Data Analysis and Machine Learning: Nearest Neighbors and Decision Trees

 $\hbox{Morten Hjorth-Jensen}^{1,2}$

Department of Physics, University of Oslo¹

Department of Physics and Astronomy and National Superconducting Cyclotron Laboratory, Michigan State University 2

May 30, 2018

© 1999-2018, Morten Hjorth-Jensen. Released under CC Attribution-NonCommercial 4.0 license

```
Decision trees, overarching aims
```

```
Nearest Neighbors

import mglearn
import numpy as np
from sklearn import linear_model
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures
from sklearn.preprocessing import PolynomialFeatures
from sklearn.neighbors import KNeighborsClassifier

# Generate sample data
X = np.sort(5-np.random.rand(40,1), axis=0)
y = X+03
y=y.ravel()

# Add noise to targets
X[::4] +=(0.5 - np.random.rand(1))
y[::5] +=(0.5 - np.random.rand(8))
a=np.array(X)
b=np.array(X)
b=np.array(y)

X_train=a[::9]
X_test=a[:9:1
y_train=b[::19]
y_test=b[:19:1]
y_test=b[::19]
y_test=b[::1]
model=mpodel_mit(('poly', PolynomialFeatures(degree=3)),('linear', Linemodel=model_mit(X_train_v_train)
```

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression

steps=250

distance=0
x=0
distance_list=[]
steps_list=[]
while x<steps:
    distance=imp.random.randint(-1,2)
    distance_list_append(distance)
    x+=1
    steps_list.append(x)
plt.plot(steps_list_distance_list, color='green', label="Random Walk D"

steps_list=np.asarray(steps_list)
distance_list=np.asarray(distance_list)

X=steps_list[:,np.newaxis]
#Polynomial fits
#Degree 2
poly_features=PolynomialFeatures(degree=2, include_bias=False)
Y_nolw=noly_features_fit_transform(X)
```