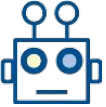


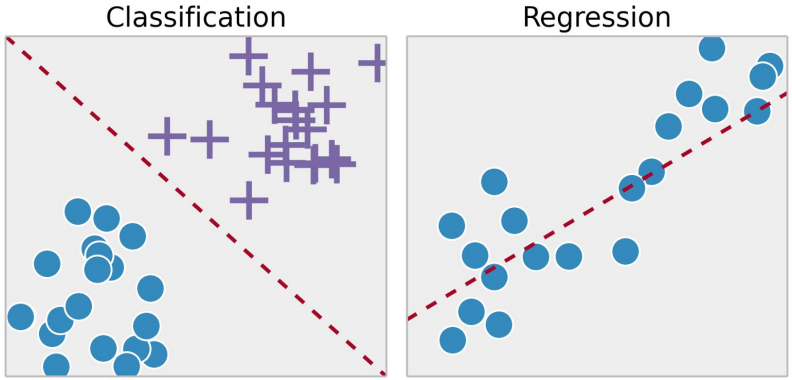
Machine Learning for Networking

Seminar 3

Marc Carrascosa Zamacois: marc.carrascosa@upf.edu

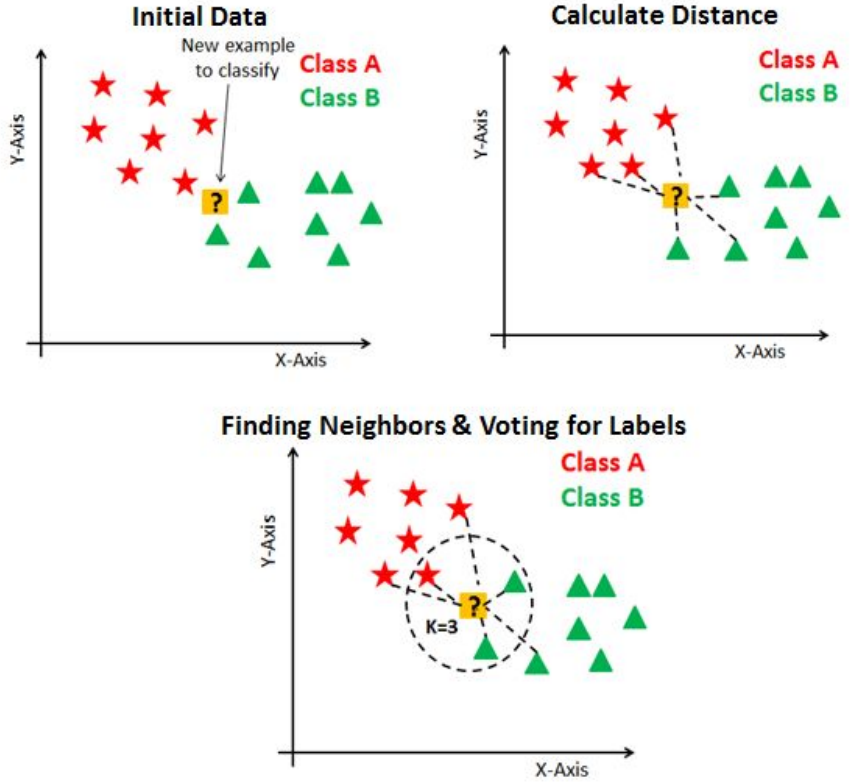
Classification

- Used when we have a finite output
- Instead of predicting a real value (like throughput in Mbps), we predict a class (is the user active? yes/no)
- Several methods available
  - K-Nearest Neighbours
  - Logistic Regression
  - Naive Bayes
  - Support Vector Machines
  - Decision Trees



K-Nearest Neighbour

- The model are the training labels
- For each new data point we calculate the distance between it and every training point
- We then take the K data points that have the shortest distance
- We use the mode to choose the class



