## 1. Loading the NIPS papers

```
# Importing modules
import pandas as pd

# Read datasets/papers.csv into papers
papers = pd.read_csv('datasets/papers.csv')

# Print out the first rows of papers
print(papers.head())
```

```
     id  year                                              title
event_type  \
0     1  1987  Self-Organization of Associative Database and ...
NaN
1    10  1987  A Mean Field Theory of Layer IV of Visual Cort...
NaN
2   100  1988  Storing Covariance by the Associative Long-Ter...
NaN
3  1000  1994  Bayesian Query Construction for Neural Network...
NaN
4  1001  1994  Neural Network Ensembles, Cross Validation, an...
NaN

                                            pdf_name          abstract
\
0  1-self-organization-of-associative-database-an...  Abstract Missing

1  10-a-mean-field-theory-of-layer-iv-of-visual-c...  Abstract Missing

2  100-storing-covariance-by-the-associative-long...  Abstract Missing

3  1000-bayesian-query-construction-for-neural-ne...  Abstract Missing

4  1001-neural-network-ensembles-cross-validation...  Abstract Missing


                                          paper_text
0  767\n\nSELF-ORGANIZATION OF ASSOCIATIVE DATABA...
1  683\n\nA MEAN FIELD THEORY OF LAYER IV OF VISU...
2  394\n\nSTORING COVARIANCE BY THE ASSOCIATIVE\n...
3  Bayesian Query Construction for Neural\nNetwor...
4  Neural Network Ensembles, Cross\nValidation, a...
```

## 2. Preparing the data for analysis

```
# Remove the columns
columns=['id', 'event_type', 'pdf_name']
papers.drop(columns, axis=1, inplace=True)
```

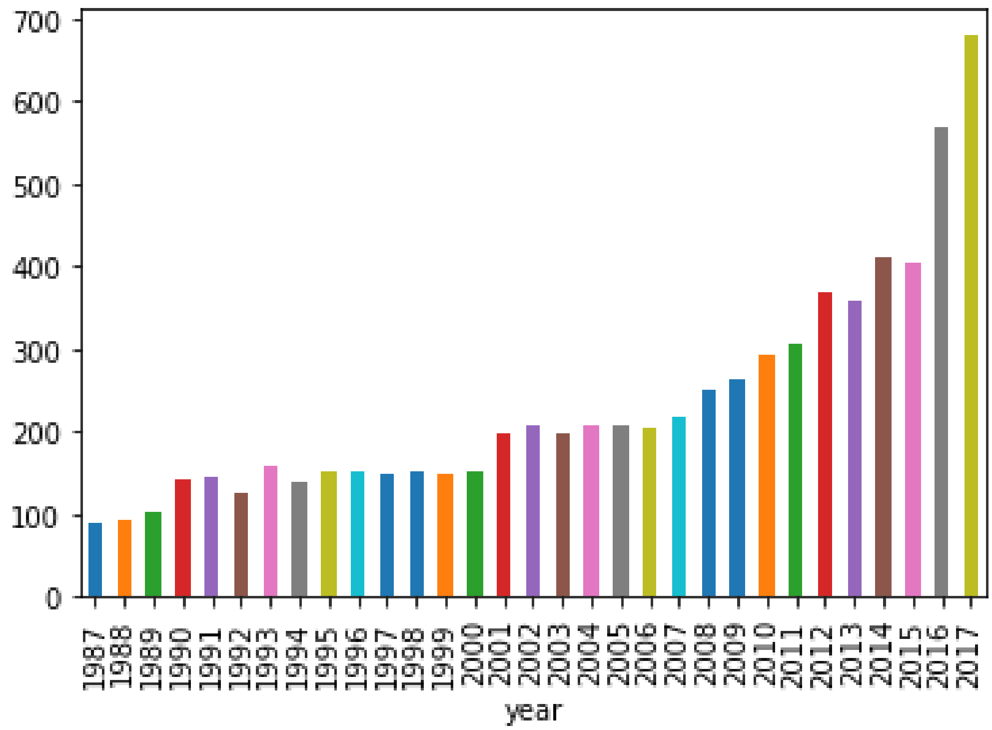
```
# Print out the first rows of papers
print(papers.head())
```

```
   year                                              title
abstract  \
0  1987  Self-Organization of Associative Database and ...  Abstract
Missing
1  1987  A Mean Field Theory of Layer IV of Visual Cort...  Abstract
Missing
2  1988  Storing Covariance by the Associative Long-Ter...  Abstract
Missing
3  1994  Bayesian Query Construction for Neural Network...  Abstract
Missing
4  1994  Neural Network Ensembles, Cross Validation, an...  Abstract
Missing

                                          paper_text
0  767\n\nSELF-ORGANIZATION OF ASSOCIATIVE DATABA...
1  683\n\nA MEAN FIELD THEORY OF LAYER IV OF VISU...
2  394\n\nSTORING COVARIANCE BY THE ASSOCIATIVE\n...
3  Bayesian Query Construction for Neural\nNetwor...
4  Neural Network Ensembles, Cross\nValidation, a...
```

