ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΚΕΝΤΡΟ ΕΠΙΜΟΡΦΩΣΗΣ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗΣ

Επιμορφωτικό Πρόγραμμα Επίλυση Προβλημάτων με Τεχνητή Νοημοσύνη και Προηγμένοι Αλγόριθμοι Εκτίμησης (Problem Solving with Artificial Intelligence and Advanced Estimation Algorithms)

ΕΡΓΑΣΙΕΣ

Εκπαιδευτική Ενότητα 2 - Μηχανική Μάθηση και Τεχνητή Νοημοσύνη

ΑΣΠΡΟΥΔΗΣ ΙΑΣΟΝΑΣ-ΧΡΙΣΤΟΦΟΡΟΣ

ΑΘΗΝΑ 2022

Contents

1. Προηγμένες εφαρμογές (toolboxes/packages) & Σύνθετα Εργαλεία Μηχανικής Μάθησης στα MatLab/Octave

***Α) Αναζητήστε στο Διαδίκτυο**: Τα πρόσθετα εργαλεία και τις εφαρμογές (toolboxes/packages) των MatLab/Octave για Μηχανική Μάθηση. Περιγράψτε τι υλοποιεί το κάθε εργαλείο και ελέγξτε ότι είναι εγκατεστημένα στον Η/Υ σας.*

Statistics and Machine Learning Toolbox

Analyze and model data using statistics and machine learning

Statistics and Machine Learning Toolbox™ provides functions and apps to describe, analyze, and model data. You can use descriptive statistics, visualizations, and clustering for exploratory data analysis; fit probability distributions to data; generate random numbers for Monte Carlo simulations, and perform hypothesis tests. Regression and classification algorithms let you draw inferences from data and build predictive models either interactively, using the Classification and Regression Learner apps, or programmatically, using AutoML.

For multidimensional data analysis and feature extraction, the toolbox provides principal component analysis (PCA), regularization, dimensionality reduction, and feature selection methods that let you identify variables with the best predictive power.

The toolbox provides supervised, semi-supervised, and unsupervised machine learning algorithms, including support vector machines (SVMs), boosted decision trees, shallow neural nets, k-means, and other clustering methods. You can apply interpretability techniques such as partial dependence plots, Shapley values and LIME, and automatically generate C/C++ code for embedded deployment. Native Simulink blocks let you use predictive models with simulations and Model-Based design. Many toolbox algorithms can be used on data sets that are too big to be stored in memory.

Descriptive Statistics and Visualization

Explore data numerically by generating summary statistics, including measures of central tendency, dispersion, shape, and correlation. Visualize your data using univariate, bivariate, and multivariate plots. Available options include box plots, histograms, and probability plots. Find natural groupings in your data using cluster analysis techniques such as hierarchical clustering and *k*-Means clustering.

Cluster Analysis

Identify patterns and features by applying k-means, hierarchical, DBSCAN and other clustering methods, and dividing data into groups or clusters. Determine the optimal number of clusters for the data using different evaluation criteria. Detect anomalies to identify outliers and novelties.

Cluster analysis, also called segmentation analysis or taxonomy analysis, partitions sample data into groups, or *clusters*. Clusters are formed such that objects in the same cluster are similar, and objects in different clusters are distinct. Statistics and Machine Learning Toolbox™ provides

several clustering techniques and measures of similarity (also called *distance metrics*) to create the clusters. Additionally, *cluster evaluation* determines the optimal number of clusters for the data using different evaluation criteria. *Cluster visualization* options include dendrograms and silhouette plots. The toolbox also provides several *anomaly detection* features to identify outliers and novelties.

ANOVA

Assign sample variance to different sources and determine whether the variation arises within or among different population groups. Use one-way, two-way, multiway, multivariate, and nonparametric ANOVA, as well as analysis of covariance (ANOCOVA) and repeated measures analysis of variance (RANOVA).

Analysis of Variance (ANOVA) is a procedure for determining whether variation in the response variable arises within or among different population groups. Statistics and Machine Learning Toolbox™ provides one-way, two-way, and N-way analysis of variance (ANOVA); multivariate analysis of variance (MANOVA); repeated measures models; and analysis of covariance (ANCOVA).

Regression

Use the Regression Learner app or programmatically train and assess models such as linear regression, Gaussian processes, support vector machines, neural networks, and ensembles.

Regression models describe the relationship between a response (output) variable, and one or more predictor (input) variables. Statistics and Machine Learning Toolbox™ allows you to fit linear, generalized linear, and nonlinear regression models, including stepwise models and mixed-effects models. Once you fit a model, you can use it to predict or simulate responses, assess the model fit using hypothesis tests, or use plots to visualize diagnostics, residuals, and interaction effects.

Statistics and Machine Learning Toolbox also provides nonparametric regression methods to accommodate more complex regression curves without specifying the relationship between the response and the predictors with a predetermined regression function. You can predict responses for new data using the trained model. Gaussian process regression models also enable you to compute prediction intervals.

Classification

Use the Classification Learner app or programmatically train and validate models such as logistic regression, support vector machines, boosted trees, and shallow neural networks.

Classification is a type of supervised machine learning in which an algorithm “learns” to classify new observations from examples of labeled data. To explore classification models interactively, use the **Classification Learner** app. For greater flexibility, you can pass predictor or feature data with corresponding responses or labels to an algorithm-fitting function in the command-line interface.

To train regression models, such as logistic regression, regression trees, Gaussian process regression, and support vector regression, see Regression.

Dimensionality Reduction and Feature Extraction

Extract features from images, signals, text, and numeric data. Iteratively explore and create new features and select the ones that optimize performance. Reduce dimensionality by transforming existing features into new predictor variables and drop less descriptive features after transformation, or by applying automated feature selection.

Feature transformation techniques reduce the dimensionality in the data by transforming data into new features. *Feature selection* techniques are preferable when transformation of variables is not possible, e.g., when there are categorical variables in the data. For a feature selection technique that is specifically suitable for least-squares fitting, see Stepwise Regression.

Probability Distributions

Fit continuous and discrete distributions, use statistical plots to evaluate goodness-of-fit, and compute probability density functions and cumulative distribution functions for more than 40 different distributions.

Fit probability distributions to sample data, evaluate probability functions such as pdf and cdf, calculate summary statistics such as mean and median, visualize sample data, generate random numbers, and so on. Work with probability distributions using probability distribution objects, command line functions, or interactive apps. For more information about each of these options, see Working with Probability Distributions.

Hypothesis Tests

Draw inferences about a population based on statistical evidence from a sample. Perform t-tests, distribution tests, and nonparametric tests for one, paired, or independent samples. Test for autocorrection and randomness, and compare distributions.

Statistics and Machine Learning Toolbox™ provides parametric and nonparametric hypothesis tests to help you determine if your sample data comes from a population with particular characteristics.

Distribution tests, such as Anderson-Darling and one-sample Kolmogorov-Smirnov, test whether sample data comes from a population with a particular distribution. Test whether two sets of sample data have the same distribution using tests such as two-sample Kolmogorov-Smirnov.

Location tests, such as *z*-test and one-sample *t*-test, test whether sample data comes from a population with a particular mean or median. Test two or more sets of sample data for the same location value using a two-sample *t*-test or multiple comparison test.

Dispersion tests, such as Chi-square variance, test whether sample data comes from a population with a particular variance. Compare the variances of two or more sample data sets using a two-sample *F*-test or multiple-sample test.

Determine additional features of sample data by cross-tabulating, conducting a run test for randomness, and determine the sample size and power for a hypothesis test.

Industrial Statistics

Statistically analyze effects and data trends. Design experiments to create and test practical plans for how to manipulate data inputs to generate information about their effects on data outputs. Visualize and analyze time-to-failure data with and without censoring and monitor and assess the quality of industrial processes.

Statistics and Machine Learning Toolbox™ provides tools for designing experiments, analyzing reliability and survival data, process quality control, and data surveillance.

Design of experiments helps determine how certain factors impact the outcome (response) of a process. You can design experiments including full and fractional factorial, D-optimal, quasi-random, and response surface designs, or visualize experiment results.

Survival analysis studies the time until an event occurs. Visualize and estimate parameters, compute survival and hazard functions, and fit semi-parametric models to censored or uncensored lifetime data.

Statistical process control techniques monitor and assess the quality of industrial processes. Measure process capability, perform gage repeatability and reproducibility study, and monitor process data using control charts.

Analysis of Big Data with Tall Arrays

Use tall arrays and tables with many classification, regression, and clustering algorithms to train models on data sets that do not fit in memory without changing your code.

Statistics and Machine Learning Toolbox™ contains a variety of functions that work with tall arrays. Tall arrays provide a convenient way to work with data that does not fit in memory, that is, the sample size can be arbitrarily large. To create a tall array, first create a datastore that references the data, and then use the tall function to convert the datastore into a tall array. For more information about tall arrays in MATLAB®, see Tall Arrays. For a list of supported statistics functions, see Function List (Tall Arrays).

If you have Parallel Computing Toolbox™, then the use of parallel computing can speed up certain statistical computations with tall arrays. To use parallel computing with tall arrays, see Extend Tall Arrays with Other Products.

Code Generation

Generate portable and readable C/C++ code for inference of classification and regression models, descriptive statistics, and probability distributions. Generate C/C++ prediction code with reduced precision, and update parameters of deployed models without regenerating the prediction code.

MATLAB® Coder™ generates readable and portable C and C++ code from Statistics and Machine Learning Toolbox functions that support code generation. For example, you can classify new

observations on hardware devices that cannot run MATLAB by deploying a trained support vector machine (SVM) classification model to the device using code generation.

You can generate C/C++ code for these functions in several ways:

- Use saveLearnerForCoder, loadLearnerForCoder, and codegen (MATLAB Coder) for an object function of a machine learning model.
- Use a coder configurer created by learnerCoderConfigurer for predict and update object functions of a machine learning model. Configure code generation options by using the configurer and update model parameters in the generated code.
- Use codegen for other functions that support code generation.

You can also generate fixed-point C/C++ code for the prediction of some machine learning models. This type of code generation requires Fixed-Point Designer™.

To integrate the prediction of a machine learning model into Simulink®, use a MATLAB Function block or the Simulink blocks in the Statistics and Machine Learning Toolbox library.

To learn about code generation, see Introduction to Code Generation.

For a list of functions that support code generation, see Function List (C/C++ Code Generation).