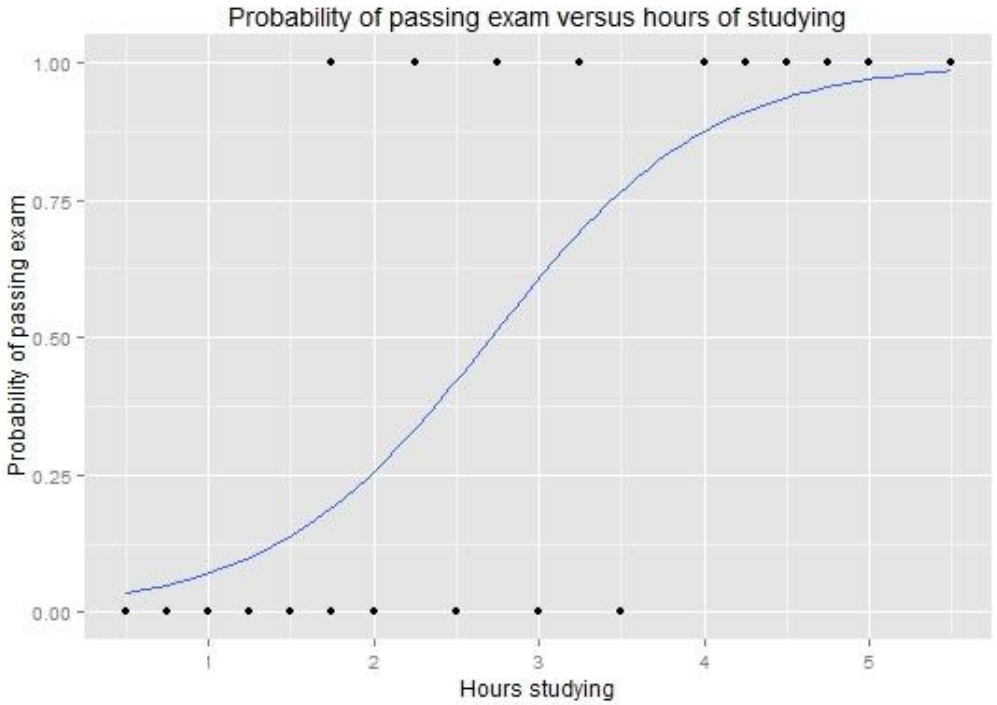
Probabilistic classifiers

- Naive Bayes classifiers are based on Bayes' theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

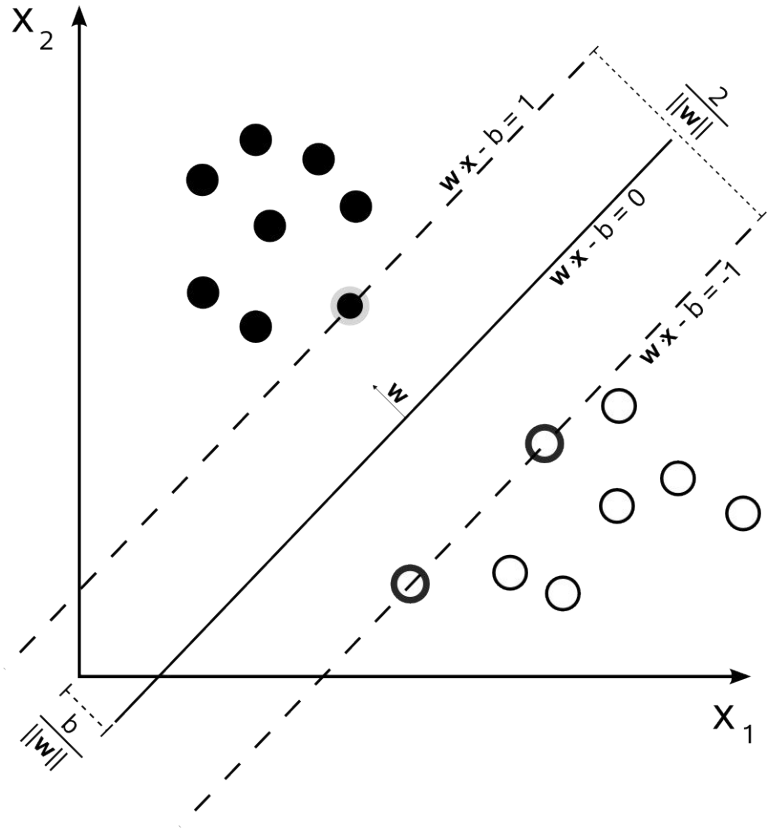
- Logistic Regression calculates the log odds of an event using a sigmoid function

