HANDWRITTEN DIGITS RECOGNITION

Scientific Experimentation & Evaluation

YOUR TASK

- Experiment with different machine learning algorithms to identify the most suitable algorithm for recognising handwritten digits.
- Have an unbiased evaluation of different algorithms.
- Present your results and explain your selection.

MNIST HANDWRITTEN DATA SET

- The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

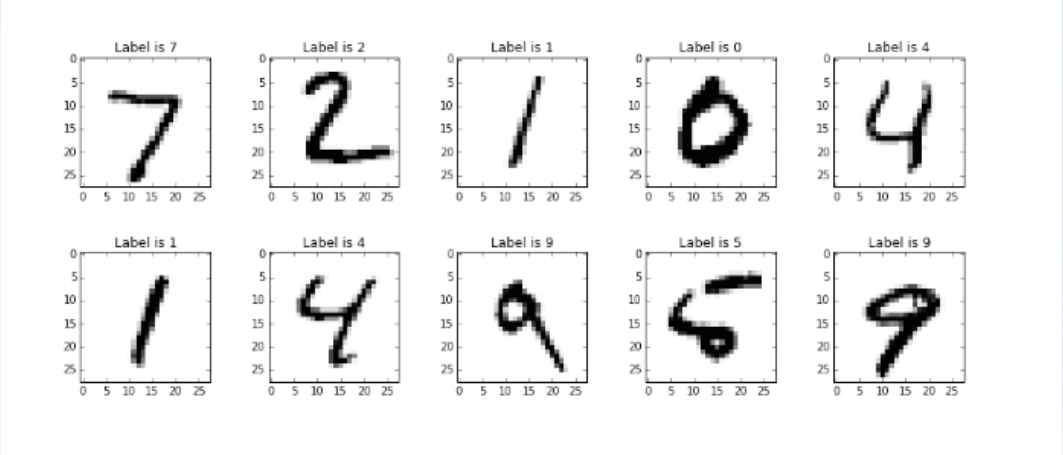
http://yann.lecun.com/exdb/mnist/

MNIST HANDWRITTEN DATA SET

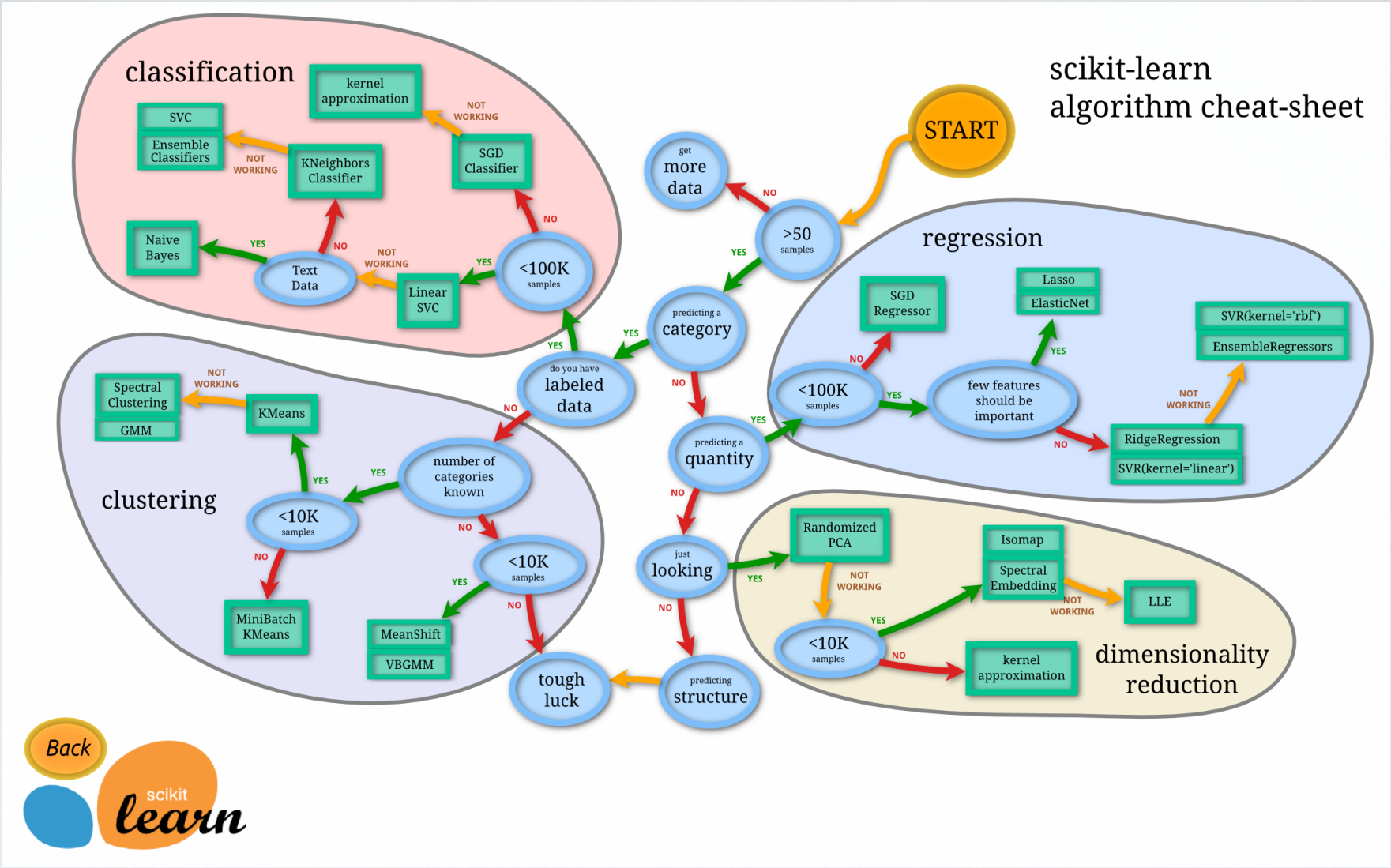
- Each image size is 28x28 pixels.
- Each image has a label corresponding to it's value.
- The data set is in an idx3 format, when changed into a CSV file or array, each number in the data set then represents the value of the pixel.

MNIST HANDWRITTEN DATA SET

- Images can be reconstructed from the pixel values for better representation of the data.



MACHINE LEARNING



WHAT KIND OF PROBLEM ARE WE DEALING WITH?

- Classification: Identifiying to which category an object belongs to.
- Regression: Predicting a continuous-valued attribute associated with an object.
- Clustering: Automatic grouping of similar objects into sets.

http://scikit-learn.org/

CLASSIFICATION

CLASSIFICATION METHODS

- Support Vector Machines: Set of supervised learning methods, which can be used for both classification and regression.
- K-Nearest Neighbours: The principle behind nearest neighbour methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these.
- Decision Tree: The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
- There are other methods used for classification such as:
    - Stochastic Gradient Decsent.
    - Random Forest.

EVALUATION

HOW TO EVALUATE LEARNT MODELS

- Learning Accuracy.
- Random Partitioning.
- Cross Validation.
- Confusion Matrices.
- Learning Curves.
- Precision.
- Recall.
- F1-score.
- Error.
- True positive rate.
- False Positive rate.