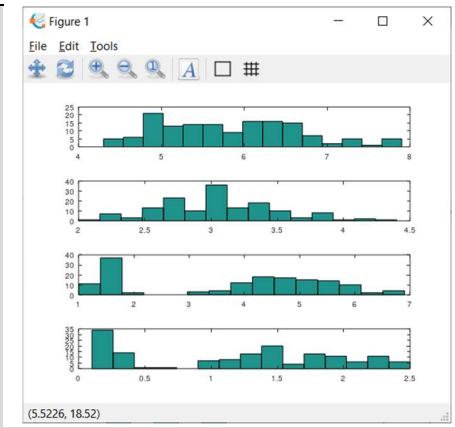
2. Αναζήτηση/κατέβασμα δεδομένων ταξινόμησης από online datasets όπως τα iris-data

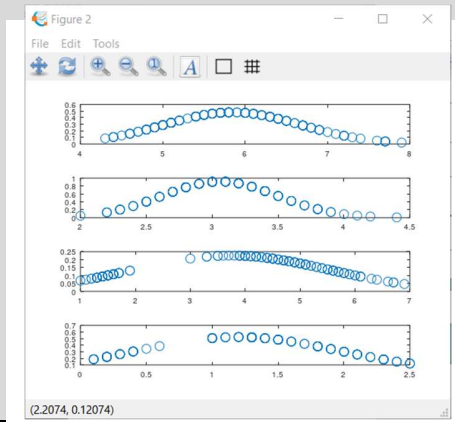
Α) Αναζητήστε στο Διαδίκτυο*: Το online dataset των δεδομένων ταξινόμησης Iris Data. Κατεβάστε τα στον Η/Υ και δώστε μια σύντομη περιγραφή και απεικονίστε τα σε διάγραμμα*

```
%eisagwgh iris dataset:
fid = fopen('iris.data');  #prosoxi edw vazoume to absolute path tou iris.data ston diko mas upologisti
iris_data = textscan(fid, '%f %f %f %f %s', 200, 'Delimiter', ',');
fclose(fid);
irisMatrix = cell2mat(iris_data(:,1:4));


%afairoume thn teleutaia grammh se periptwsh pou einai NaN
irisMatrix(end,:) = [];
disp(irisMatrix);


%statistikoi upologismoi (mesh timh, tupikh apoklish)


mu_vector = mean(irisMatrix); %epistrefei ena dianusma me tis 4 meses times twn sthlwn
printf("\n");
```

```
%disp(mu_vector);


%apo8hkeuoume kathe mesh timh se ksexwristh metavlhth
mu_sl = mu_vector(1);
mu_sw = mu_vector(2);
mu_pl = mu_vector(3);
mu_pw = mu_vector(4);
printf("Sepal Lenght mean = %d\n", mu_sl);
printf("Sepal Width mean = %d\n", mu_sw);
printf("Petal Lenght mean = %d\n", mu_pl);
printf("Petal Width mean = %d\n", mu_pw);


std_vector = std(irisMatrix); %epistrefei ena dianusma me 4 tupikes apokliseis (twn sthlwn)
disp(std_vector);
printf("\n");
%apo8hkeuoume kathe tupikh apoklish se ksexwristh metavlhth
std_sl = std_vector(1);
std_sw = std_vector(2);
std_pl = std_vector(3);
std_pw = std_vector(4);
printf("Sepal Lenght std = %d\n", std_sl);
printf("Sepal Width std = %d\n", std_sw);
printf("Petal Lenght std = %d\n", std_pl);
printf("Petal Width std = %d\n", std_pw);



%h subplot mas epitrepei na exoume polles diaforetikes sunarthseis ektupwmenes se ena figure
%typwnoume ta istogrammata twn 4 sthlwn tou dataset
```

```
subplot(4,1,1)
hist(irisMatrix(:,1), 15);
subplot(4,1,2)
hist(irisMatrix(:,2), 15);
subplot(4,1,3);
hist(irisMatrix(:,3), 15);
subplot(4,1,4);
hist(irisMatrix(:,4), 15);

figure(2);

%kanoyme fit tis 4 sthles twn metavlhtwn se mia kanonikh katanomh Gauss

subplot(4,1,1)
f1 = exp(-(irisMatrix(:,1)-mu_sl).^2./(2*std_sl^2))./(std_sl*sqrt(2*pi));
plot(irisMatrix(:,1), f1, 'o');

subplot(4,1,2)
f2 = exp(-(irisMatrix(:,2)-mu_sw).^2./(2*std_sw^2))./(std_sw*sqrt(2*pi));
plot(irisMatrix(:,2), f2, 'o');

subplot(4,1,3)
f3 = exp(-(irisMatrix(:,3)-mu_pl).^2./(2*std_pl^2))./(std_pl*sqrt(2*pi));
plot(irisMatrix(:,3), f3, 'o');

subplot(4,1,4)
f4 = exp(-(irisMatrix(:,4)-mu_pw).^2./(2*std_pw^2))./(std_pw*sqrt(2*pi));
plot(irisMatrix(:,4), f4, 'o');



%dhmiourgoume ton pinaka kai ton suntelesth susxetishs
figure(3);
plotmatrix(irisMatrix);
rho = corr(irisMatrix);
disp(rho);
```





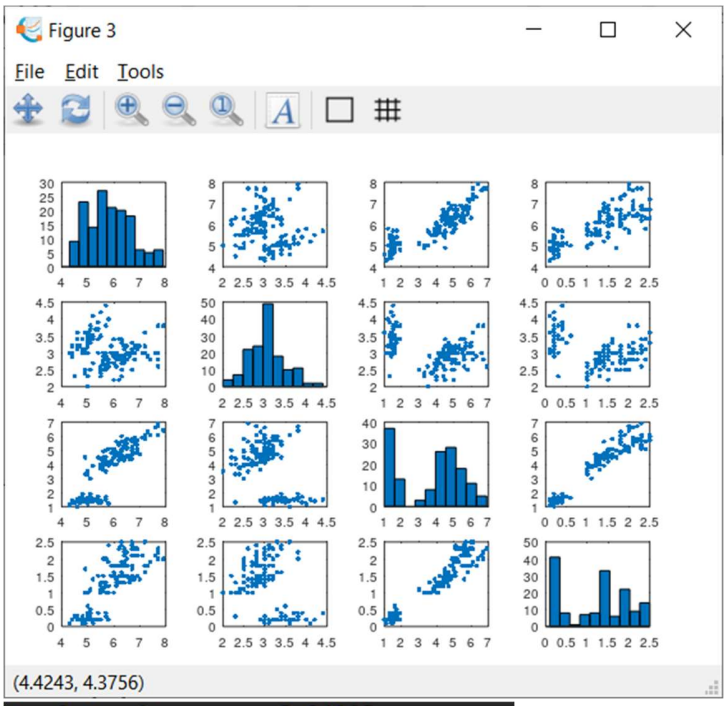
3. Προ-επεξεργασία & εξαγωγή χαρακτηριστικών

Υλοποιείστε το Παράδειγμα της 1.4 χρησιμοποιώντας τα δεδομένα iris-data

```
%eisagwgh iris dataset:

fid = fopen('iris.data');  #prosoxi edw vazoume to absolute path tou iris.data ston diko mas upologisti

iris_data = textscan(fid, '%f %f %f %f %s', 200, 'Delimiter', ',');

fclose(fid);

irisMatrix = cell2mat(iris_data(:,1:4));


%afairoume thn teleutaia grammh se periptwsh pou einai NaN

irisMatrix(end,:) = [];

disp(irisMatrix);


%statistikoi upologismoi (mesh timh, tupikh apoklish)


mu_vector = mean(irisMatrix); %epistrefei ena dianusma me tis 4 meses times twn sthlwn

printf("\n");

%disp(mu_vector);


%apo8hkeuoume kathe mesh timh se ksexwristh metavlhth

mu_sl = mu_vector(1);

mu_sw = mu_vector(2);

mu_pl = mu_vector(3);

mu_pw = mu_vector(4);

printf("Sepal Lenght mean = %d\n", mu_sl);

printf("Sepal Width mean = %d\n", mu_sw);

printf("Petal Lenght mean = %d\n", mu_pl);

printf("Petal Width mean = %d\n", mu_pw);


std_vector = std(irisMatrix); %epistrefei ena dianusma me 4 tupikes apokliseis (twn sthlwn)

disp(std_vector);

printf("\n");

%apo8hkeuoume kathe tupikh apoklish se ksexwristh metavlhth

std_sl = std_vector(1);

std_sw = std_vector(2);

std_pl = std_vector(3);

std_pw = std_vector(4);
```

```
printf("Sepal Lenght std = %d\n", std_sl);
printf("Sepal Width std = %d\n", std_sw);
printf("Petal Lenght std = %d\n", std_pl);
printf("Petal Width std = %d\n", std_pw);



%h subplot mas epitrepei na exoume polles diaforetikes sunarthseis ektupwmenes se ena figure
%typwnoume ta istogrammata twn 4 sthlwn tou dataset
subplot(4,1,1)
hist(irisMatrix(:,1), 15);
subplot(4,1,2)
hist(irisMatrix(:,2), 15);
subplot(4,1,3);
hist(irisMatrix(:,3), 15);
subplot(4,1,4);
hist(irisMatrix(:,4), 15);

figure(2);

%kanoyme fit tis 4 sthles twn metavlhtwn se mia kanonikh katanomh Gauss

subplot(4,1,1)
f1 = exp(-(irisMatrix(:,1)-mu_sl).^2./(2*std_sl^2))./(std_sl*sqrt(2*pi));
plot(irisMatrix(:,1), f1, 'o');

subplot(4,1,2)
f2 = exp(-(irisMatrix(:,2)-mu_sw).^2./(2*std_sw^2))./(std_sw*sqrt(2*pi));
plot(irisMatrix(:,2), f2, 'o');

subplot(4,1,3)
f3 = exp(-(irisMatrix(:,3)-mu_pl).^2./(2*std_pl^2))./(std_pl*sqrt(2*pi));
plot(irisMatrix(:,3), f3, 'o');

subplot(4,1,4)
f4 = exp(-(irisMatrix(:,4)-mu_pw).^2./(2*std_pw^2))./(std_pw*sqrt(2*pi));
plot(irisMatrix(:,4), f4, 'o');
```

```
%dhmiourgoume ton pinaka kai ton suntelesth susxetishs
figure(3);
plotmatrix(irisMatrix);
rho = corr(irisMatrix);
disp(rho);
```

