Python and Machine Learning 101: the road more aspired to be traveled but lesser understood

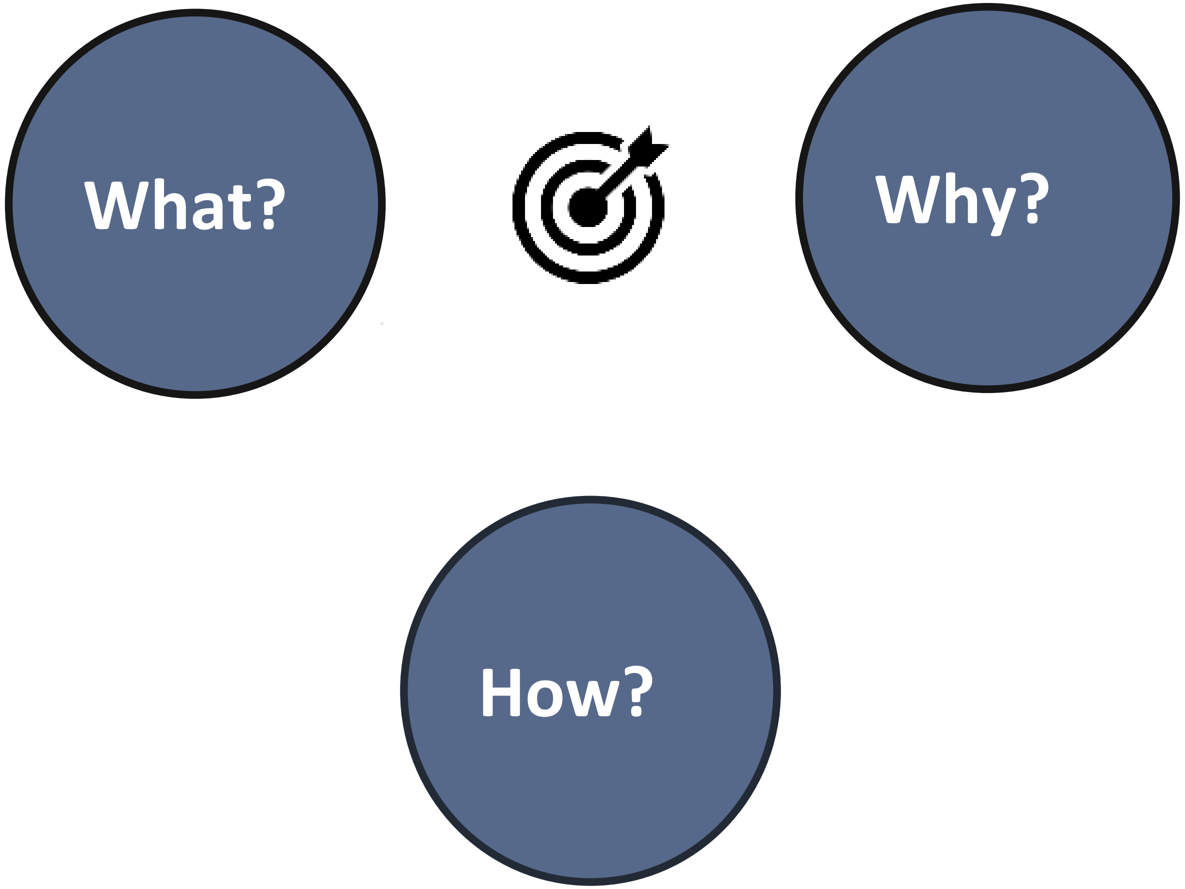
Maria Shoaib, Asst. Product Dev Manager, Ebryx, 2017- Present
Master's in Computer Science,
2014-2016, Rochester Institute of Technology, New York

Overview

- What? Why? How?
- Concepts of machine learning: 1) classification, 2) regression, 3) training, 4) testing, and 5) validation
- Machine learning project in Python
- Resources

Fundamental Concepts

The ‘learning’