$$P(Y=1) = \frac{1}{1+e^{-(b_0+b_1x_1+b_2x_2)}}$$

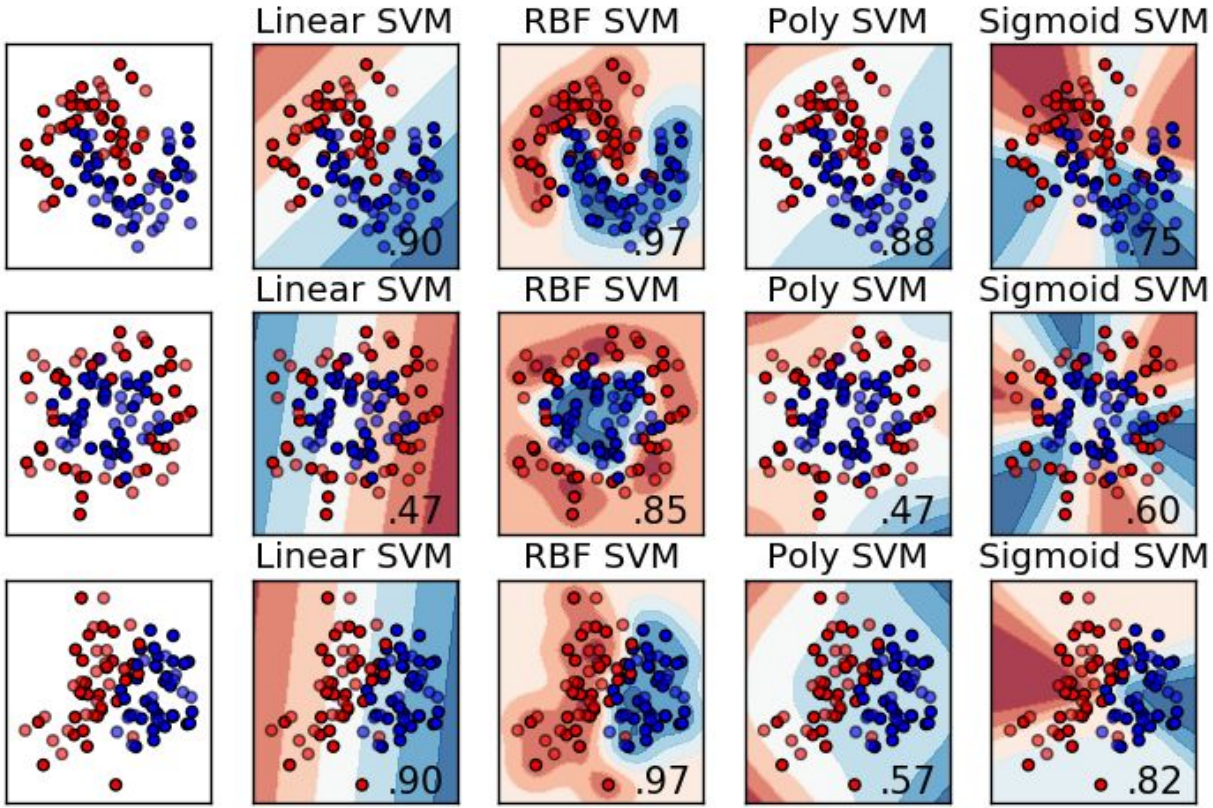


Support Vector Machines

- There are multiple vectors that can separate the data
- SVM finds the one that maximizes the separation, minimizing the possibility of errors
- SVM are highly customizable