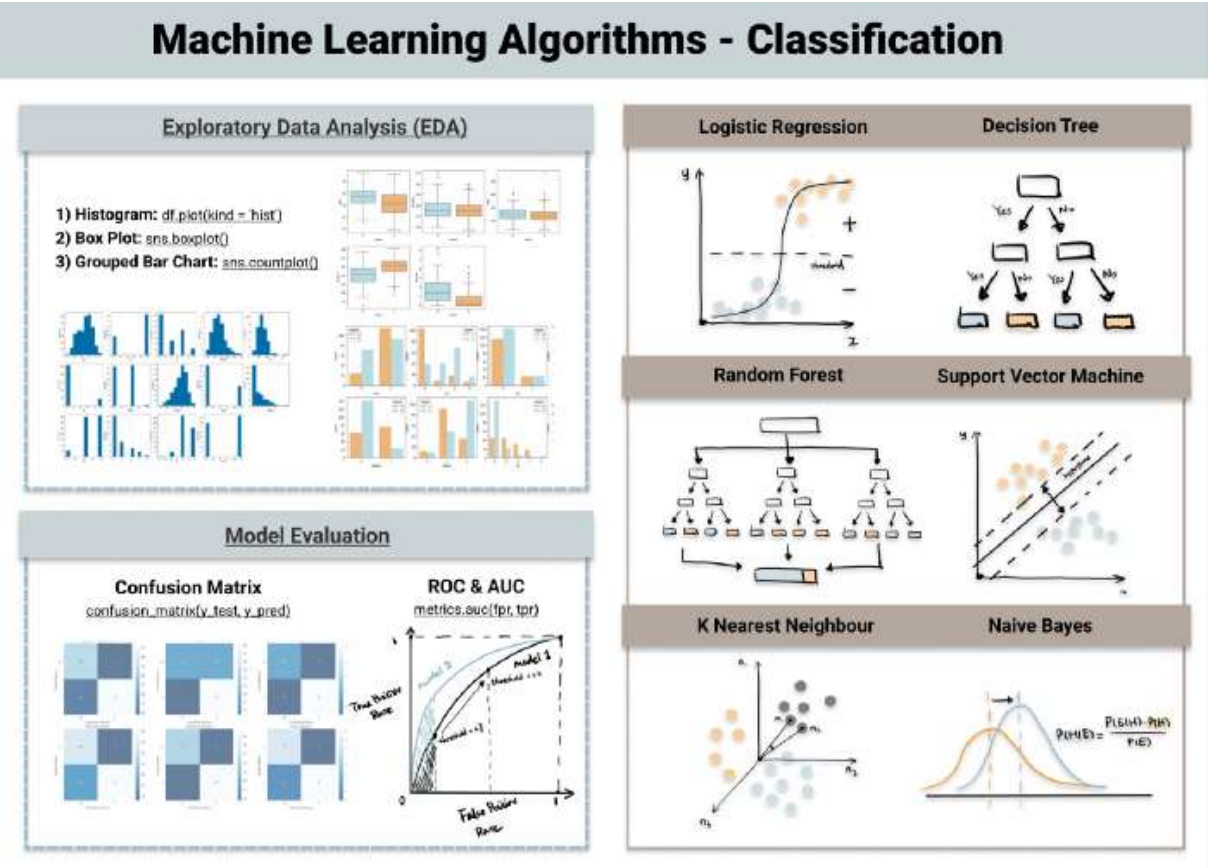



```
Sepal Lenght mean = 5.84295
Sepal Width mean = 3.05772
Petal Lenght mean = 3.74899
Petal Width mean = 1.1953
   0.8308   0.4373   1.7678   0.7632

Sepal Lenght std = 0.830846
Sepal Width std = 0.437311
Petal Lenght std = 1.76779
Petal Width std = 0.763202
   1.0000  -0.1175   0.8731   0.8193
  -0.1175   1.0000  -0.4286  -0.3662
   0.8731  -0.4286   1.0000   0.9627
   0.8193  -0.3662   0.9627   1.0000
>>
```

4. Αναζήτηση μεθόδων ταξινόμησης και σύντομη παρουσίαση - Εργαλείων Ταξινόμησης των MatLab/Octave

Α) Αναζητήστε στο Διαδίκτυο*: Μεθόδους Ταξινόμησης και βρείτε τα αντίστοιχα* εργαλεία των MatLab/Octave. Δώστε μια σύντομη περιγραφή κάθε μεθόδου και τον τρόπο χρήσης και απεικονίστε τα σε διάγραμμα

Top 6 Machine Learning Algorithms for Classification



Supervised vs. Unsupervised vs. Reinforcement Learning

The easiest way to distinguish a supervised learning and unsupervised learning is to see whether the data is labelled or not.

Supervised learning learns a function to make prediction of a defined label based on the input data. It can be either classifying data into a category (classification problem) or forecasting an outcome (regression algorithms).

Unsupervised learning reveals the underlying pattern in the dataset that are not explicitly presented, which can discover the similarity of data points (clustering algorithms) or uncover hidden relationships of variables (association rule algorithms) …

Reinforcement learning is another type of machine learning, where the agents learn to take actions based on its interaction with the environment, with the aim to maximize rewards. It is most similar to the learning process of human, following a trial-and-error approach.

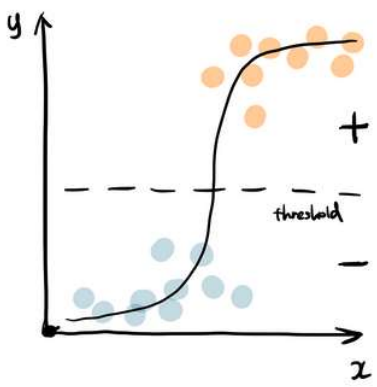
Classification vs Regression

Supervised learning can be furthered categorized into classification and regression algorithms. **Classification model** identifies which category an object belongs to whereas **regression model** predicts a continuous output.

Sometimes there is an ambiguous line between classification algorithms and regression algorithms. Many algorithms can be used for both classification and regression, and classification is just regression model with a threshold applied. When the number is higher than the threshold it is classified as true while lower classified as false.

In this article, we will discuss top 6 machine learning algorithms for classification problems, including: l*ogistic regression, decision tree, random forest, support vector machine, k nearest neighbour and naive bayes*. I summarized the theory behind each as well as how to implement each using python. Check out the code for model pipeline on my website.

1. Logistic Regression



Logistics regression uses sigmoid function above to return the probability of a label. It is widely used when the classification problem is binary — true or false, win or lose, positive or negative ...

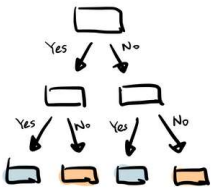
The sigmoid function generates a probability output. By comparing the probability with a pre-defined threshold, the object is assigned to a label accordingly. Check out my posts on logistic regression for a detailed walkthrough.

Below is the code snippet for a default logistic regression and the common hyperparameters to experiment on — see which combinations bring the best result.

```
from sklearn.linear_model import LogisticRegression
reg = LogisticRegression()
reg.fit(X_train, y_train)
y_pred = reg.predict(X_test)
```

logistic regression common hyperparameters: penalty, max_iter, C, solver

2. Decision Tree

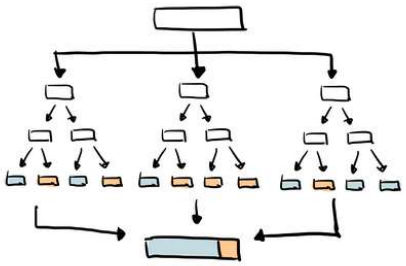


Decision tree builds tree branches in a hierarchy approach and each branch can be considered as an if-else statement. The branches develop by partitioning the dataset into subsets based on most important features. Final classification happens at the leaves of the decision tree.

```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)
y_pred = dtc.predict(X_test)
```

decision tree common hyperparameters: criterion, max_depth, min_samples_split, min_samples_leaf; max_features

3. Random Forest

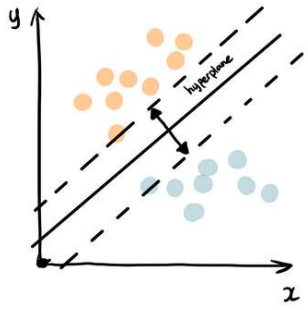


As the name suggest, random forest is a collection of decision trees. It is a common type of ensemble methods which aggregate results from multiple predictors. Random forest additionally utilizes bagging technique that allows each tree trained on a random sampling of original dataset and takes the majority vote from trees. Compared to decision tree, it has better generalization but less interpretable, because of more layers added to the model.

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(X_train, y_train)
y_pred = rfc.predict(X_test)
```

random forest common hyperparameters: n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf, boostrap

4. Support Vector Machine (SVM)

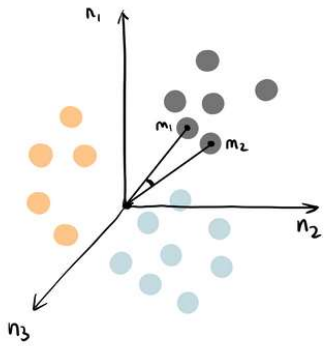


Support vector machine finds the best way to classify the data based on the position in relation to a border between positive class and negative class. This border is known as the hyperplane which maximize the distance between data points from different classes. Similar to decision tree and random forest, support vector machine can be used in both classification and regression, SVC (support vector classifier) is for classification problem.

```
from sklearn.svm import SVC
svc = SVC()
svc.fit(X_train, y_train)
y_pred = svc.predict(X_test)
```

support vector machine common hyperparameters: c, kernel, gamma

5. K-Nearest Neighbour (KNN)

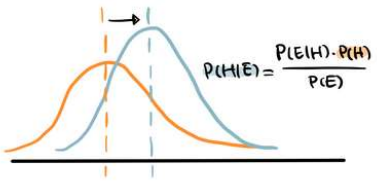


You can think of k nearest neighbour algorithm as representing each data point in a n dimensional space — which is defined by n features. And it calculates the distance between one point to another, then assign the label of unobserved data based on the labels of nearest observed data points. KNN can also be used for building recommendation system, check out my article on "Collaborative Filtering for Movie Recommendation" if you are interested in this topic.

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
```

KNN common hyperparameters: n_neighbors, weights, leaf_size, p

6. Naive Bayes



Naive Bayes is based on Bayes' Theorem — an approach to calculate conditional probability based on prior knowledge, and the naive assumption that each feature is independent to each other. The biggest advantage of Naive Bayes is that, while most machine learning algorithms rely on large amount of training data, it performs relatively well even when the training data size is small. Gaussian Naive Bayes is a type of Naive Bayes classifier that follows the normal distribution.