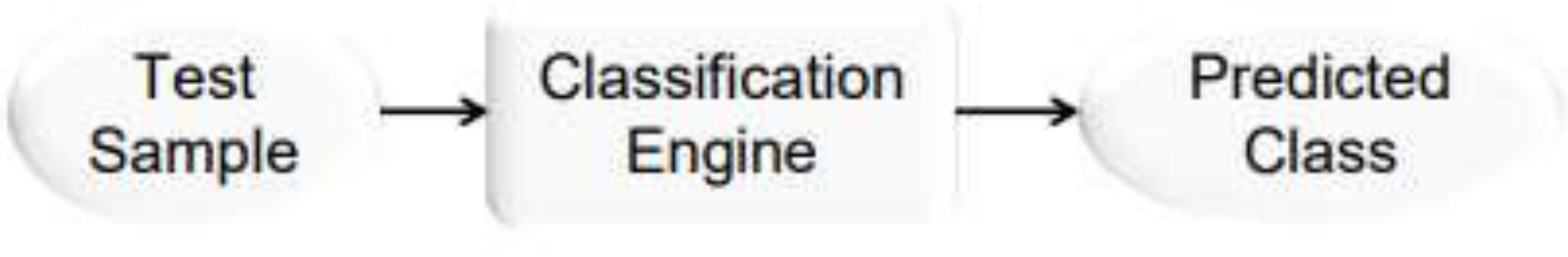


What?

Why?

How?

Classification



Test Sample → Classification Engine → Predicted Class

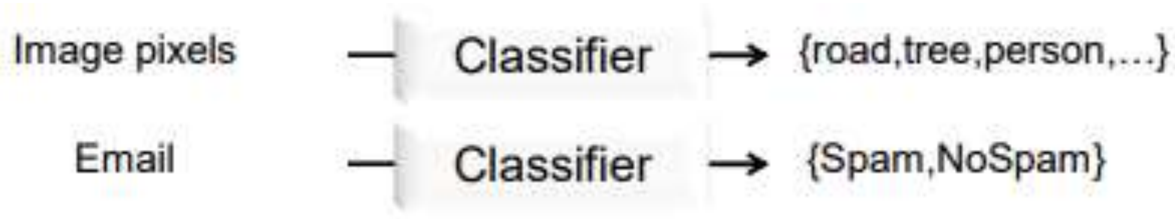
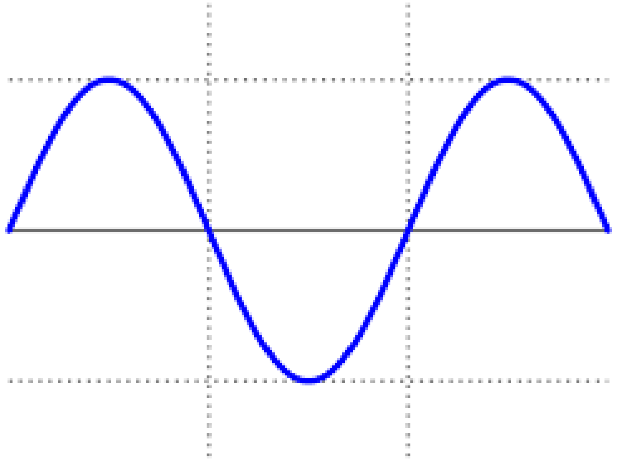