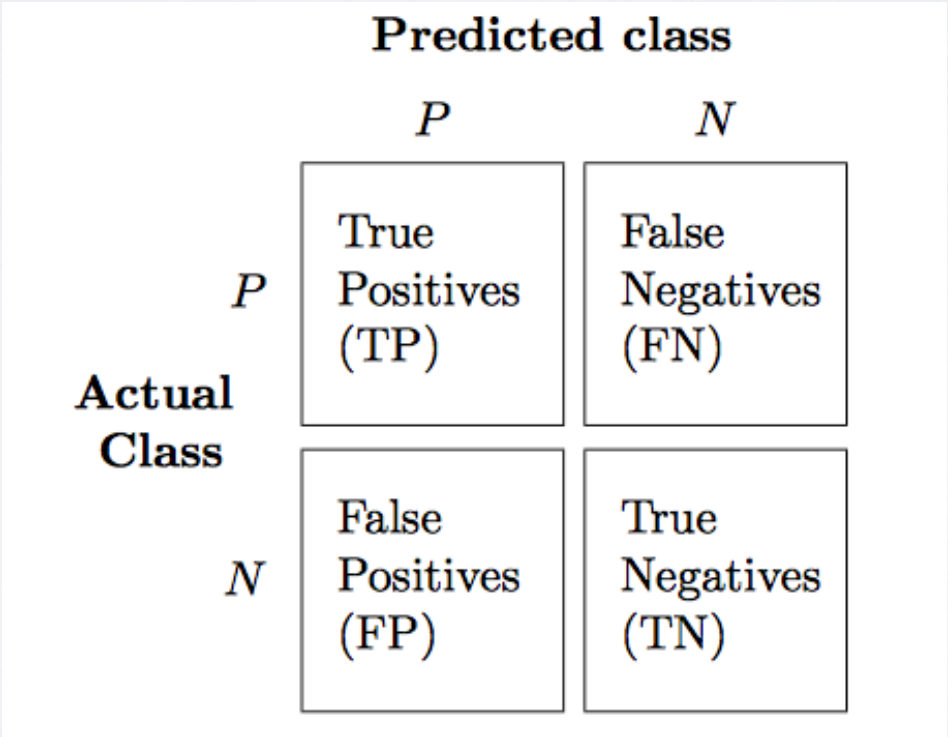
CROSS VALIDATION

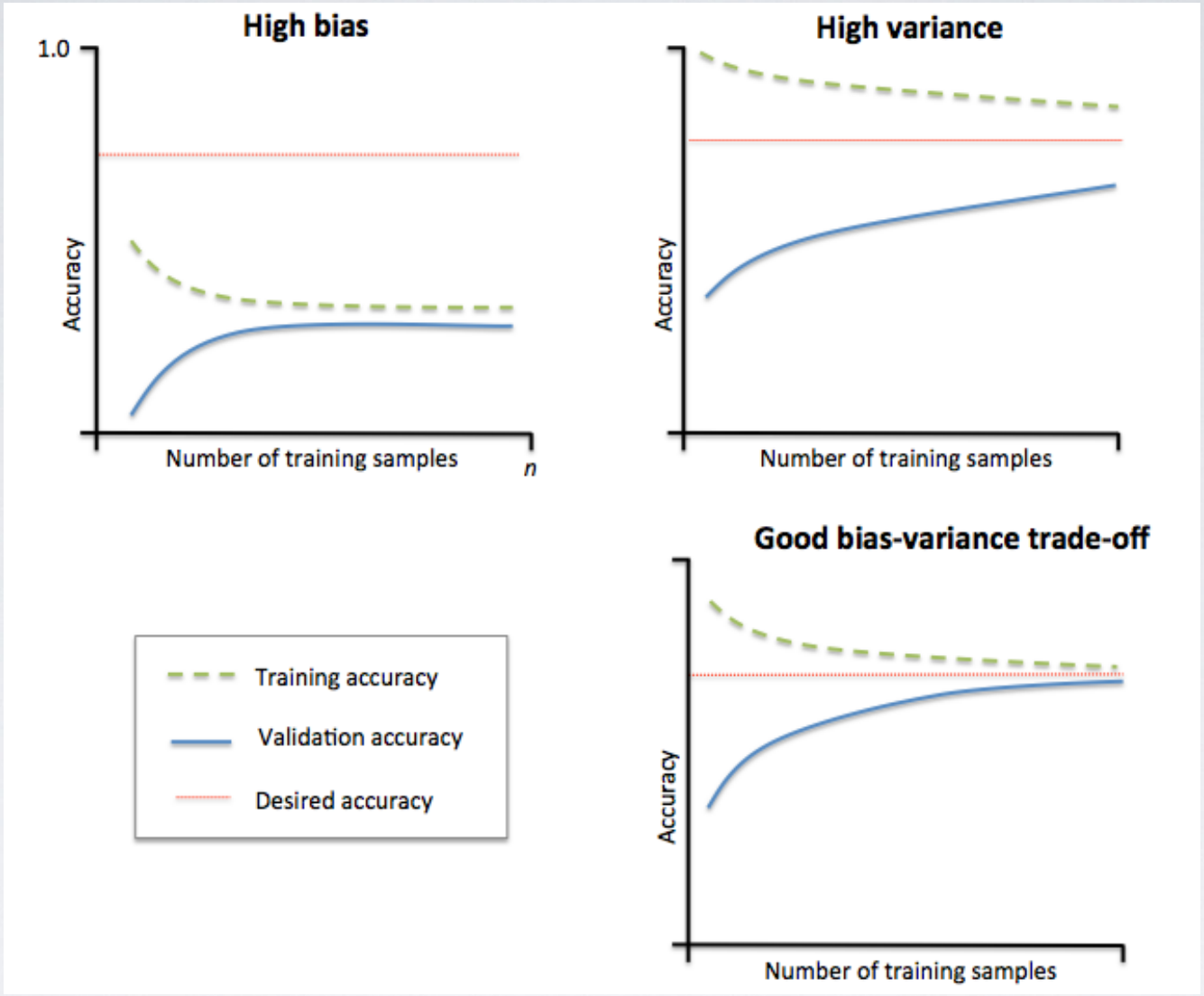
- Evaluate the model over data it has not seen before.
- Split the data into 3 partitions, training, validation, and test set.
- Use the training and validation sets to create model and use the test set to test the model on unseen data.
- Randomly partition training, validation, and test set to get a more unbiased evaluation.

K-FOLD CROSS VALIDATION

CONFUSION MATRIX

- It is a square matrix that reports the counts of true positives, true negatives, false positives, and false negatives.





LEARNING CURVES

ERROR & ACCURACY

$$ERR = \frac{FP + FN}{FP + FN + TP + TN}$$

Sum of false predictions over total number of predictions.

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - ERR$$

Sum of true prediction over total number of predictions.