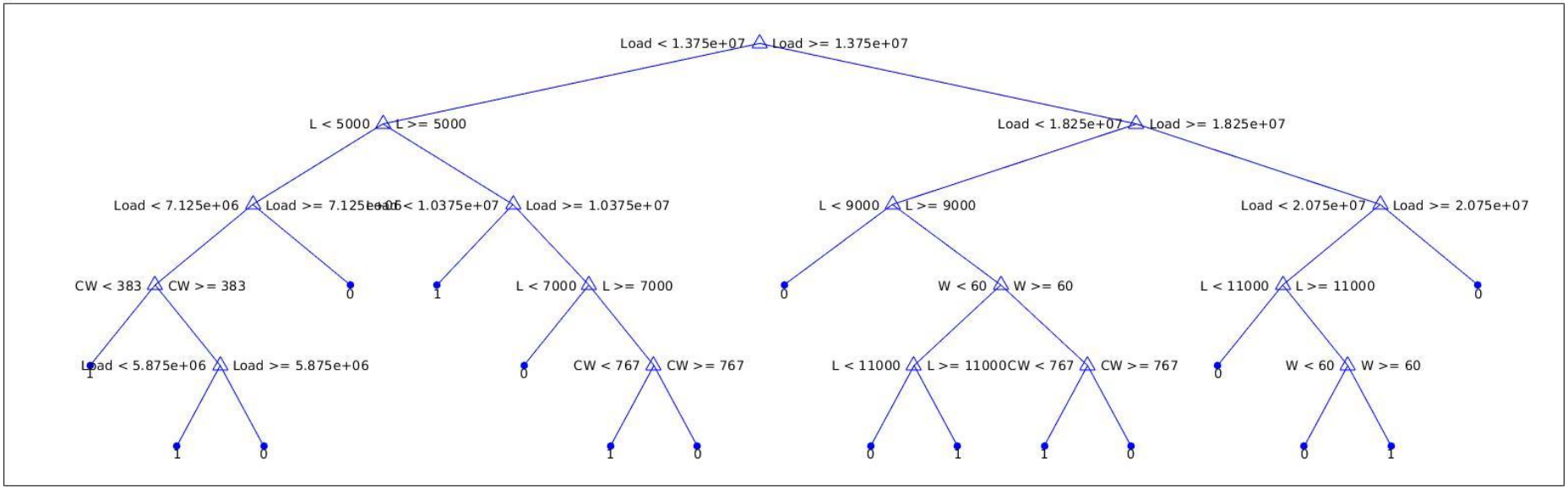


SVM kernels

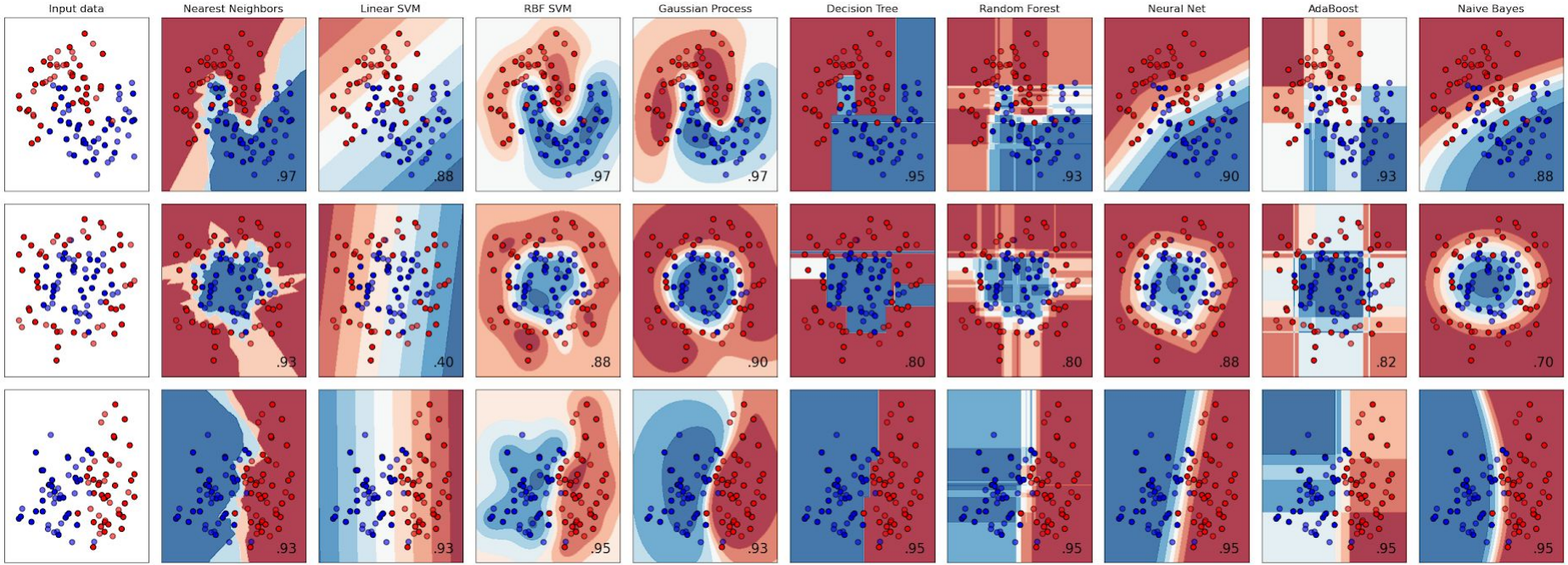

Linear SVM
RBF SVM
Poly SVM
Sigmoid SVM
.90
.97
.88
.75
Linear SVM
RBF SVM
Poly SVM
Sigmoid SVM
.47
.85
.47
.60
Linear SVM
RBF SVM
Poly SVM
Sigmoid SVM
.90
.97
.57
.82

Decision Trees

We saw them last seminar!