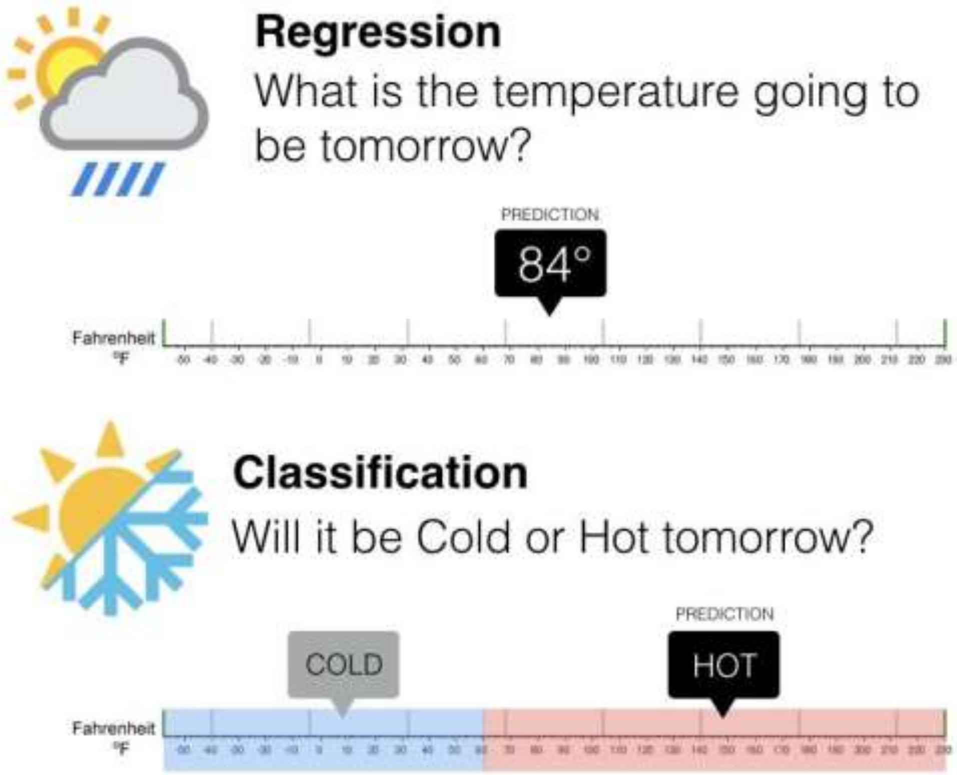
Image pixels — Classifier → {road,tree,person,...}

Email — Classifier → {Spam,NoSpam}

Classification vs. Regression

Discrete/categorical vs. real/continuous





Regression

What is the temperature going to be tomorrow?

Classification

Will it be Cold or Hot tomorrow?

http://www.medium.com

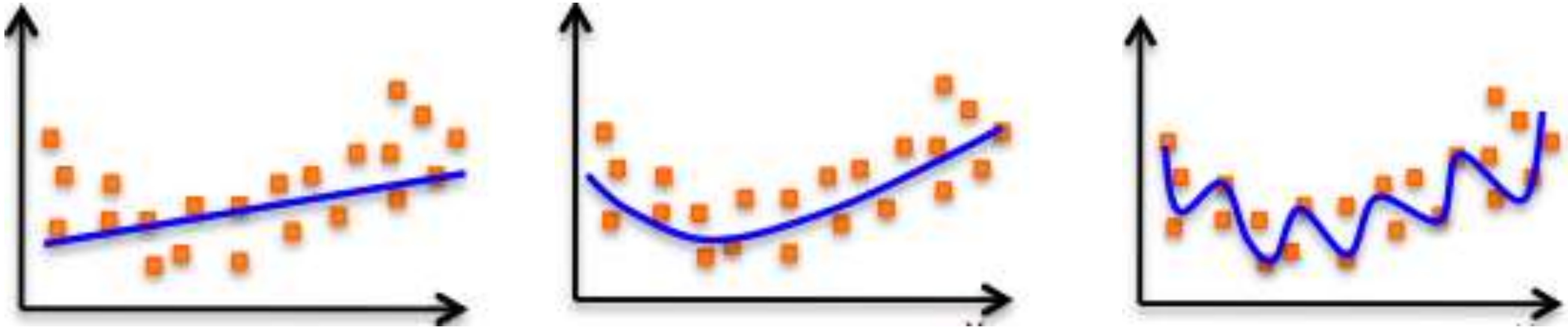
Training ‘the engine’

- Training samples to train the engine/model and for each training sample, we have our input features, and a class label
- Input features are things we can measure or observe about our samples

Training ‘the engine’

- Then, given some unknown test sample, extract the features we used for our training samples
- Try and predict the test sample’s class label

A curse: over-fitting the engine



A curse: over-fitting the engine

Real-world example?

Validation and Testing

- Samples are hard to obtain!
- Training partition (60%)
- Validation/verification partition (20%) to select the optimal model

Validation and Testing

- Independent test (20%) set to report a non-biased measure of performance
- Train with training samples, optimize model with verification set, then verify performance on test set

Machine Learning Project in Python

Goal

Evaluate 4 different engines/algorithms for the IRIS dataset:

- K-Nearest Neighbors (KNN)
- Gaussian Naive Bayes (NB)
- Random Forest (RF)
- Logistic Regression (LR)
- https://github.com/mariashoaib01/Machine-Learning-101

Steps

- Step 1: package installations
- Step 2: import libraries and load dataset
- Step 3: models and cross validation
- Step 4: score and select model
- Step 5: test set model accuracy

Step 1: package installations

- Installing required packages
- Anaconda vs. scipy, numpy, pandas etc. individually
- https://www.continuum.io/
- OS X, Windows versions
- scikit-learn, theano, tensorflow, and keras.