## 3. Plotting how machine learning has evolved over time

```
# Group the papers by year
groups = papers.groupby('year')

# Determine the size of each group
counts = groups.size()
#print(counts)

# Visualise the counts as a bar plot
import matplotlib.pyplot
%matplotlib inline
#counts.plot(kind='line')
counts.plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f55fede76a0>
```




### 4. Preprocessing the text data

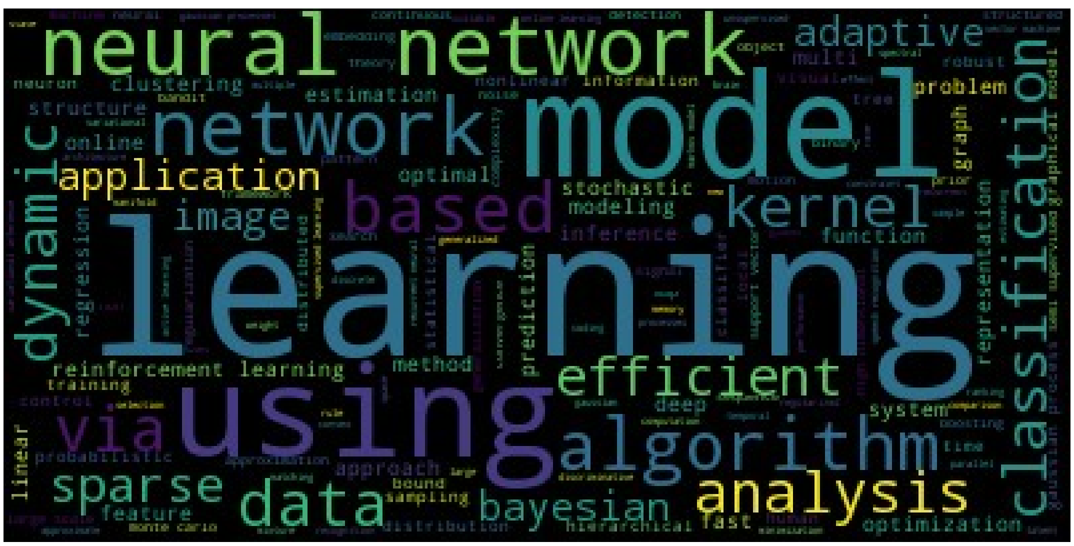
```
# Load the regular expression library
import re

# Print the titles of the first rows
print(papers['title'].head())

# Remove punctuation
papers['title_processed'] = papers['title'].map(lambda x:
re.sub('[,\.!?]', '', x))

# Convert the titles to lowercase
papers['title_processed'] = papers['title_processed'].map(lambda x:
x.lower())

# Print the processed titles of the first rows
print(papers['title_processed'].head())
```

```
0    Self-Organization of Associative Database and ...
1    A Mean Field Theory of Layer IV of Visual Cort...
2    Storing Covariance by the Associative Long-Ter...
3    Bayesian Query Construction for Neural Network...
4    Neural Network Ensembles, Cross Validation, an...
Name: title, dtype: object
0    self-organization of associative database and ...
1    a mean field theory of layer iv of visual cort...
2    storing covariance by the associative long-ter...
```

```
3    bayesian query construction for neural network...
4    neural network ensembles cross validation and ...
Name: title_processed, dtype: object
```

## 5. A word cloud to visualize the preprocessed text data

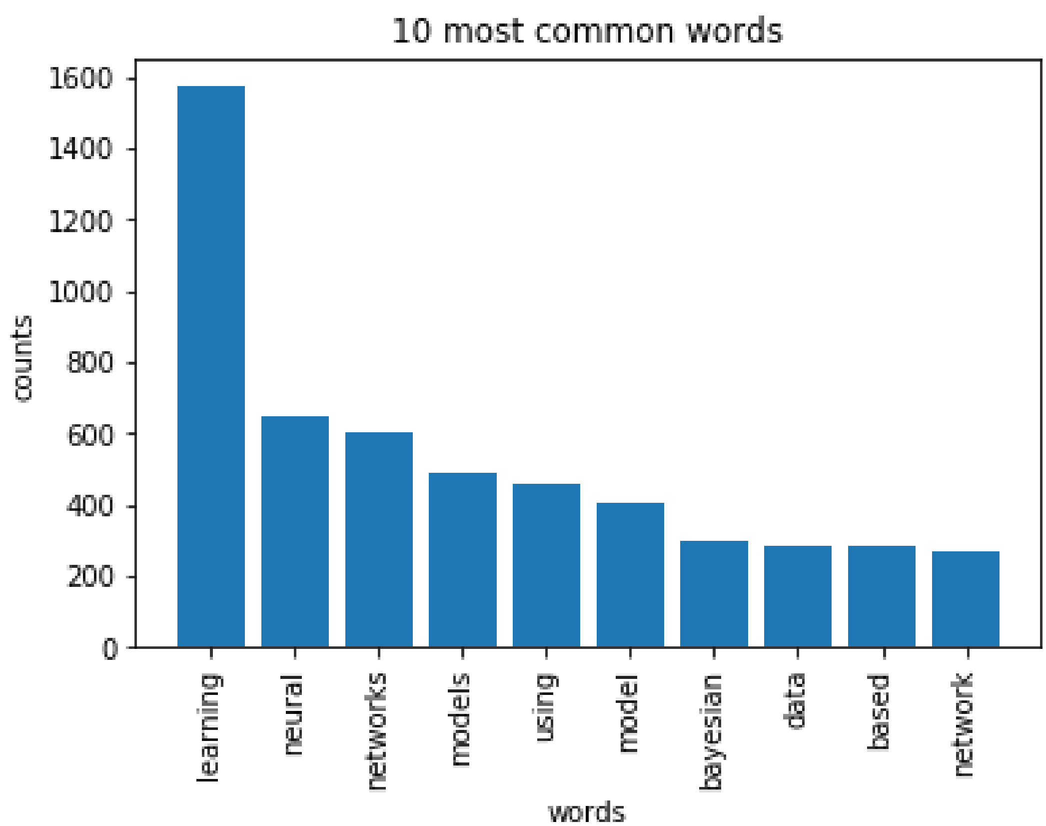
```python
# Import the wordcloud library
from wordcloud import WordCloud, STOPWORDS

# Join the different processed titles together.
long_string = ' '
long_string = long_string.join(papers['title_processed'])
#print(long_string)

# Create a WordCloud object
wc=WordCloud()

# Generate a word cloud
wc.generate(long_string)

# Visualize the word cloud
wc.to_image()
```