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
```

gaussian naive bayes common hyperparameters: priors, var_smoothing

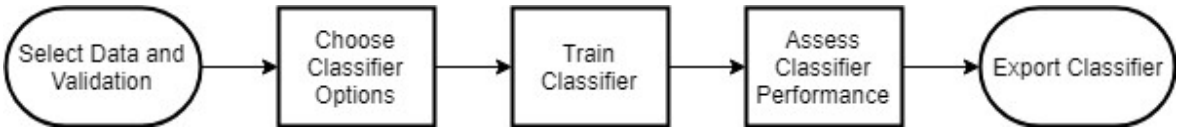
Classification Learner

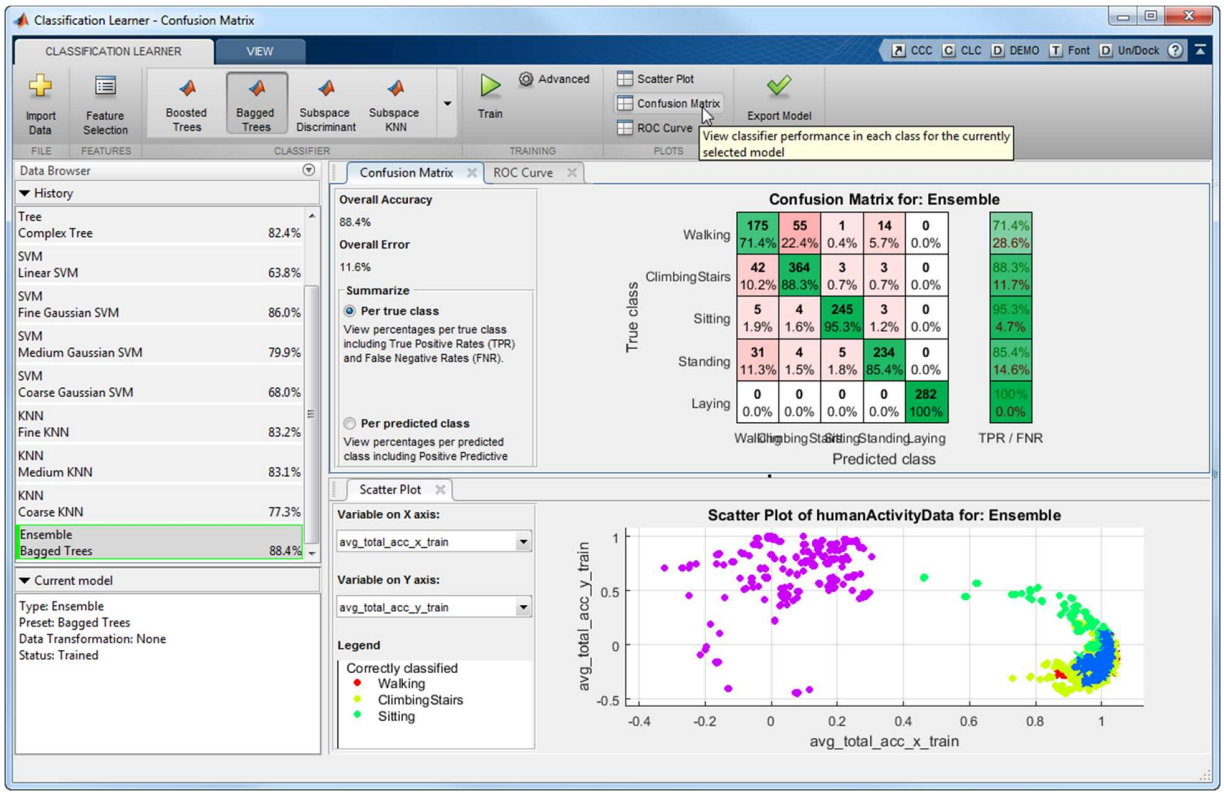
Classification Learner App

Interactively train, validate, and tune classification models

Choose among various algorithms to train and validate classification models for binary or multiclass problems. After training multiple models, compare their validation errors side-by-side, and then choose the best model. To help you decide which algorithm to use, see Train Classification Models in Classification Learner App.

This flow chart shows a common workflow for training classification models, or classifiers, in the Classification Learner app.





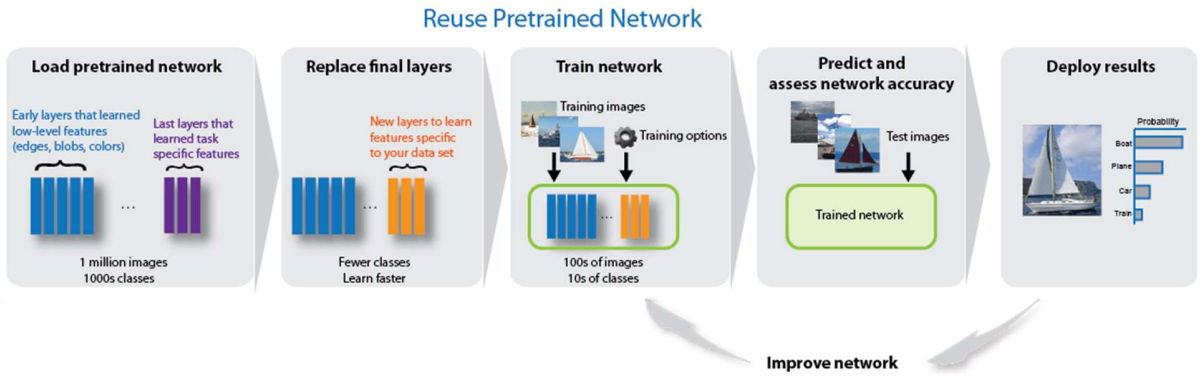
Deep Network Designer

Pretrained Networks for Images

Use pretrained networks to quickly learn new tasks

Use transfer learning to take advantage of the knowledge provided by a pretrained network to learn new patterns in new data. Fine-tuning a pretrained image classification network with transfer learning is typically much faster and easier than training from scratch. Using pretrained deep

networks enables you to quickly create models for new tasks without defining and training a new network, having millions of images, or having a powerful GPU.

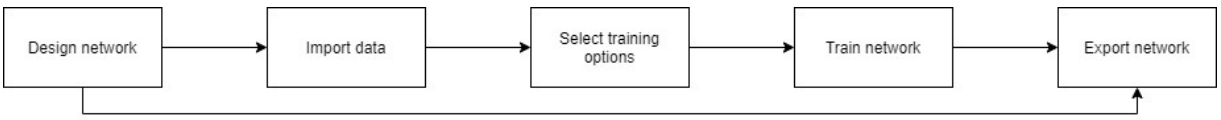


Deep Network Designer App

Interactively create and train deep learning networks

Build, visualize, and train deep learning networks interactively. Use the start page to load pretrained image classification networks suitable for transfer learning. Analyze your network to check that you have defined the architecture correctly and detect problems before training. Import and visualize training data, specify training options, and track network training using animated plots of training progress. Generate code to recreate network construction and training, and export trained networks to Simulink®.

This flow chart shows common workflows for building and training deep learning models in the **Deep Network Designer** app. You can train your network in **Deep Network Designer** or export your untrained network for training at the command line.



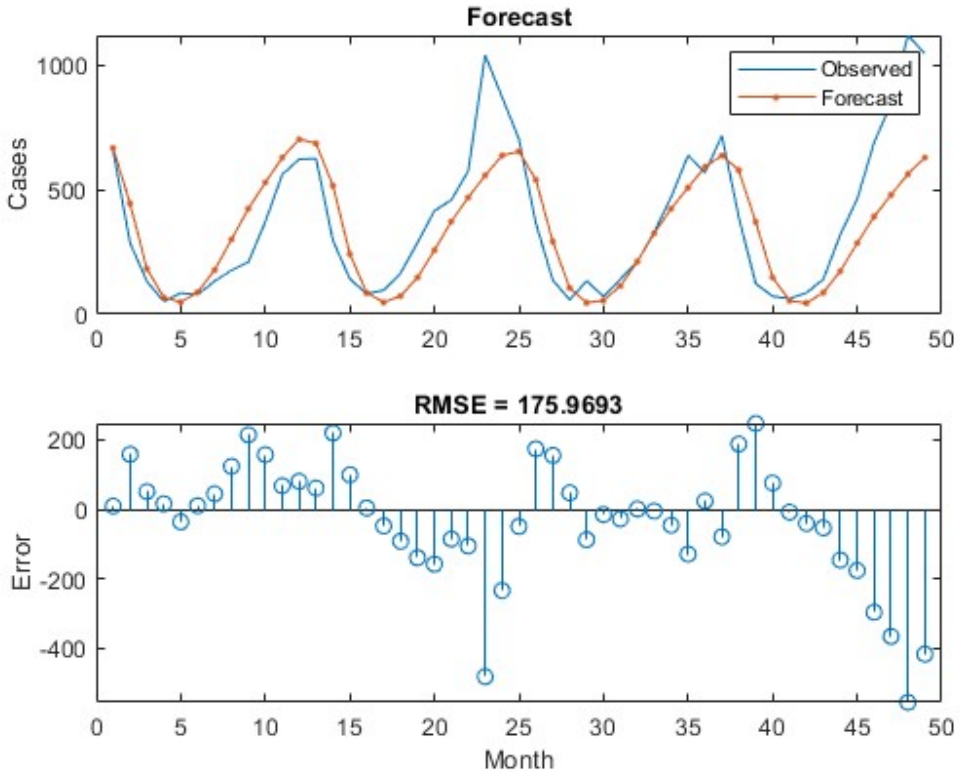
Train Network for Time Series Forecasting Using Deep Network Designer

This example shows how to forecast time series data by training a long short-term memory (LSTM) network in **Deep Network Designer**.

Deep Network Designer allows you to interactively create and train deep neural networks for sequence classification and regression tasks.

To forecast the values of future time steps of a sequence, you can train a sequence-to-sequence regression LSTM network, where the responses are the training sequences with values shifted by one time step. That is, at each time step of the input sequence, the LSTM network learns to predict the value of the next time step.

This example uses the data set chickenpox_dataset. The example creates and trains an LSTM network to forecast the number of chickenpox cases given the number of cases in previous months.



Experiment manager

Experiment Manager App

Train networks under multiple initial conditions, interactively tune training options, and assess your results

Find optimal training options for neural networks by sweeping through a range of hyperparameter values or using Bayesian optimization. Use the built-in function trainNetwork or define your own custom training function. Test different training configurations at the same time by running your experiment in parallel. Monitor your progress by using training plots. Use confusion matrices and custom metric functions to evaluate your trained network. Refine your experiments by sorting and filtering. Use annotations to record your observations.

Create a Deep Learning Experiment for Classification

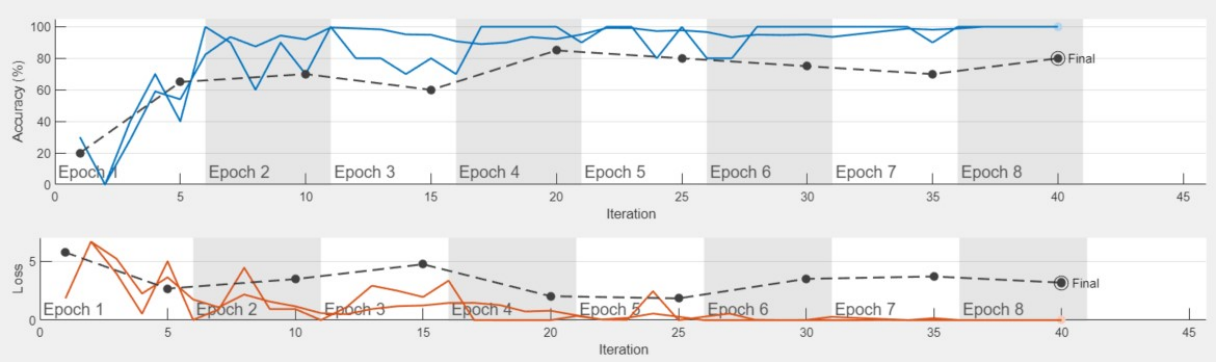
This example uses:

- Deep Learning Toolbox Model for GoogLeNet Network
- Deep Learning Toolbox

This example shows how to train a deep learning network for classification by using **Experiment Manager**. In this example, you train two networks to classify images of MathWorks merchandise into five classes. Each network is trained using three algorithms. In each case, a confusion matrix

compares the true classes for a set of validation images with the classes predicted by the trained network. For more information on training a network for image classification, see Train Deep Learning Network to Classify New Images.

This experiment requires the Deep Learning Toolbox™ Model *for GoogLeNet Network* support package. Before you run the experiment, install this support package by calling the googlenet function and clicking the download link.



Neural Net Pattern Recognition

Solve pattern recognition problem using two-layer feed-forward networks

Description

The **Neural Net Pattern Recognition** app lets you create, visualize, and train two-layer feed-forward networks to solve data classification problems.

Using this app, you can:

- Import data from file, the MATLAB® workspace, or use one of the example data sets.
- Split data into training, validation, and test sets.
- Define and train a neural network.
- Evaluate network performance using cross-entropy error and misclassification error.
- Analyze results using visualization plots, such as confusion matrices and receiver operating characteristic curves.
- Generate MATLAB scripts to reproduce results and customize the training process.
- Generate functions suitable for deployment with MATLAB Compiler™ and MATLAB Coder™ tools, and export to Simulink® for use with Simulink Coder.

Sequence Classification Using Deep Learning

This example shows how to classify sequence data using a long short-term memory (LSTM) network.

To train a deep neural network to classify sequence data, you can use an LSTM network. An LSTM network enables you to input sequence data into a network, and make predictions based on the individual time steps of the sequence data.

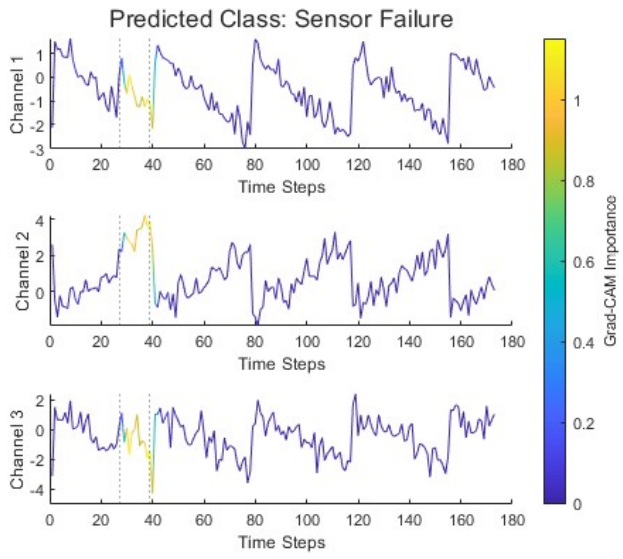
This example uses the Japanese Vowels data set as described in [1] and [2]. This example trains an LSTM network to recognize the speaker given time series data representing two Japanese vowels spoken in succession. The training data contains time series data for nine speakers. Each sequence has 12 features and varies in length. The data set contains 270 training observations and 370 test observations.

Neural Net Time Series

Interpret Deep Learning Time-Series Classifications Using Grad-CAM

This example shows how to use the gradient-weighted class activation mapping (Grad-CAM) technique to understand the classification decisions of a 1-D convolutional neural network trained on time-series data.

Grad-CAM [1] uses the gradient of the classification score with respect to the convolutional features determined by the network to understand which parts of the data are most important for classification. For time-series data, Grad-CAM computes the most important time steps for the classification decision of the network.



This image shows an example sequence with a Grad-CAM importance colormap. The map highlights the regions the network uses to make the classification decision.

This example uses supervised learning on labeled data to classify time-series data as "Normal" or "Sensor Failure". You can also use an autoencoder network to perform time-series anomaly detection on unlabeled data. For more information, see Time Series Anomaly Detection Using Deep Learning.

5. Αναζήτηση Εφαρμογών ταξινόμησης στην ειδικότητα

Α) Αναζητήστε στο Διαδίκτυο*: Προηγμένες εφαρμογές Ταξινόμησης, συναφείς με την ειδικότητα σας και παρουσιάστε μια σύντομη περιγραφή τους.*

Multi Class Object Classification for Retail Products

Computer vision for friction-less store experience.

Retail has never been a stagnant industry. Retailers just can not afford to stand still if they want to succeed. They must adapt and innovate or risk being left behind.

The application of computer vision in retail is set to fundamentally change the shopping experience for customers and retailers. In this blog I will be making a computer vision based multi class object classification model for retail products. This project has been inspired from the famous Amazon Go store. I hope you enjoy this article and find it insightful.