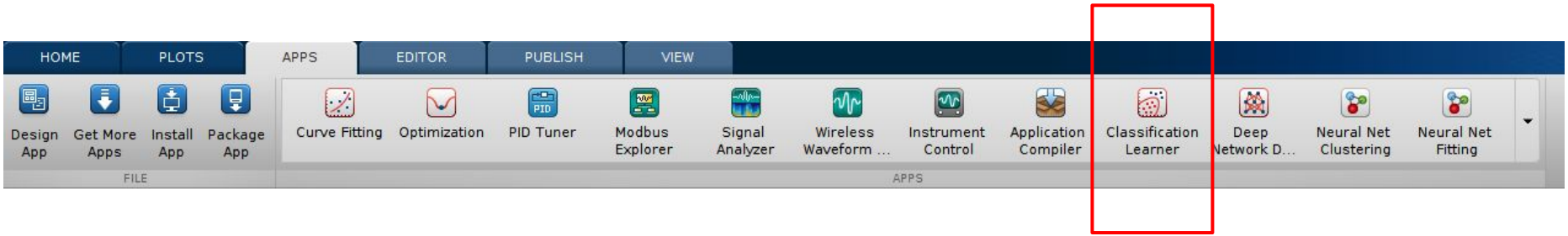


Classifiers



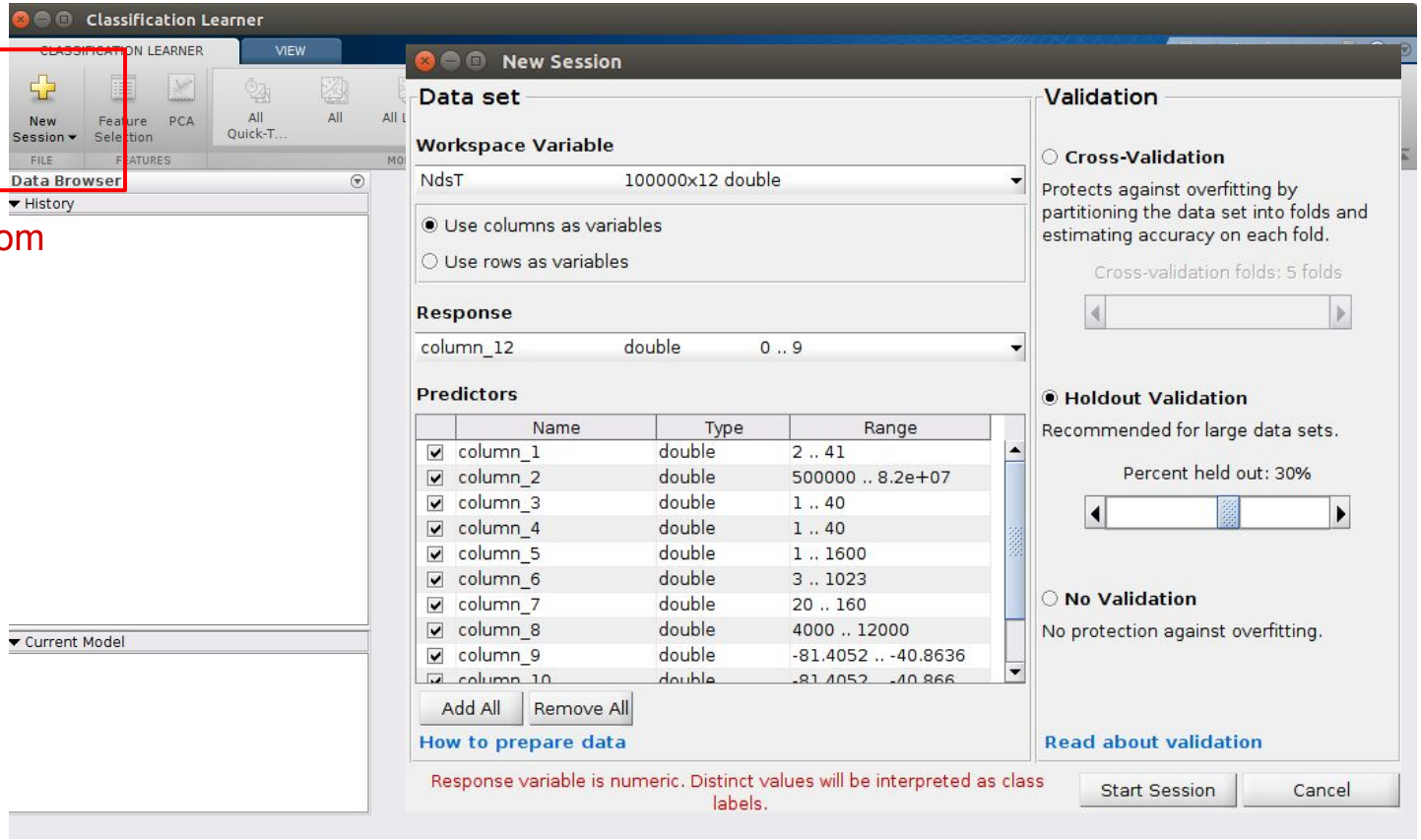
MATLAB classification learner

- MATLAB has a classification tool that simplifies the model selection/ training process
- It can be intensive on you RAM and CPU usage



