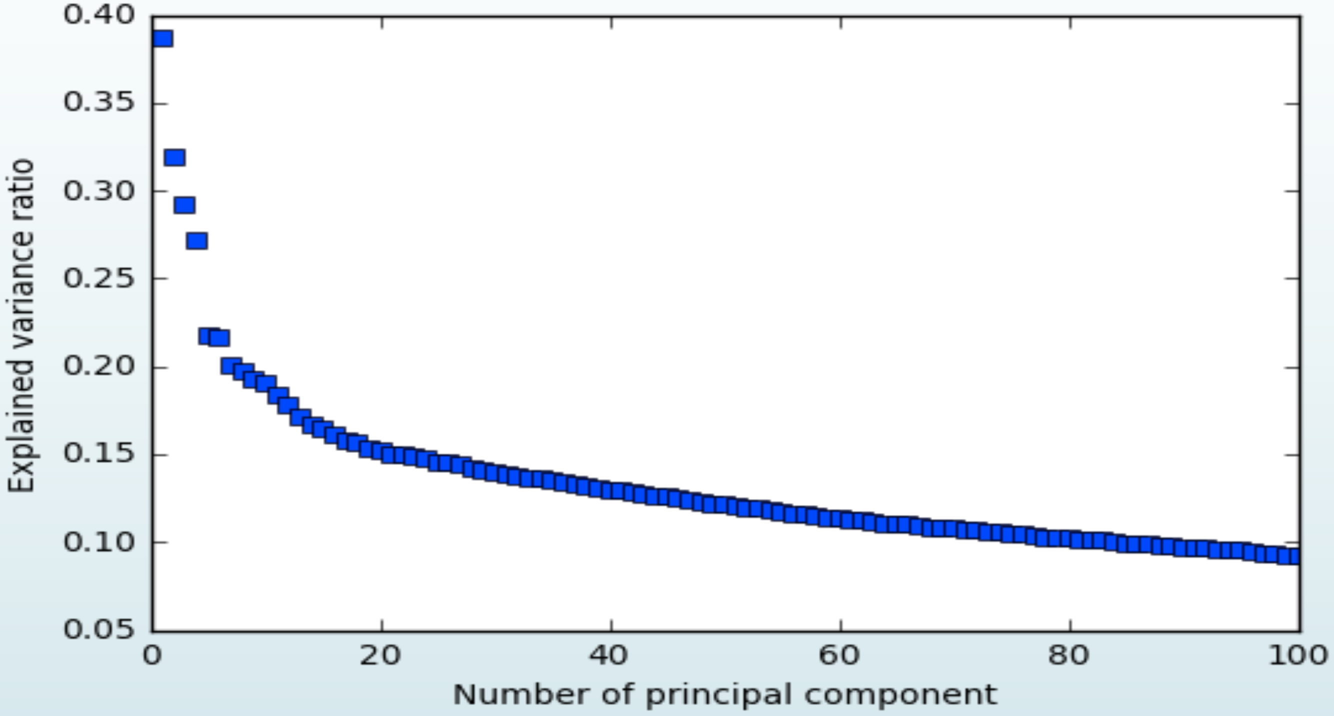
The Classification Problem



Latent semantic analysis (**LSA**) is a technique in natural language processing, in particular distributional semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms.

Visualization of top 100 principal components

- Too many features
- Less correlation between features
- Less information contained in top principal components



PMI (Pointwise Mutual Information)