Step 2: import libraries and load dataset

```
from sklearn.neighbors import KNeighborsClassifier

from sklearn.naive_bayes import GaussianNB

from sklearn.ensemble import RandomForestClassifier

from sklearn.linear_model import LogisticRegression
```

Step 2: import libraries and load dataset

```
import pandas from pandas.tools.plotting

import scatter_matrix

import matplotlib.pyplot as plt

from sklearn import model_selection
```

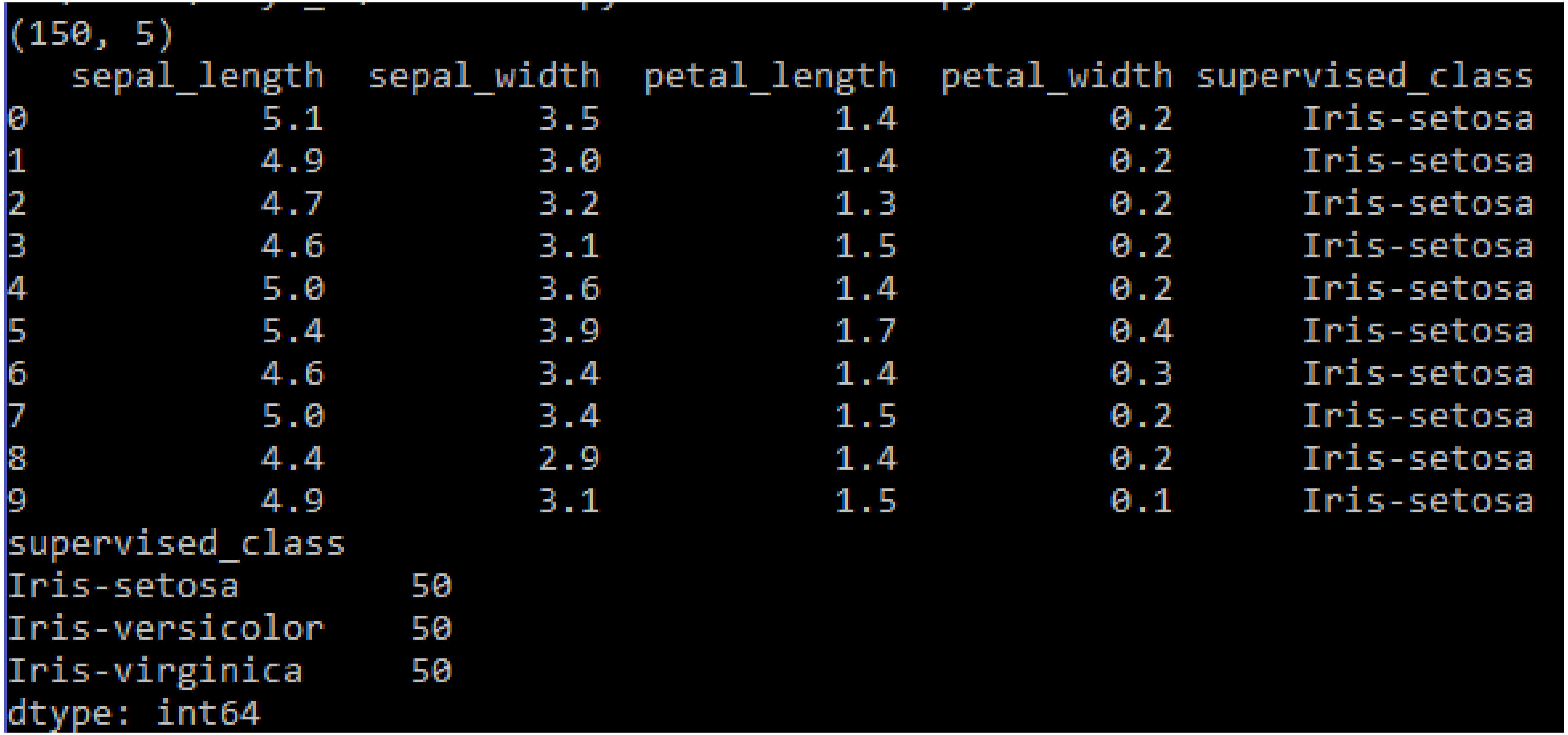
Step 2: import libraries and load dataset

```
from sklearn.metrics import classification_report

from sklearn.metrics import confusion_matrix

from sklearn.metrics import accuracy_score

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
```