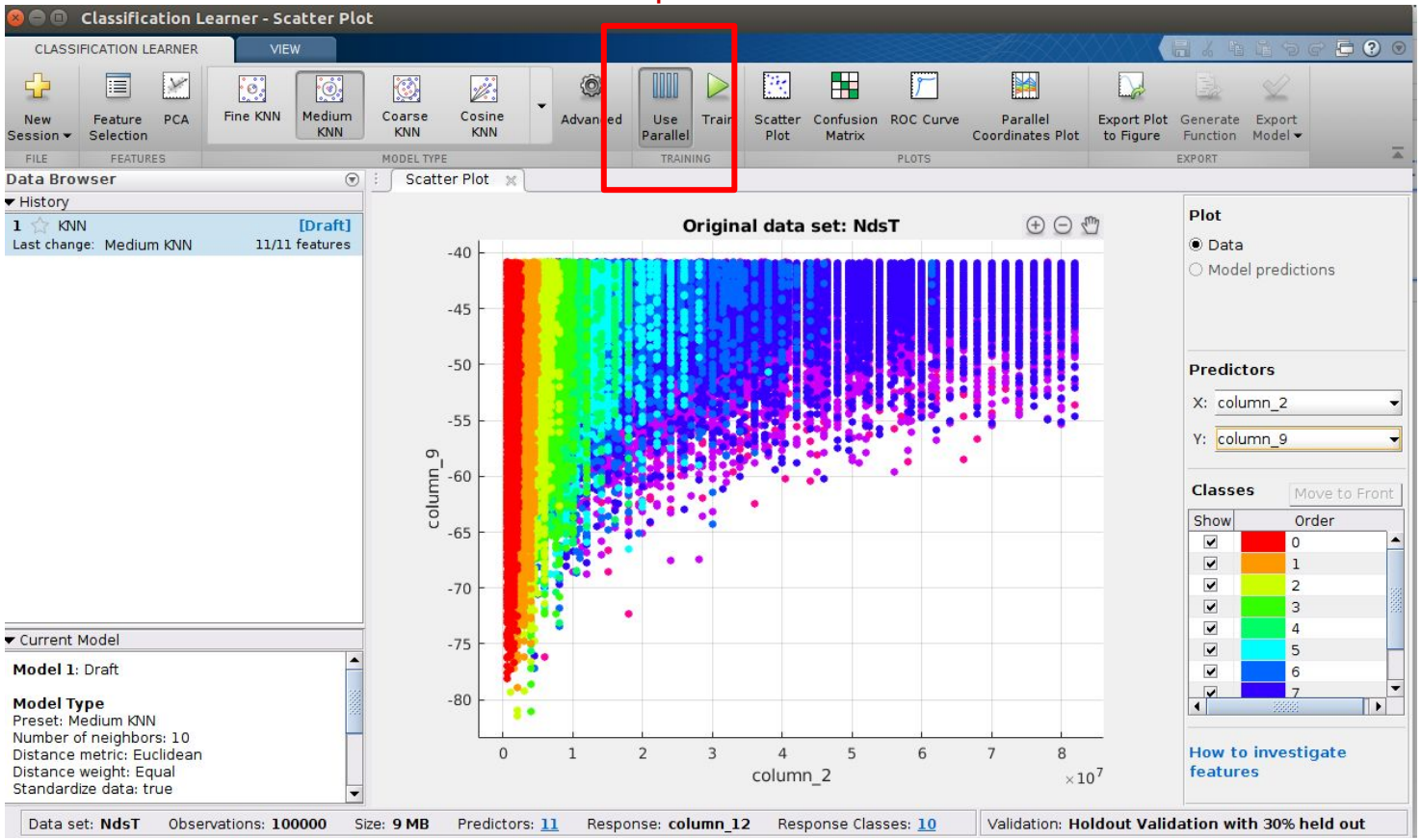
Importing the dataset

Import the dataset from your workspace

