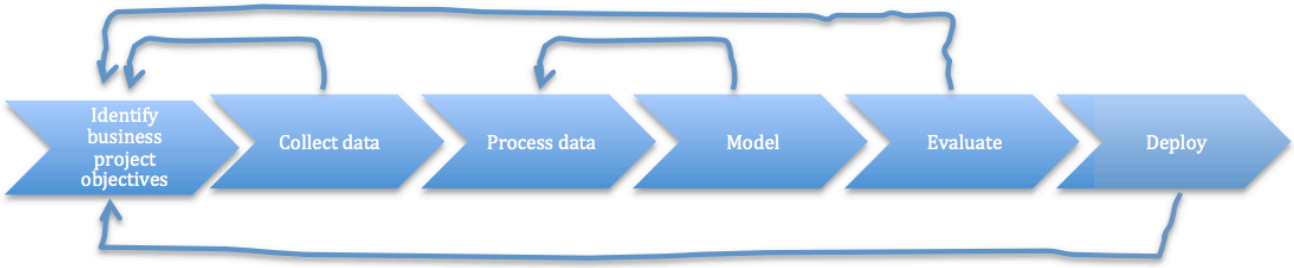
Machine Learning with Python



Select Data

Pandas is your friend.

Get your data into a Pandas Data Frame.

```
import pandas as pd
  # Start by importing this Python module
```

Load a DataFrame from a CSV file

```
X = pd.read_csv('file.csv') # simplest
X = pd.read_csv('file.csv', header = None) # no header
X = pd.read_csv('file.csv', skiprows = 5)  # skip first 5 rows
          # add column names:
X = pd.read_csv('file.csv', names=['a', 'b', 'c'])
          # use a column as Index:
X = pd.read_csv('file.csv', index_col='id_column')
          # additional expressions to recognise as NaN:
X = pd.read_csv('file.csv', na_values='?')
          # different separator than commas (tab):
X = pd.read_csv('file.csv',  sep='\t')
          # semicolon as separator:
X = pd.read_csv('file.csv',  sep=';')
```

Load a DataFrame from a HTML table

```
X = pd.read_html(theURL, header=1)[0]
```

Get DataFrame information

```
X.shape         # nr. Rows and columns
X.columns       # list of column names
X.index         # extract of index values (rows)
X.info()        # index & data types
X.head(10)      # get first 10 rows. Default=5
X.tail(7)       # get last 7 rows. Default=5
```

```
X.describe()   # summary stats
```

Process Data - Clean

Rename the column names

```
X.rename(columns={'old.1':'new-1', 'old.2':'new-2'},
          inplace=True)
```

Extract output values

```
y = X.y_column.copy() # copy “y” column values out
y = X[X.columns[0]].copy() # copy from first column out

X.drop(['y_name'], axis=1, inplace=True) # then, drop y column
X.drop(X.columns[0], axis=1, inplace=True)
```

Map different values

Category to number

```
y = y.map({'class1':0, 'class2':1, 'class3':2})
X.ColumnName = X.ColumnName.map({'Yes':1, 'No':0})
```

Categorical to dummies (create extra columns)

```
X = pd.get_dummies(X, columns=['cat1', 'cat2', 'cat3'])
```

Date and time

```
X.date_col = pd.to_datetime(X.date_col, errors='coerce')
X.time_col = pd.to_timedelta(X.time_col, errors='coerce')
```

NaN values

Find NaN

```
X.isnull().values.any() # return True | False if NaNs present or not
X.isnull().any(axis=1)  # all rows with NaN
X[pd.isnull(X).any(axis=1)]   to print all rows with NaNs
```

Replace NaN

```
X.fillna(scalar_n, inplace=True)  # replace NaN with constant value
X.fillna(X.mean(), inplace=True)  # replace NaN with mean values
```

Remove NaN

```
X.dropna(axis=0, inplace=True) # Drop any row that has NaNs within it
  # Drop any row that has at least 4 NON-NaNs within it:
X.dropna(axis=0, thresh = 4, inplace=True)
X.dropna(axis=1, inplace=True) # Drop any column that has NaNs
```

Process Data - Normalising

Work in Progress

Split data set into train and test

Import this module

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

Split data

```
  # test size must be between 0 and 1;
  # use random_state for reproducibility
X_train, X_test, y_train, y_test =
                 train_test_split(X, y, test_size=0.2, random_state=7)
```

Modeling

Work in Progress

Evaluation

Work in Progress

Extra: Generic Python Processing

Loops

```
for x in range(0, 3):
     # do something

for x in range(11):
     # do something
```