Amazon describes Amazon Go as a new kind of store with no checkout required. That means when you shop at Amazon Go, you'll never have to wait in line. The store works with an Amazon Go application — you enter Amazon Go, take the products you want and thanks to the app just leave again. It works by using the same types of technologies found in self-driving cars, such as computer vision, sensor fusion and deep learning. This technology can detect when products are taken or returned to the shelves and keeps track of them in your virtual cart. When you leave the store with your goods, your Amazon account is charged and you are sent a receipt.

Applications of Computer Vision in Retail

1. Customizing experiences using facial recognition.
2. Making shopping feel more human and less transactional.
3. Computer Vision based inventory management.
4. Blurring the line between in-store and online.
5. The friction-less store experience.

Data

I have used Freiburg Groceries Dataset for this project. It consists of 5000 256x256 RGB images of 25 food classes. The paper can be found here and the dataset here.

Environment and tools

1. Numpy
2. Pandas
3. Scikit-image
4. Matplotlib
5. Scikit-learn
6. Keras

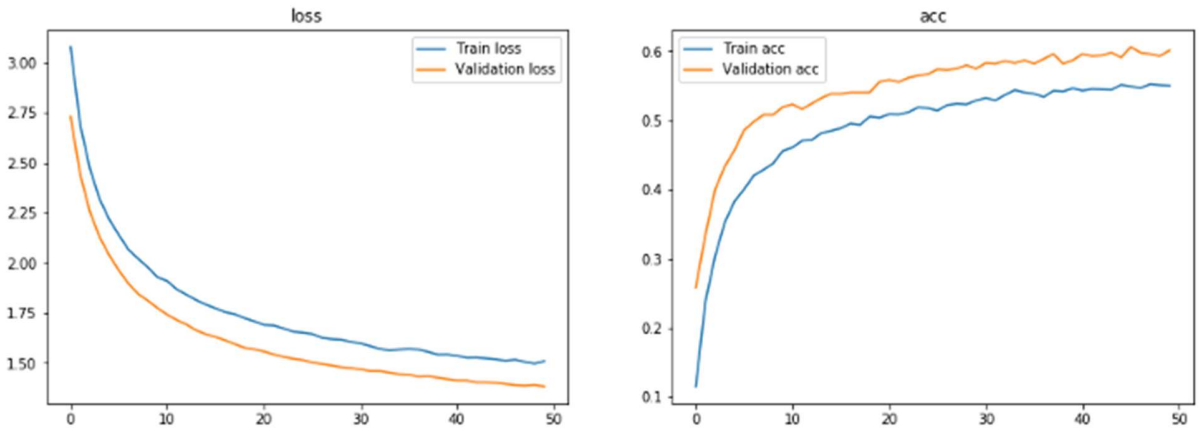
Image Classification

The complete image classification pipeline can be formalized as follows:

- Our input is a training dataset that consists of *N* images, each labeled with one of 25 different classes.
- Then, we use this training set to train a classifier to learn what every one of the classes looks like.
- In the end, we evaluate the quality of the classifier by asking it to predict labels for a new set of images that it has never seen before. We will then compare the true labels of these images to the ones predicted by the classifier.

Results

Loss/Accuracy vs Epoch



Loss/Accuracy vs Epoch

The model is able to reach a validation accuracy of 60% which is quite good considering the number of classes(25) with 100–200 images in each category. Feel free to use different architectures or play with the hyper-parameters for better results.

https://towardsdatascience.com/multi-class-object-classification-for-retail-products-aa4ecaaaa096

6 .Ταξινόμηση & λήψη αποφάσεων

Α) Αναζητήστε στο Διαδίκτυο*: Μεθόδους Λήψης Αποφάσεων και βρείτε τα αντίστοιχα εργαλεία των MatLab/Octave. Δώστε μια σύντομη περιγραφή κάθε μεθόδου και τον τρόπο χρήσης και απεικονίστε τα σε διάγραμμα*

THE FOUR CATEGORIES OF DECISION MAKING

BUSINESS LEADERS AND THE COMPLEXITY BEHIND STRATEGIC DECISIONS

Decisions are a part of life and while they range in complexity, we face various decisions on a daily basis. But how can we best make decisions that lead to optimal outcomes? With an ever-

growing wealth of research on the topic, decision making is being transformed into a science that can aid greatly in guiding decisions. However, undoubtedly useful in certain circumstances, the application of such research highly depends on the kind of decisions being made, especially in the business world.

There are clear limits and, to date, most decision making research applies to one type of decision, and it's not the type that's most challenging for managers. Their most important and most difficult decisions are strategic decisions with consequences for the performance of the company. Before managers take any advice on how to make better decisions, they must recognize how decisions differ.

Decisions vary along two dimensions: control and performance. Control considers how much we can influence the terms of the decision and the outcome. And performance addresses the way we measure success. Combining them creates four categories of decisions:

1] Making routine choices and judgments. When you go shopping in a supermarket or a department store, you typically pick from the products before you. Those items, perhaps a jug of milk or a jar of jam, are what they are. You have no ability to improve them. Control is low. Moreover, you make the choice that suits you best—it doesn't matter what anyone else is buying. Performance is absolute. The same goes for most personal investment decisions. You may be able to decide which company's shares to buy, but you can't improve their performance after you buy them. You want high returns but aren't trying to do better than others. The goal is to do well, not to finish first in a competition.

2] Influencing outcomes. Many decisions involve more than selecting among options we cannot improve or making judgments about things we cannot influence. In so much of life, we use our energy and talents to make things happen. Imagine that the task at hand is to determine how long we will need to complete a project. That's a judgment we can control; indeed, it's up to us to get the project done. Here, positive thinking matters. By believing we can do well, perhaps even holding a level of confidence that is by some definitions a bit excessive, we can often improve performance. Optimism isn't useful in picking stocks whose performance we cannot change, but in the second field, where we have the ability to influence outcomes, it can be very important.

3] Placing competitive bets. The third category introduces a competitive dimension. Success is no longer a matter of absolute performance but depends on how well you do relative to others. The best decisions must anticipate the moves of rivals. That's the essence of strategic thinking, which Princeton professor Avinash Dixit and Yale professor Barry Nalebuff define as "the art of outdoing an adversary, knowing that the adversary is trying to do the same to you." Investments in stocks are typically first-field decisions, but if you're taking part in a contest where the investor with the highest return takes the prize, you're in the third field. Now you need to make decisions with an eye to what your rivals will do, anticipating their likely moves so that you can have the best chance of winning.

4] Making strategic decisions. In this fourth category of decision making, we can actively influence outcomes and success means doing better than rivals. Here we find the essence of strategic management. Business executives aren't like shoppers picking a product or investors

choosing a stock, simply making a choice that leads to one outcome or another. By the way they lead and communicate, and through their ability to inspire and encourage, executives can influence outcomes. That's the definition of "management." Moreover, they are in charge of organizations that compete vigorously with others; doing better than rivals is vital. That's where strategy comes in.

The decisions to enter a new market, release a new product, or acquire another firm are all in the fourth field, but we can find many examples beyond business. In sports, a coach shapes the performance of athletes, melding them into an effective team that can outperform the opponent. Or in politics a winning political campaign depends on a smart assessment of rivals as well as the ability to mobilize supporters, often in the face of long odds.

The constraint of decision making research

In the course of their daily responsibilities, executives face a range of decisions, often in each of the four fields outlined here. But decision making research cannot be universally applied. When turning to research to help make better decisions, it's critical to understand these four fields of decision making.

For first- and second-field decisions, cognitive psychologists have demonstrated that people make decisions in ways that do not conform to the tenets of economic rationality. They exhibit systematic biases, which, if surfaced, managers could use to shape routine decisions. For third-field decisions, guidance comes from the branch of economics that studies competitive dynamics: game theory. Game theory can illuminate areas from price competition to geopolitics, yet it has an important limitation: Players cannot alter the terms of the game.

So while a great deal of attention has focused on teaching executives to excel at game theory and to be aware of common biases in order to avoid their ill effects, if we apply only those lessons to the world of strategic management, we're missing a trick.

The fourth field of decision making has a layer of complexity that doesn't lend itself to the careful controls of laboratory experiments, so we know less about how best to make them. What sort of mind-set do they require? When we can influence outcomes, it is useful to summon high levels of self-belief. And when we need to outperform rivals, such elevated levels are not just useful but indeed essential. Only those who are able to muster a degree of commitment and determination that is by some definitions excessive will be in a position to win.

https://www.imd.org/research-knowledge/articles/the-four-categories-of-decision-making/

Classification Learning

Multilabel Image Classification Using Deep Learning

