Aerofiler

Seminar

# Coding for Lawyers From Basics to AI

22 October 2019

## Setup

- **Setup** - https://jupyter.org/try

## Signpost - What Will We Be Building?

- Clause predictor
- Clause classifier

Aerofiler

## Getting started

## Variables - Used to Store Data in Memory

- print("Hello world")
- court = "High Court"
- court[0:4]
- court = "Supreme Court"
- court[0:4]

## Getting started

## Conditional Execution

```
court = "High Court"
judge = ""
if court == "High Court":
    judge = "Edelman J"
elif court == "Supreme Court":
    judge = "Bathurst J"
else:
    judge = "Unknown"
print(judge)
```

## Getting started

## Function

Define function: (whitespace is important)

```
def get_judge(court):
        judge = ""
        if court == "High Court":
            judge = "Edelman J"
        elif court == "Supreme Court":
            judge = "Bathurst J"
        else:
            judge = "Unknown"
        return judge
```

Use it:

```
        get_judge("High Court")
```

## Getting started

## Arrays

- court = "Supreme Court of New South Wales"
- word_list = court.split()
- word_list[2:4]
- word_list[2:]

## Getting started

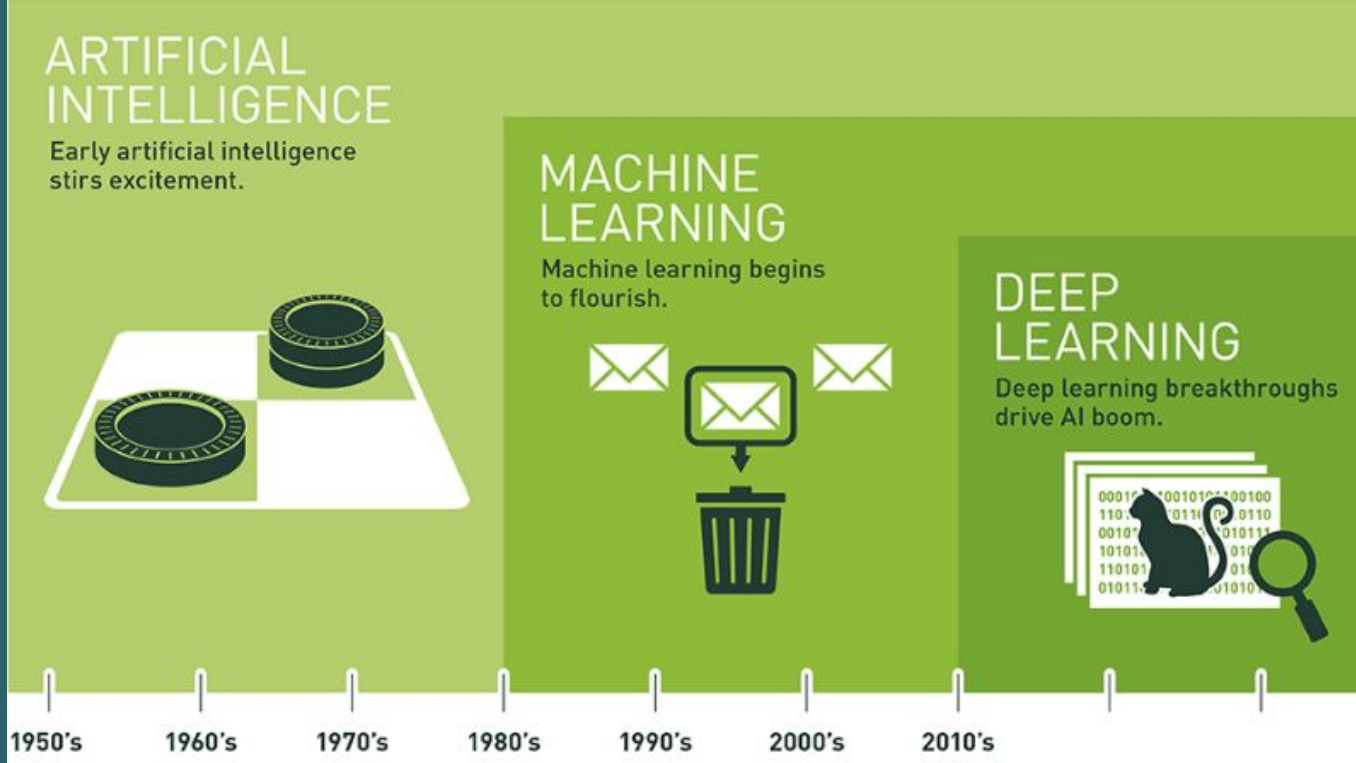
## Loops

- court = "New South Wales Supreme Court"
- word_list = "New South Wales Supreme Court".split()
- for word in word_list:
  print(word)

## Practical Application

- from requests import get
- legislation = get('https://raw.githubusercontent.com/deltanovember/ml/master/ca2001172.txt')
- legislation.content

# AI Primer




Aerofiler

## Artificial Intelligence

- **Artificial intelligence** - any technique that allows a computer to mimic human intelligence

Aerofiler

## Artificial Intelligence

- **Artificial intelligence** - any technique that allows a computer to mimic human intelligence

- Consider the following set of rules:

    1. Cycle through all contracts
    2. Find all contracts that have the exact phrase “change of control” or “change in control”
    3. If there is a match, flag the contract
    4. Return all flagged contracts

## Machine Learning

- **Machine learning** - a system that learns from experience

Aerofiler