Step 2: import libraries and load dataset

```
names = ['sepal_length', 'sepal_width', 'petal_length',
'petal_width', 'supervised_class’]

dataset = pandas.read_csv(url, names=names)

print(dataset.shape)

print(dataset.head(10))

print(dataset.groupby('supervised_class').size())
```

Step 2: import libraries and load dataset



```
(150, 5)
   sepal_length  sepal_width  petal_length  petal_width supervised_class
0           5.1          3.5           1.4          0.2      Iris-setosa
1           4.9          3.0           1.4          0.2      Iris-setosa
2           4.7          3.2           1.3          0.2      Iris-setosa
3           4.6          3.1           1.5          0.2      Iris-setosa
4           5.0          3.6           1.4          0.2      Iris-setosa
5           5.4          3.9           1.7          0.4      Iris-setosa
6           4.6          3.4           1.4          0.3      Iris-setosa
7           5.0          3.4           1.5          0.2      Iris-setosa
8           4.4          2.9           1.4          0.2      Iris-setosa
9           4.9          3.1           1.5          0.1      Iris-setosa
supervised_class
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

Step 3: models and cross validation

```
seed = 10

array = dataset.values

X = array[:,0:4]

Y = array[:,4]
```

Step 3: models and cross validation

```
val_partition = 0.20

X_train, X_val, Y_train, Y_val =
model_selection.train_test_split(X, Y,
test_size=val_partition, random_state=seed)
```

Step 3: models and cross validation

```
engines = []

engines.append(('KNN', KNeighborsClassifier()))

engines.append(('NB', GaussianNB()))

engines.append(('RF', RandomForestClassifier()))

engines.append(('LR', LogisticRegression()))
```

Step 3: models and cross validation

```
results = []

names = []