TRUE POSITIVE RATE & FALSE POSITIVE RATE

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN}$$

$$TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$$

PRECISION, RECALL & F1 SCORE

$$PRE = \frac{TP}{TP + FP}$$

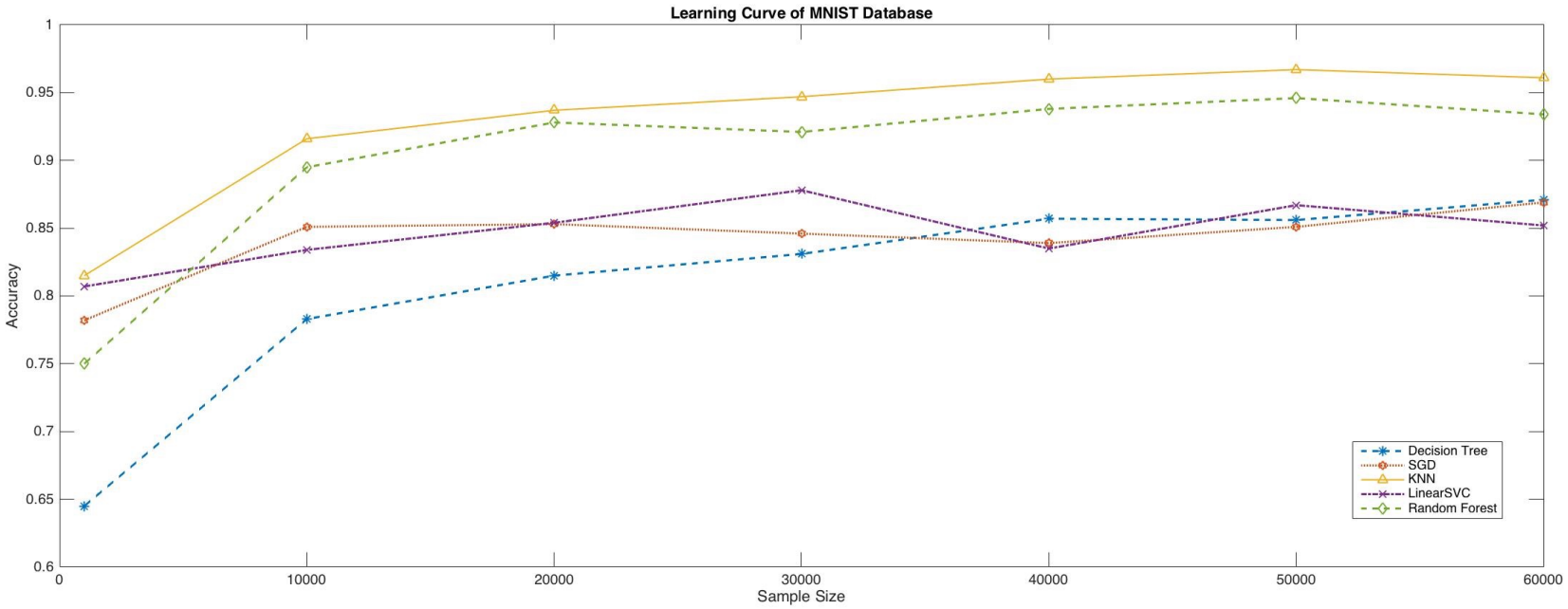
$$F1 = 2\frac{PRE \times REC}{PRE + REC}$$