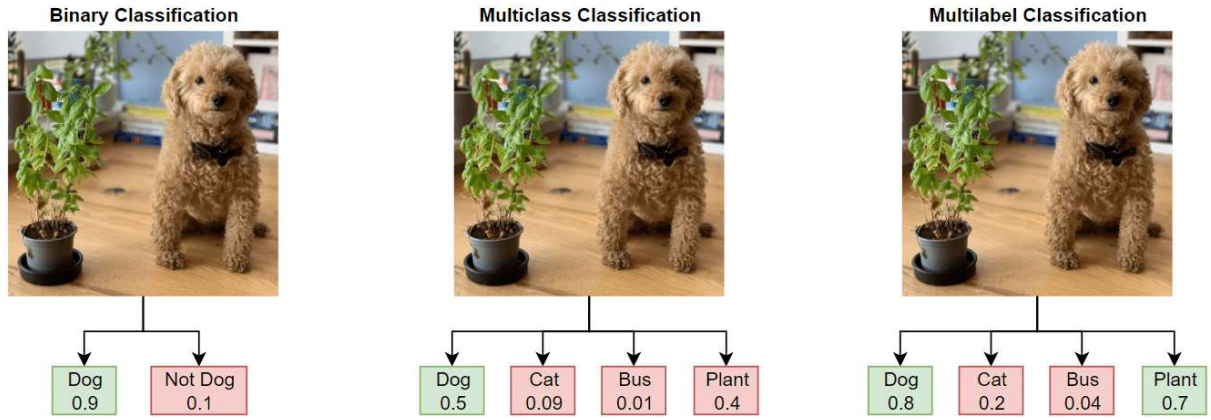
This example shows how to use transfer learning to train a deep learning model for multilabel image classification.

In binary or multiclass classification, a deep learning model classifies images as belonging to one of two or more classes. The data used to train the network often contains clear and focused images,

with a single item in frame and without background noise or clutter. This data is often not an accurate representation of the type of data the network will receive during deployment. Additionally, binary and multiclass classification can apply only a single label to each image, leading to incorrect or misleading labeling.

In this example, you train a deep learning model for multilabel image classification by using the COCO data set, which is a realistic data set containing objects in their natural environments. The COCO images have multiple labels, so an image depicting a dog and a cat has two labels.

In multilabel classification, in contrast to binary and multiclass classification, the deep learning model predicts the probability of each class. The model has multiple independent binary classifiers, one for each class—for example, "Cat" and "Not Cat" and "Dog" and "Not Dog."



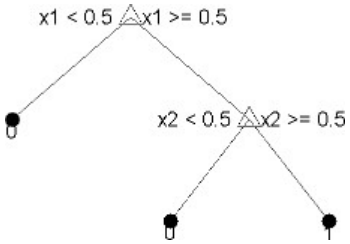
Regression Trees — Apps

Supervised Learning Workflow and Algorithms

Decision Trees

Decision trees, or classification trees and regression trees, predict responses to data. To predict a response, follow the decisions in the tree from the root (beginning) node down to a leaf node. The leaf node contains the response. Classification trees give responses that are nominal, such as 'true' or 'false'. Regression trees give numeric responses.

Statistics and Machine Learning Toolbox™ trees are binary. Each step in a prediction involves checking the value of one predictor (variable). For example, here is a simple classification tree:



This tree predicts classifications based on two predictors, x1 and x2. To predict, start at the top node, represented by a triangle (Δ). The first decision is whether x1 is smaller than 0.5. If so, follow the left branch, and see that the tree classifies the data as type 0.

If, however, x1 exceeds 0.5, then follow the right branch to the lower-right triangle node. Here the tree asks if x2 is smaller than 0.5. If so, then follow the left branch to see that the tree classifies the data as type 0. If not, then follow the right branch to see that the tree classifies the data as type 1.

Optimization

Optimization Toolbox

Solve linear, quadratic, conic, integer, and nonlinear optimization problems

Optimization Toolbox™ provides functions for finding parameters that minimize or maximize objectives while satisfying constraints. The toolbox includes solvers for linear programming (LP), mixed-integer linear programming (MILP), quadratic programming (QP), second-order cone programming (SOCP), nonlinear programming (NLP), constrained linear least squares, nonlinear least squares, and nonlinear equations.

You can define your optimization problem with functions and matrices or by specifying variable expressions that reflect the underlying mathematics. You can use automatic differentiation of objective and constraint functions for faster and more accurate solutions.

You can use the toolbox solvers to find optimal solutions to continuous and discrete problems, perform tradeoff analyses, and incorporate optimization methods into algorithms and applications. The toolbox lets you perform design optimization tasks, including parameter estimation, component selection, and parameter tuning. It enables you to find optimal solutions in applications such as portfolio optimization, energy management and trading, and production planning.

Traveling Salesman Problem: Solver-Based

This example shows how to use binary integer programming to solve the classic traveling salesman problem. This problem involves finding the shortest closed tour (path) through a set of stops (cities). In this case there are 200 stops, but you can easily change the nStops variable to get a different problem size. You'll solve the initial problem and see that the solution has subtours. This means the optimal solution found doesn't give one continuous path through all the points, but instead has several disconnected loops. You'll then use an iterative process of determining the subtours, adding constraints, and rerunning the optimization until the subtours are eliminated.

For the problem-based approach, see Traveling Salesman Problem: Problem-Based.

Problem Formulation

Formulate the traveling salesman problem for integer linear programming as follows:

- Generate all possible trips, meaning all distinct pairs of stops.
- Calculate the distance for each trip.
- The cost function to minimize is the sum of the trip distances for each trip in the tour.
- The decision variables are binary, and associated with each trip, where each 1 represents a trip that exists on the tour, and each 0 represents a trip that is not on the tour.
- To ensure that the tour includes every stop, include the linear constraint that each stop is on exactly two trips. This means one arrival and one departure from the stop.

Decision Theory Toolbox (DTT)

The DTT implements some of the most popular and applied algorhythms originated from the Decision Theory Area:

- Multi Value Attribute Theory;
- Hierarchical Analysis
- Electre III

These methods have proved to be a valuable mean to help decision makers extract a rank from a pool of alternatives, each alternative representing a set of actions, on the basis of the qualitative/quantitative impact each alternative has on a manifoldness of indicators.

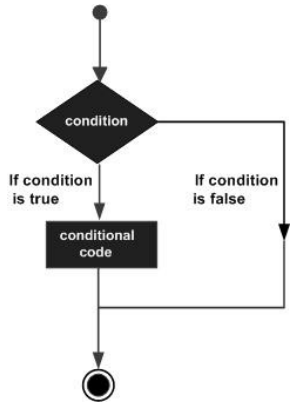
The functions developed fully emulates the methods above mentioned, handling exceptions arising from a particular input dataset.

The Multi Value Attribute Theory function (MAVT) lets you either directly provide the weight vector (a weight factor for each indicator) or interactively derive it through a succession of couples-preference test.

The Hierarchical Analysis function (HYEA) lets you define incomplete hierarchical structures on the input dataset.

The DT Toolbox contains a GUI demo, DTdemo, providing users with an overview of the capabilities offered by the functions herein implemented.

MATLAB - Decision Making



Decision making structures require that the programmer should specify one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

Following is the general form of a typical decision making structure found in most of the programming languages −

MATLAB provides following types of decision making statements

7. Εφαρμογές Ταξινόμησης και Αποφάσεων στην ειδικότητα

Α) Αναζητήστε στο Διαδίκτυο*: Προηγμένες εφαρμογές Λήψης Αποφάσεων, συναφείς με την ειδικότητα σας και παρουσιάστε μια σύντομη περιγραφή τους.*

Real-Time Decision-Making Use Cases in the Retail Industry – Part 2

In my last blog on Real-Time Decision-Making (RTDM), I shifted from a theoretical discussion of what RTDM is to uses cases that highlight key aspects of the value it can bring to business in periods of market uncertainty. I'm using the Retail Industry because its an Industry everyone knows something about coupled with the fact that its one of the sectors most impacted by our current business disruptor, COVID-19.

Further, I used Kiabi, a Global retailer headquartered in France, as the use case to delve into the importance of combining existing historical data patterns with disparate new sources of data to generate RTDM insights for a strategic capability. In this case, Kiabi's customer loyalty program. The key point I wanted you to take away from that example is that in times of market uncertainties, enhanced customer focus requires business agility that can only be achieved by a complete, common operational picture, and that requires additional pieces to the data puzzle.

Democratizing the Common Operational Picture

I briefly mentioned that the Common Operational Picture or COP has several stakeholders and users who may need to leverage the COP in different ways suited to their specific roles, responsibilities, and evolving needs during periods of instability. Making current and accurate information available to and between parties is critical and can make all the difference in service. Take, for example, the old days when you needed a taxi, you'd call the dispatch center, the center would send you a cab, and the cab dispatcher would call you when the cab arrived.