no_of_splits=10

scoring = 'accuracy'
```

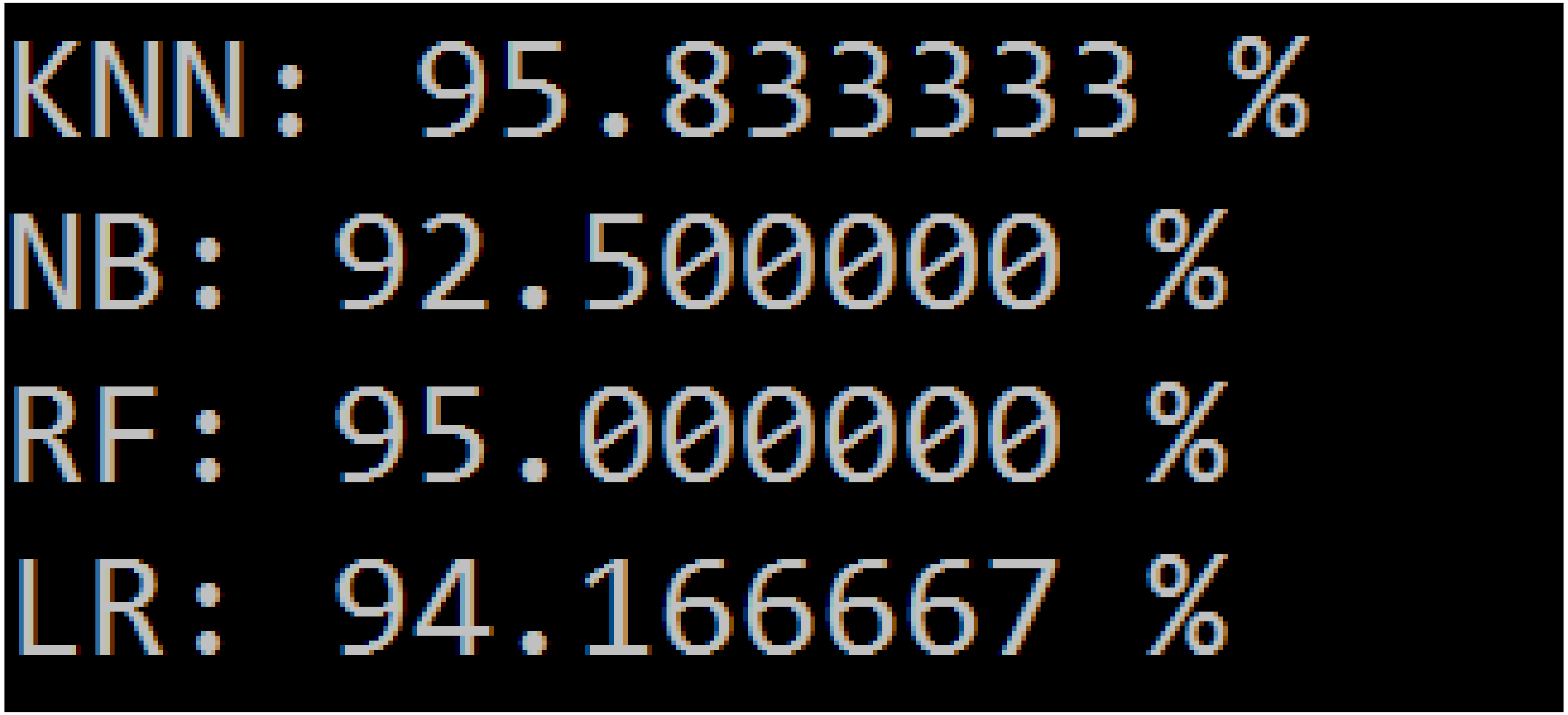
Step 3: models and cross validation

```
for name, model in engines:
     kfold_cross_val = model_selection.Kfold(
     no_of_splits, random_state=seed)
     cross_val_results = model_selection.
     cross_val_score(model, X_train, Y_train, cv=
     kfold_cross_val, scoring = scoring)
```

Step 3: models and cross validation

```
results.append(cross_val_results)

names.append(name)

msg = "%s: %f" % (name, 100.0*

cross_val_results.mean())

print(msg)
```

Step 4: score and select model



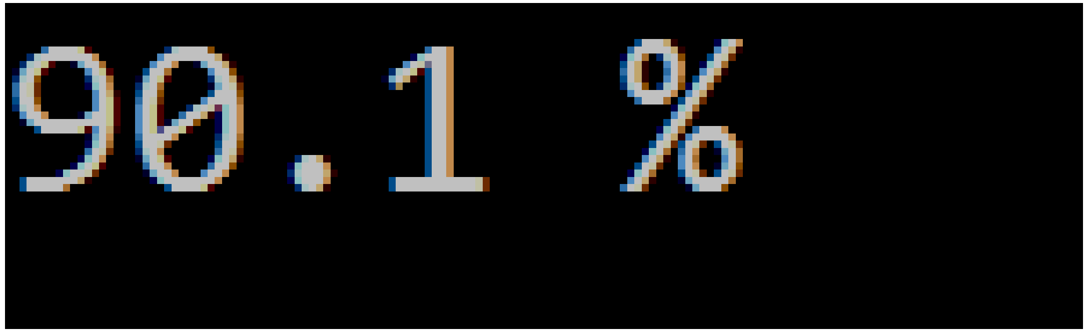
```
KNN: 95.833333 %
NB: 92.500000 %
RF: 95.000000 %
LR: 94.166667 %
```

Step 5: test set model accuracy