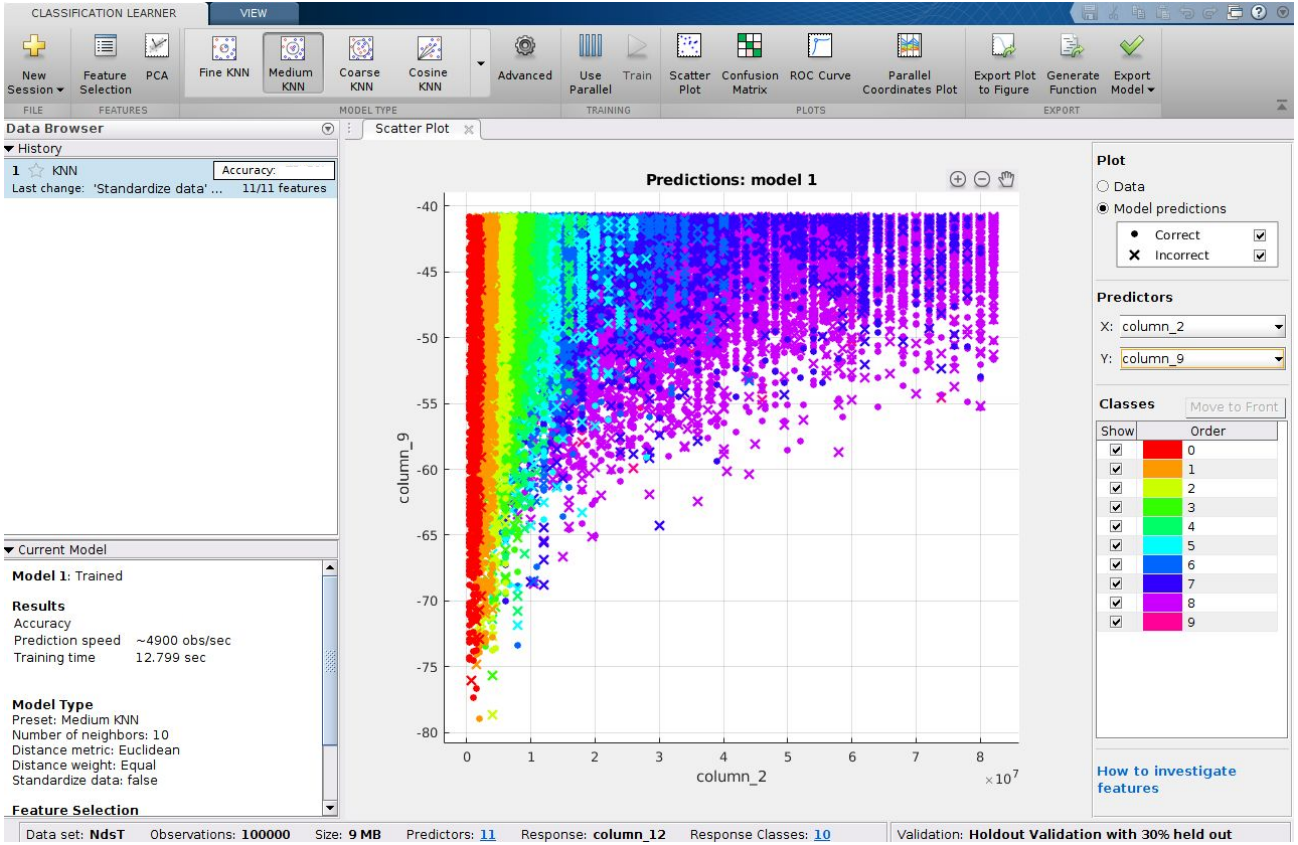


Classification learner

Disable parallel computing to improve performance!