If you got lost and couldn't make it to the pick-up point, they wouldn't know. If the cabby got lost, you wouldn't know. If you were running late, they wouldn't know and just leave. Along came Uber and Lyft and, now you and the driver know precisely where you are at all times, and you have multiple means of immediate communication with them – the middleman is the app, and it delivers the COP. In other words, you have different roles – driver and fare-paying passenger – but you have a COP and a means of leveraging it to make decisions and act on them independent of any third party, in a peer-to-peer fashion.

In modern integrated online and brick-and-mortar Retail, you want a COP around stock availability and location (specific stores and online) for your customer-facing employees – online, in call centers, at stores – and customers. Under normal circumstances, it would be great to know before you make a trip to a store if the item you want to purchase is there. Now it's critical as we try to limit trips and their duration to reduce chances of infection rates going up with COVID-19. A less life-threatening but critical business requirement as well such that you don't risk frustrating customers who remain dissatisfied long after this period passes.

Decentralization of Decision Making

In periods of market uncertainty, the issues may be more complicated than is something in or out of stock, and historical data based on business-as-usual stock depletion and reordering patterns will only get you so far. You may need visibility into daily or even hourly buying patterns across your stores – and not just at the corporate level. As an individual store manager, you may even need visibility into your vendor's stock and the ability order from them or trade with other stores.

Actian helped another retailer, LeRoy Merlin, build out an RTDM strategic capability targeting their individual stores. The capability they wanted to enhance was the ability to monitor sales performance as a function of product SKU, time of purchase and quantity, stock on hand, and several other factors. LeRoy Merlin is a Do-It-Yourself home improvement retailer with over 400 stores in 12 countries across Europe, Asia, Africa, and South America. Their Enterprise data warehouse (EDW) was simply too slow to provide analytics services to high volumes of concurrent interactive users across tens of thousands of products. With Actian Vector (the engine inside the Actian Avalanche Cloud Data Warehouse), LeRoy Merlin was able to perform daily uploads from their EDW to provide real-time ad hoc queries and reporting by up to 2,000 interactive users. The intelligence from the Actian solution enables individual store managers to determine what's selling and what's sitting on the shelves so that they can adjust stock – critical during periods of rapid change in demand as we've seen with COVID-19.

But it doesn't have to be COVID-19, for example, DIY retailers in the US can tell where they will run out of stock on Generators by hurricane trajectory forecasts. The key point here is instead of a small group of decision-makers in HQ determining what stock orders should be made and sent where on a quarterly or annual basis or scaling that team up to reduce the hindsight view down to months or weeks. The RTDM capability provides LeRoy Merlin the ability to do it daily at the individual store manager level, decentralizing decision-making to improve speed, accuracy, and business agility.

Dynamic Pricing and Dynamic Rationing

As was the case with the RTDM strategic capability Actian supported for Kiabi, the support for LeRoy Merlin can be used for more than business agility to change stock and improve sales performance. It's not uncommon for store managers to have the latitude to discount items that aren't moving off the shelves. For this solution, the business-as-usual requirement we helped Kiabi with was speed. They couldn't get back real-time reports and queries to so many people at the same time. It was not an accuracy issue or in other terms, the freshness of the day you could use the sales performance data the Actian solution is providing would certainly help them make the right decision about this, but the data necessary to decide what to discount or what to reorder is probably not stale if it's refreshed on a weekly or even monthly basis.

In a period of market uncertainty that generates a rapid change in demand, you still need the real-time response, the speed, but you also need accuracy, current data. With panicked customers, you may need to reorder immediately, or you will see an empty shelf and, before your shelves empty of existing stock, you may need to adjust limits on purchasing quantities and even adjust those limits on a daily basis and let customers know before they make a trip to the store. Those limits may need to be different at different stores, again leveraging those closer to the transaction to make the decision.

Capability Reuse

You may even look to leverage the RTDM capability you've built across multiple core business processes. Everything we've just discussed could be combined with your loyalty program, and

you could send out notifications stating when new deliveries of toilet paper will arrive and what the purchasing quota will be, setting expectations in advance.

Finally, as I mentioned last time, RTDM capabilities are needed in almost any industry, and virtually all industries are impacted in some way by business disruptions. In the next blog, we'll start to gradually shift to a discussion of use cases in other industries and what it takes to build a world-class RTDM capability.

https://www.actian.com/blog/data-analytics/real-time-decision-making-use-cases-in-the-retail-industry-part-2/

compareHoldout

Compare accuracies of two classification models using new data

Description

compareHoldout statistically assesses the accuracies of two classification models. The function first compares their predicted labels against the true labels, and then it detects whether the difference between the misclassification rates is statistically significant.

You can determine whether the accuracies of the classification models differ or whether one model performs better than another. compareHoldout can conduct several McNemar test variations, including the asymptotic test, the exact-conditional test, and the mid-*p*-value test. For cost-sensitive assessment, available tests include a chi-square test (requires Optimization Toolbox™) and a likelihood ratio test.

8. Αναζήτηση Πινάκων Ενδεχομένων και επεξήγηση των δεικτών τους

Α) Αναζητήστε στο Διαδίκτυο*: Του διαφόρους τρόπους απεικόνισης των Πινάκων Ενδεχομένων (ή Σύγχυσης) και τις ποσότητες/δείκτες που προκύπτουν από αυτούς.*

Confusion matrix

In the field of machine learning and specifically the problem of statistical classification, a **confusion matrix**, also known as an error matrix,[10] is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a **matching matrix**). Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa – both variants are found in the literature.[11] The name stems from the fact that it makes it easy to see whether the system is confusing two classes (i.e. commonly mislabeling one as another).

It is a special kind of contingency table, with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions (each combination of dimension and class is a variable in the contingency table).

Example

Given a sample of 12 individuals, 8 that have been diagnosed with cancer and 4 that are cancer-free, where individuals with cancer belong to class 1 (positive) and non-cancer individuals belong to class 0 (negative), we can display that data as follows:

Individual Number	1	2	3	4	5	6	7	8	9	10	11	12
Actual Classification	1	1	1	1	1	1	1	1	0	0	0	0

Assume that we have a classifier that distinguishes between individuals with and without cancer in some way, we can take the 12 individuals and run them through the classifier. The classifier then makes 9 accurate predictions and misses 3: 2 individuals with cancer wrongly predicted as being cancer-free (sample 1 and 2), and 1 person without cancer that is wrongly predicted to have cancer (sample 9).

Individual Number	1	2	3	4	5	6	7	8	9	10	11	12
Actual Classification	1	1	1	1	1	1	1	1	0	0	0	0
Predicted Classification	0	0	1	1	1	1	1	1	1	0	0	0