$$REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$$

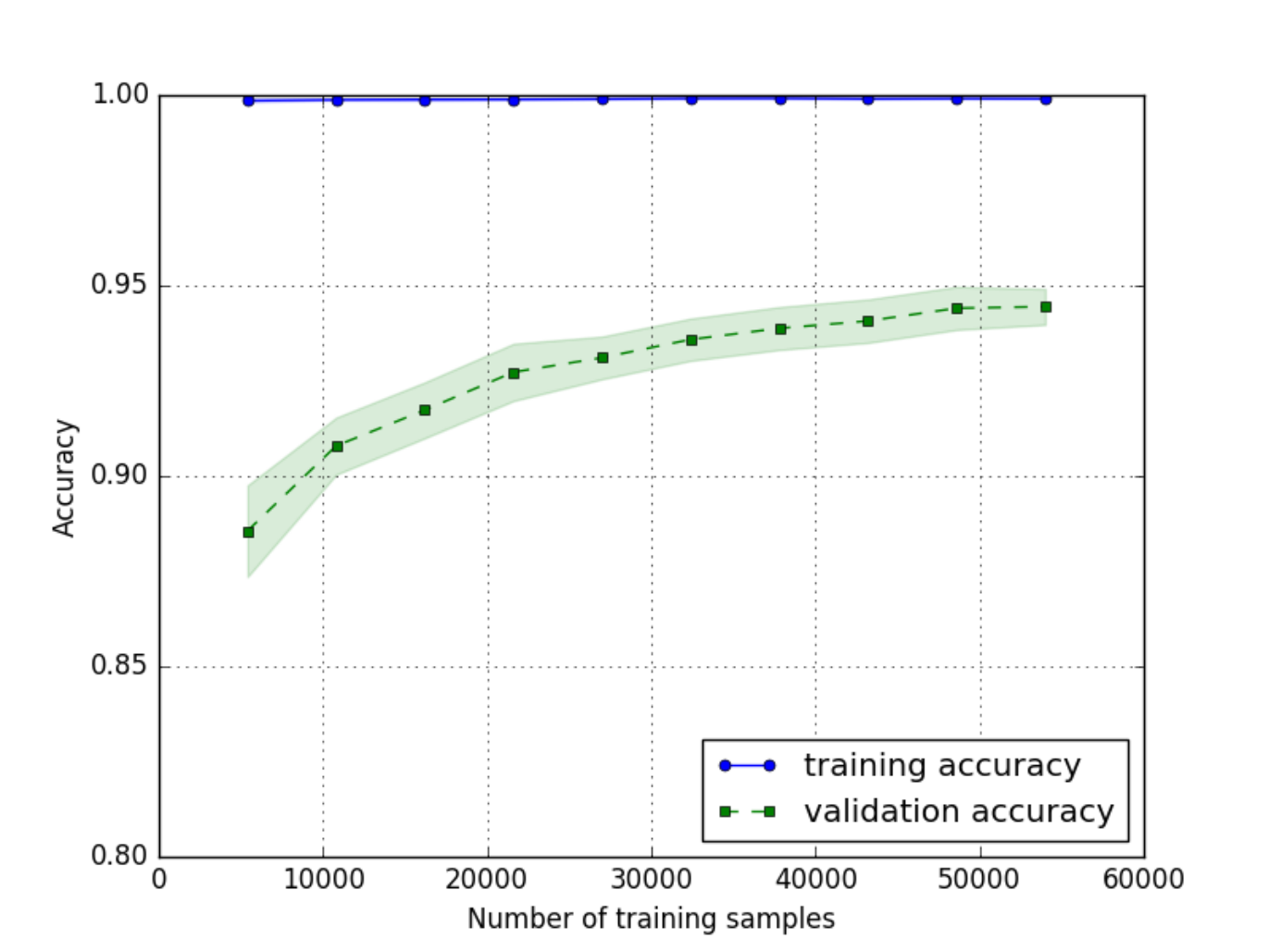
SAMPLE RESULTS

```
Classification report for classifier RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False):
             precision    recall  f1-score   support

        0.0       0.95      0.99      0.97       980
        1.0       0.98      0.99      0.98      1135
        2.0       0.93      0.96      0.94      1032
        3.0       0.94      0.94      0.94      1010
        4.0       0.94      0.95      0.94       982
        5.0       0.95      0.93      0.94       892
        6.0       0.96      0.96      0.96       958
        7.0       0.96      0.94      0.95      1028
        8.0       0.95      0.92      0.93       974
        9.0       0.94      0.91      0.93      1009

avg / total       0.95      0.95      0.95     10000


Confusion matrix:
[[ 967    1    1    0    0    3    4    1    2    1]
 [   0 1121    5    1    0    2    3    0    3    0]
 [  11    3  988    6    5    0    4    8    7    0]
 [   3    1   19  948    0   12    1   14   10    2]
 [   5    3    5    1  931    0    8    2    3   24]
 [  10    1    4   23    6  827    7    1    6    7]
 [  10    3    2    0    8   11  924    0    0    0]
 [   2    5   19    6    7    1    0  971    4   13]
 [   5    2   15   15    7   13    7    3  895   12]
 [   8    3    6   10   29    6    1    9   14  923]]
Classification accuracy score:
0.9495
```

SAMPLE RESULTS



SAMPLE RESULTS



SAMPLE RESULTS

```
start
data loaded
Test Accuracy: 0.999
CV accuracy scores: [ 0.95004995  0.94002999  0.93517747  0.9495      0.94433333  0.9429905
  0.94082347  0.93882314  0.94013674  0.95697131]
CV accuracy: 0.944 +/- 0.006
```

REFERENCES

- http://scikit-learn.org/stable/
- Raschka, Sebastian. "Python Machine Learning." (2015).