$$PMI(w1, w2) = \frac{P(w1a \wedge w2)}{P(w1)P(w2)}$$

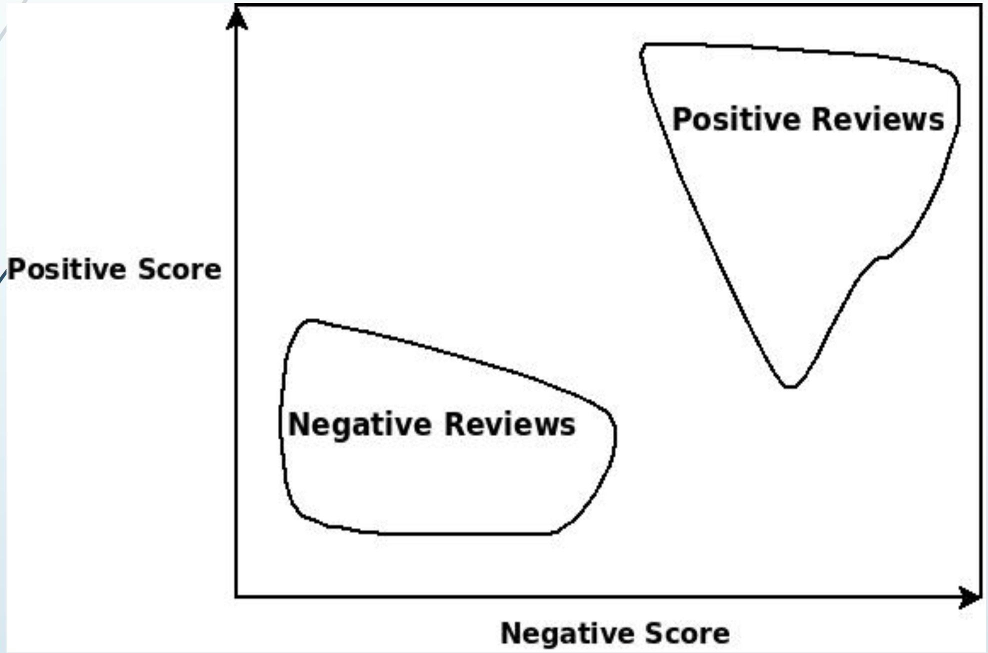
How “close” two words could be

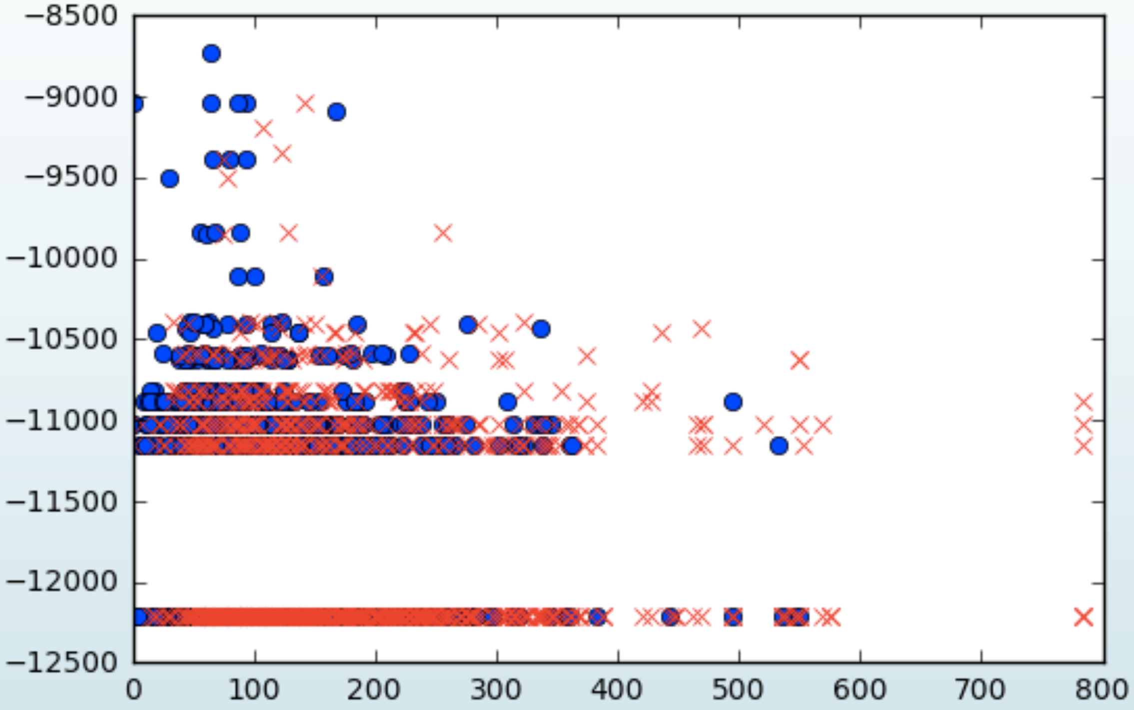
Semantic Orientation

$$\boldsymbol{SO(w) = \sum_{w' \in P^+} PMI(w, w') - \sum_{w' \in P^-} PMI(w, w')}$$

What’s the “Orientation” of each word

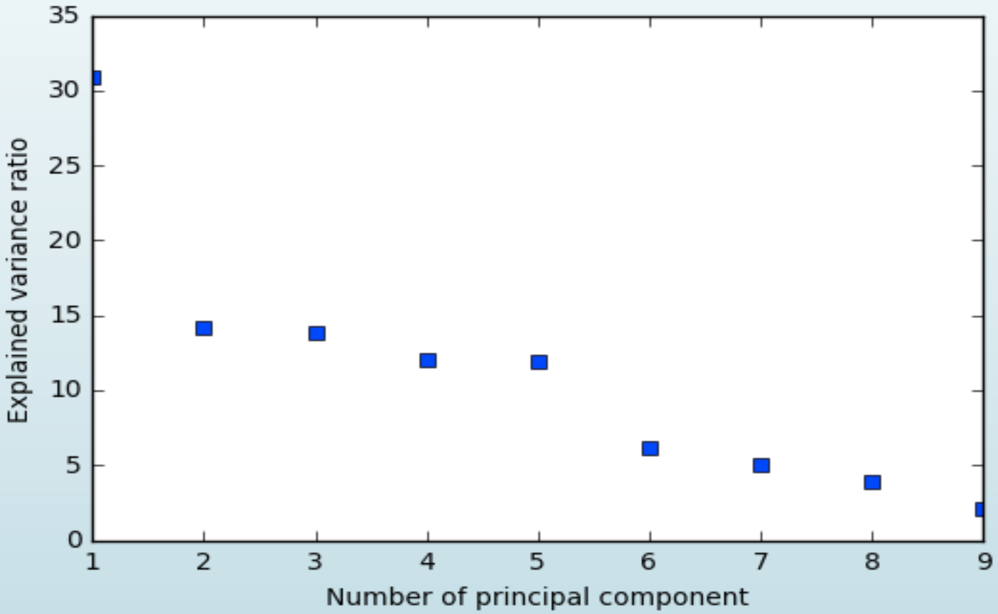
Semantic Orientation Analysis





SYSTEM DIAGRAM FOR COMBINING FEATURE SELECTION AND PCA

Visualization of principal components



Final visualization