```
knn= KNeighborsClassifier()

knn.fit(X_train, Y_train)

predictions = knn.predict(X_val)

print(accuracy_score(Y_val, predictions))
```

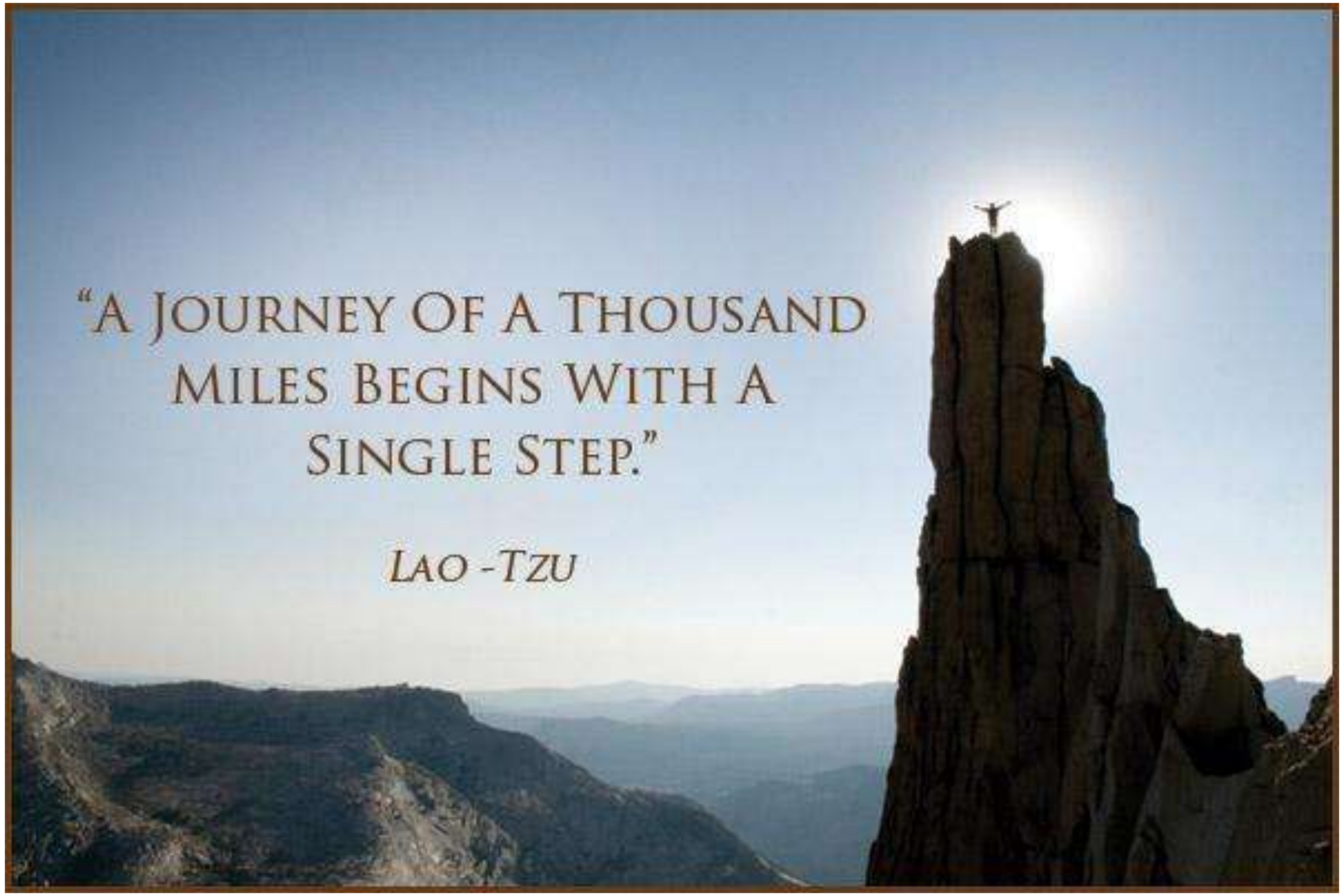
Step 5: test set model accuracy

- **KNN**



```
90.1 %
```

Key takeaways



http://www.picturequotes.com

Key takeaways

- Don’t Get Overwhelmed by the “overwhelmingness”
- Don’t need to understand how algorithms work. ATM at least!
- Don’t have to be a Python expert, just learn by starting out

Helpful Resources

- Packages: https://www.continuum.io/
- Machine learning tutorials: https://pythonprogramming.net/machine-learning-tutorial-python-introduction/

Helpful Resources

- Andrew Ng's Machine Learning course from Stanford via Coursera
- Remember, you can always do help(“FunctionName”) in Python

Thank you, any questions?