## Machine Learning

- **Machine learning** - a system that learns from experience

| Clause | Text |
| --- | --- |
| Change of Control | ▸ In the event of a Change of Control (as defined...) |
| Change of Control | ▸ If there is a Change of Control in the Company ... |
| Change of Control | ▸ In the event that the Company undergoes a ... |
| Governing Law | ▸ This Agreement shall be governed according to ... |
| Governing Law | ▸ This agreement shall be governed by ... |

## Clause predictor

## Setup

- Download: https://bit.ly/32uxATU

```
pip install pandas
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC
from sklearn.pipeline import Pipeline
from sklearn.metrics import
confusion_matrix,classification_report
```

## Clause predictor

## Load Data

- data_frame = pd.read_csv('./clauses.csv')
- data_frame['Clause'].value_counts()
- X = data_frame['Text']
- y = data_frame['Clause']

## Clause predictor

## Training

- X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.33)
- text_clf = Pipeline([('tfidf',TfidfVectorizer()),('clf',LinearSVC())])
- text_clf.fit(X_train, y_train)

## Clause predictor

## Testing

- predictions = text_clf.predict(X_test)
- print(classification_report(y_test, predictions))
- text_clf.predict(["This agreement may be terminated at will be either party"])

## Clause classifier

## Setup

- Download: https://bit.ly/2N9oaqD

  pip install pandas

  ```
  import pandas as pd
  from sklearn.feature_extraction.text import CountVectorizer
  from sklearn.decomposition import
  LatentDirichletAllocation
  ```

## Clause classifier

## Load Data

- npr = pd.read_csv('raw_clauses.csv')
- npr.head()

Aerofiler

## Clause classifier

## Training

- cv = CountVectorizer(max_df=0.9,min_df=2,stop_words='english')
- data_matrix = cv.fit_transform(npr['Text'])
- LDA = LatentDirichletAllocation(n_components=60,random_state=42)
- LDA.fit(data_matrix)

## Clause classifier

## Results

- topic_results = LDA.transform(data_matrix)
- npr['Topic'] = topic_results.argmax(axis=1)
- npr

```
counter = 0
for topic_number in npr['Topic']:
    if 23 == topic_number:
        print("- " + npr['Text'][counter])
    counter = counter + 1
```

# Q&A

- Questions

Aerofiler