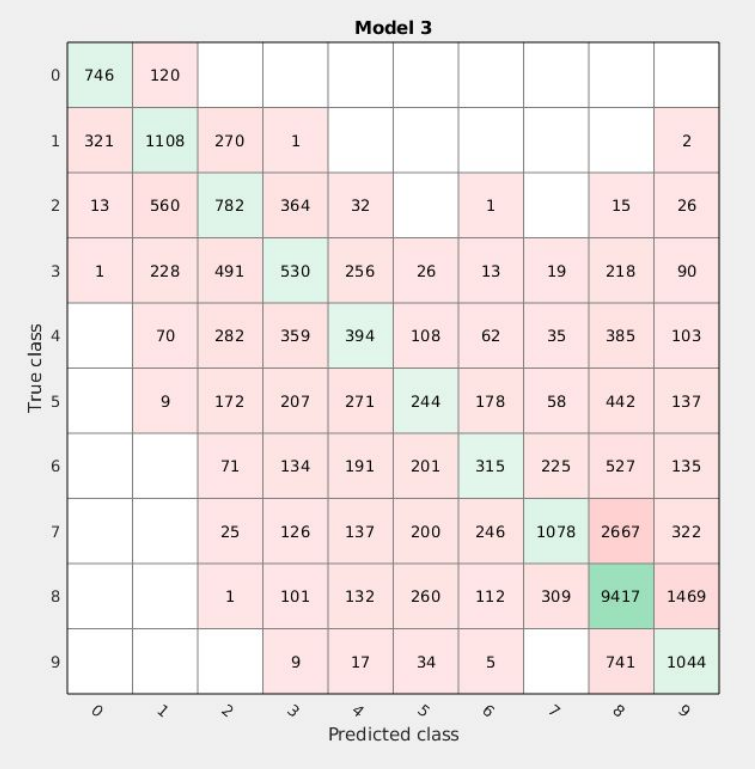


Classification learner

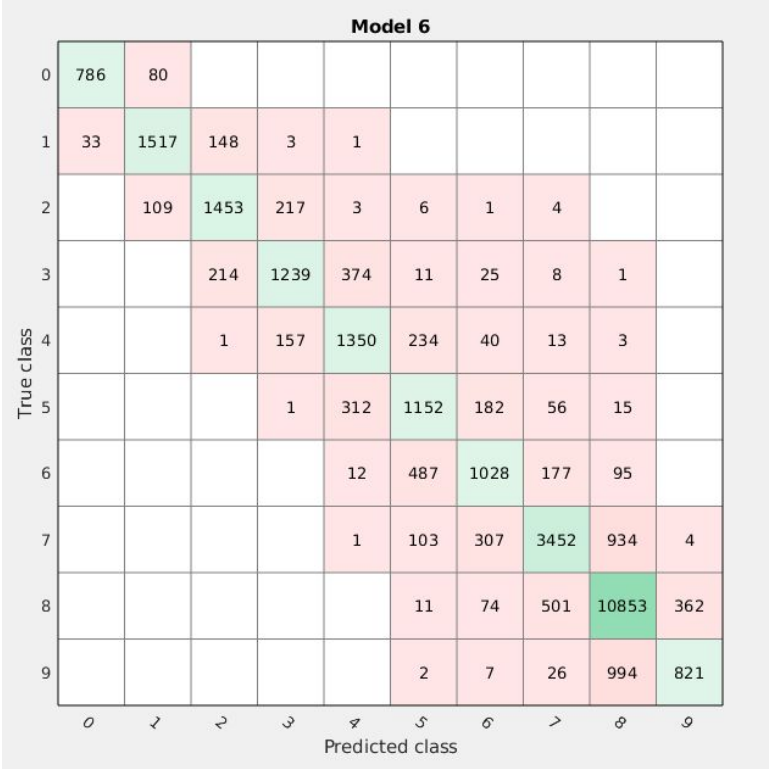


- Plots show where predictions failed once the model is trained
- You can also check the confusion matrix
- Models can be compared and once one is chosen exported to the workspace

Classification learner



10-Nearest Neighbour



Decision tree

Classification learner

Gaussian Naive Bayes

Model 4: Trained

Results

Accuracy	
Prediction speed	~580000 obs/sec
Training time	1.4087 sec

Kernel Naive Bayes

Model 5: Trained

Results

Accuracy	
Prediction speed	~200 obs/sec
Training time	303.59 sec

6 SVM Training

Last change: 'Standardize data' = 'f... 11/11 features

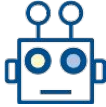
Training time: hours

Current Model

Model 6: Training

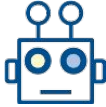
Model Type
Preset: Linear SVM
Kernel function: Linear
Kernel scale: Automatic
Box constraint level: 1
Multiclass method: One-vs-One
Standardize data: false

To reduce execution time, reduce the features used (based on what we saw in past seminars)



Commands

- c = confusionmat(y_test,y_predicted);
  - Get confusion matrix
- confusionchart(c)
  - Plot confusion matrix
- fitcknn(x,y,'NumNeighbors',k)
  - Fit k-nearest neighbours classifier
- fitcnb(x,y)
  - Fit naive bayes classifier
- fitcecoc(x,y,'Learners','Linear')
  - Fit SVM
- loss(Model, x_test, y_test)
  - Calculate error for classification model
  - Alternative sum(ytest==ypred)/length(ytest)
- Fitctree (xtrain,ytrain,'MaxNumSplits',1000)
  - Decision tree



Tasks

- Load the seminar 1 dataset.
- We will use the airtime (column 16) as the output. Separate the airtime into 10 classes (class 0: [0, 0.1), class 1: [0.1, 0.2), etc.)
- Open the MATLAB classifier app.
- Train a couple of models for this new dataset and check their accuracy (Modify the number of neighbors of KNN, disable the use of standardized data, try both types of Bayesian classifiers, change the depth of the decision trees, etc). How high can the accuracy go?
- Choose a new classification for the data, you can use the information from the previous testing, check the distribution of the data, or any method you prefer (hint: use less classes)
- Re-train the models with the new classification. Are the results better or worse than before?