## 6. Prepare the text for LDA analysis

```python
# Load the library with the CountVectorizer method
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np

# Helper function
def plot_10_most_common_words(count_data, count_vectorizer):
    import matplotlib.pyplot as plt
    words = count_vectorizer.get_feature_names()
    total_counts = np.zeros(len(words))
    for t in count_data:
```

```
        total_counts+=t.toarray()[0]

    count_dict = (zip(words, total_counts))
    count_dict = sorted(count_dict, key=lambda x:x[1], reverse=True)
[0:10]
    words = [w[0] for w in count_dict]
    counts = [w[1] for w in count_dict]
    x_pos = np.arange(len(words))

    plt.bar(x_pos, counts,align='center')
    plt.xticks(x_pos, words, rotation=90)
    plt.xlabel('words')
    plt.ylabel('counts')
    plt.title('10 most common words')
    plt.show()

# Initialise the count vectorizer with the English stop words
count_vectorizer = CountVectorizer(stop_words='english')

# Fit and transform the processed titles
count_data = count_vectorizer.fit_transform(papers['title_processed'])

# Visualise the 10 most common words
plot_10_most_common_words(count_data, count_vectorizer)
```



## 7. Analysing trends with LDA

```
import warnings
warnings.simplefilter("ignore", DeprecationWarning)

# Load the LDA model from sk-learn
from sklearn.decomposition import LatentDirichletAllocation as LDA

# Helper function
def print_topics(model, count_vectorizer, n_top_words):
    words = count_vectorizer.get_feature_names()
    for topic_idx, topic in enumerate(model.components_):
        print("\nTopic #%d:" % topic_idx)
        print(" ".join([words[i]
                        for i in topic.argsort()[:-n_top_words - 1:-
1]]))

# Tweak the two parameters below (use int values below 15)
number_topics = 14
number_words = 14

# Create and fit the LDA model
lda = LDA(n_components=number_topics)
lda.fit(count_data)

# Print the topics found by the LDA model
print("Topics found via LDA:")
print_topics(lda, count_vectorizer, number_words)
```

```
Topics found via LDA:

Topic #0:
learning gradient algorithm robust networks neural matrix structure
continuous approximation descent order analog space

Topic #1:
model probabilistic decision machine generative adaptive self boosting
learning mixture using trees discriminative decomposition

Topic #2:
bayesian learning estimation fast recognition methods models sampling
multiple based search object unsupervised using

Topic #3:
gaussian regression non large processes process functions hierarchical
scale matching support vector inference motion

Topic #4:
learning classification prediction supervised high method
representations graph semi propagation applications data using time
```

```
Topic #5:
learning structured random variational visual complexity dynamic
theory sample tree map networks segmentation field

Topic #6:
neural networks learning network models information online linear
algorithms recurrent time latent graphical active

Topic #7:
image convolutional point computational action architecture noisy
simple cortical localization synthesis cells compression development

Topic #8:
sparse convex spectral human spike brain coding memory adversarial
reduction exponential using associative performance

Topic #9:
models markov regularization hidden learning mixtures function
connectionist dynamical factorization parameter distance exploration
constrained

Topic #10:
learning clustering reinforcement approach control statistical
modeling model data based systems embedding sequence recovery

Topic #11:
analysis multi inference stochastic learning efficient optimal
optimization kernel selection local approximate natural linear

Topic #12:
deep training bounds rank bandits submodular binary margin dimensional
scalable filtering tensor algorithm data

Topic #13:
detection feature using kernels risk fields estimating bayes design
complex data plasticity structural correlation
```

## 8. The future of machine learning

```
# The historical data indicates that:
more_papers_published_in_2018 = True
```