Notice, that if we compare the actual classification set to the predicted classification set, there are 4 different outcomes that could result in any particular column. One, if the actual classification is positive and the predicted classification is positive (1,1), this is called a true positive result because the positive sample was correctly identified by the classifier. Two, if the actual classification is positive and the predicted classification is negative (1,0), this is called a false negative result because the positive sample is incorrectly identified by the classifier as being negative. Third, if the actual classification is negative and the predicted classification is positive (0,1), this is called a false positive result because the negative sample is incorrectly identified by the classifier as being positive. Fourth, if the actual classification is negative and the predicted classification is negative (0,0), this is called a true negative result because the negative sample gets correctly identified by the classifier.

We can then perform the comparison between actual and predicted classifications and add this information to the table, making correct results appear in green so they are more easily identifiable.

Individual Number	1	2	3	4	5	6	7	8	9	10	11	12
Actual Classification	1	1	1	1	1	1	1	1	0	0	0	0
Predicted Classification	0	0	1	1	1	1	1	1	1	0	0	0
Result	FN	FN	TP	TP	TP	TP	TP	TP	FP	TN	TN	TN

The template for any binary confusion matrix uses the four kinds of results discussed above (true positives, false negatives, false positives, and true negatives) along with the positive and negative classifications. The four outcomes can be formulated in a 2×2 *confusion matrix*, as follows:

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

The color convention of the three data tables above were picked to match this confusion matrix, in order to easily differentiate the data.

		Predicted condition	
	Total 8 + 4 = 12	Cancer 7	Non-cancer 5
Actual condition	Cancer 8	6	2
	Non-cancer 4	1	3

Now, we can simply total up each type of result, substitute into the template, and create a confusion matrix that will concisely summarize the results of testing the classifier:

Table of confusion

In predictive analytics, a **table of confusion** (sometimes also called a **confusion matrix**) is a table with two rows and two columns that reports the number of *true positives*, *false negatives*, *false positives*, and *true negatives*. This allows more detailed analysis than simply observing the proportion of correct classifications (accuracy). Accuracy will yield misleading results if the data set is unbalanced; that is, when the numbers of observations in different classes vary greatly.

For example, if there were 95 cancer samples and only 5 non-cancer samples in the data, a particular classifier might classify all the observations as having cancer. The overall accuracy would be 95%, but in more detail the classifier would have a 100% recognition rate (sensitivity) for the cancer class but a 0% recognition rate for the non-cancer class. F1 score is even more unreliable in such cases, and here would yield over 97.4%, whereas informedness removes such bias and yields 0 as the probability of an informed decision for any form of guessing (here always guessing cancer).

According to Davide Chicco and Giuseppe Jurman, the most informative metric to evaluate a confusion matrix is the Matthews correlation coefficient (MCC).[20]

Other metrics can be included in a confusion matrix, each of them having their significance and use.

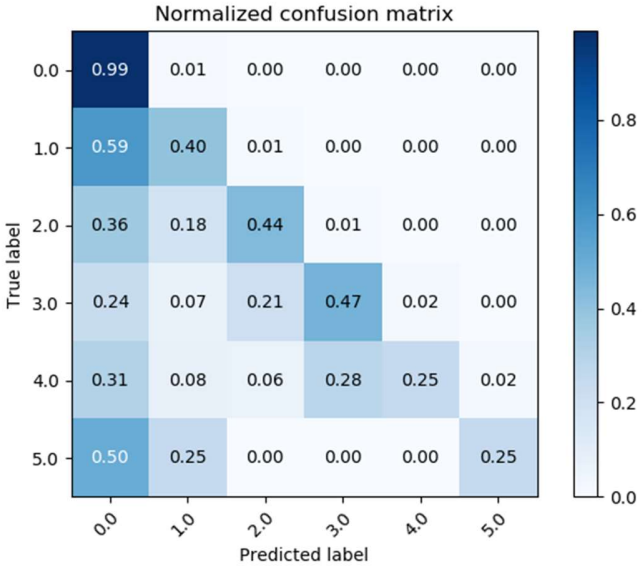
		Predicted condition		Sources: [21][22][23][24][25][26][27][28][29] view · talk · edit	
	Total population = P + N	Positive (PP)	Negative (PN)	Informedness, bookmaker informedness (BM) = TPR + TNR − 1	Prevalence threshold (PT) = $\frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	Positive (P)	True positive (TP), hit	False negative (FN), type II error, miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power = $\frac{TP}{P}$ = 1 − FNR	False negative rate (FNR), miss rate = $\frac{FN}{P}$ = 1 − TPR
	Negative (N)	False positive (FP), type I error, false alarm, overestimation	True negative (TN), correct rejection	False positive rate (FPR), probability of false alarm, fall-out = $\frac{FP}{N}$ = 1 − TNR	True negative rate (TNR), specificity (SPC), selectivity = $\frac{TN}{N}$ = 1 − FPR
	Prevalence = $\frac{P}{P+N}$	Positive predictive value (PPV), precision = $\frac{TP}{PP}$ = 1 − FDR	False omission rate (FOR) = $\frac{FN}{PN}$ = 1 − NPV	Positive likelihood ratio (LR+) = $\frac{TPR}{FPR}$	Negative likelihood ratio (LR−) = $\frac{FNR}{TNR}$
	Accuracy (ACC) = $\frac{TP+TN}{P+N}$	False discovery rate (FDR) = $\frac{FP}{PP}$ = 1 − PPV	Negative predictive value (NPV) = $\frac{TN}{PN}$ = 1 − FOR	Markedness (MK), deltaP (Δp) = PPV + NPV − 1	Diagnostic odds ratio (DOR) = $\frac{LR+}{LR-}$
	Balanced accuracy (BA) = $\frac{TPR+TNR}{2}$	F_1 score = $\frac{2PPV \times TPR}{PPV+TPR}$ = $\frac{2TP}{2TP+FP+FN}$	Fowlkes–Mallows index (FM) = $\sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) = $\sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index = $\frac{TP}{TP+FN+FP}$

Confusion matrices with more than two categories

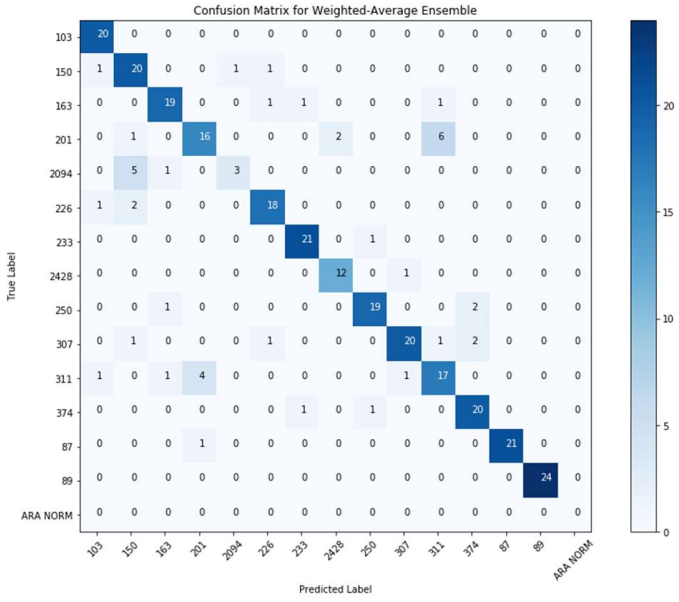
Confusion matrix is not limited to binary classification and can be used in multi-class classifiers as well.[30] The confusion matrices discussed above have only two conditions: positive and negative. For example, the table below summarizes communication of a whistled language between two speakers, zero values omitted for clarity.

Vowel produced \ Perceived vowel	i	e	a	o	u
i	**15**		1		
e	1		1		
a			**79**	5	
o			4	**15**	3
u				2	**2**

confusion matrix plot for more than 3 classes



confusion matrix returns no values in the last class



9. Αναζήτηση/κατέβασμα δεδομένων ομαδοποίησης από online datasets

Α) Αναζητήστε στο Διαδίκτυο*: Το online dataset των δεδομένων ομαδοποίησης Iris Data. Κατεβάστε τα στον Η/Υ και δώστε μια σύντομη περιγραφή και απεικονίστε τα σε διάγραμμα*

Απαντήθηκε στην 2 και 3

10. Αναζήτηση μεθόδων ομαδοποίησης και σύντομη παρουσίαση

Α) Αναζητήστε στο Διαδίκτυο*: Μεθόδους Ομαδοποίησης και βρείτε τα αντίστοιχα εργαλεία των MatLab/Octave. Δώστε μια σύντομη περιγραφή κάθε μεθόδου και τον τρόπο χρήσης και απεικονίστε τα σε διάγραμμα*

Clustering Methods

Cluster analysis, also called *segmentation analysis* or *taxonomy analysis*, is a common unsupervised learning method. Unsupervised learning is used to draw inferences from data sets consisting of input data without labeled responses. For example, you can use cluster analysis for exploratory data analysis to find hidden patterns or groupings in unlabeled data.

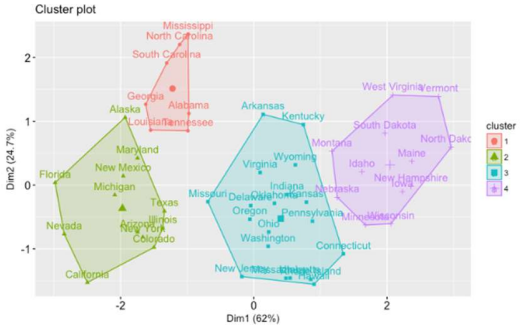
Cluster analysis creates groups, or *clusters*, of data. Objects that belong to the same cluster are similar to one another and distinct from objects that belong to different clusters. To quantify "similar" and "distinct," you can use a dissimilarity measure (or distance metric) that is specific to the domain of your application and your data set. Also, depending on your application, you might consider scaling (or standardizing) the variables in your data to give them equal importance during clustering.

Statistics and Machine Learning Toolbox provides functionality for these clustering methods:

- Hierarchical Clustering
- k-Means and k-Medoids Clustering
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN)
- Gaussian Mixture Model
- k-Nearest Neighbor Search and Radius Search
- Spectral Clustering

Hierarchical Clustering

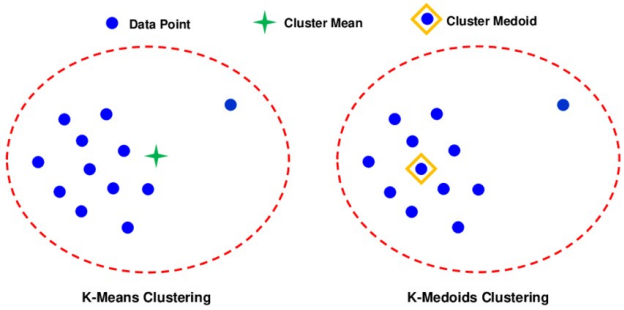
Hierarchical clustering groups data over a variety of scales by creating a cluster tree, or *dendrogram*. The tree is not a single set of clusters, but rather a multilevel hierarchy, where clusters at one level combine to form clusters at the next level. This multilevel hierarchy allows you to choose the level, or scale, of clustering that is most appropriate for your application. Hierarchical clustering assigns every point in your data to a cluster.



Use clusterdata to perform hierarchical clustering on input data. clusterdata incorporates the pdist, linkage, and cluster functions, which you can use separately

for more detailed analysis. The dendrogram function plots the cluster tree. For more information, see Introduction to Hierarchical Clustering.

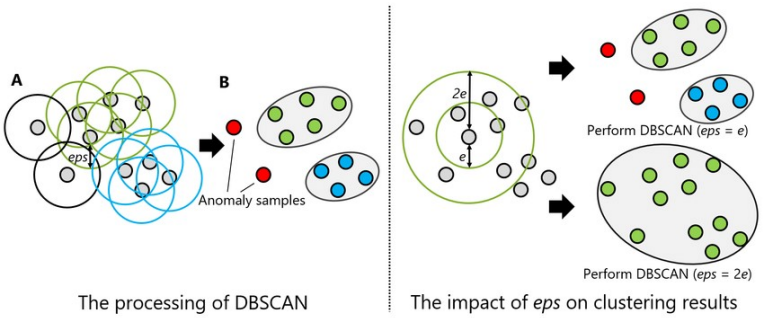
k-Means and *k*-Medoids Clustering



k-means clustering and *k*-medoids clustering partition data into *k* mutually exclusive clusters. These clustering methods require that you specify the number of clusters *k*. Both *k*-means and *k*-medoids clustering assign every point in your data to a cluster; however, unlike hierarchical clustering, these methods operate on actual observations (rather than dissimilarity measures), and create a single level of clusters. Therefore, *k*-means or *k*-medoids clustering is often more suitable than hierarchical clustering for large amounts of data.

Use kmeans and kmedoids to implement *k*-means clustering and *k*-medoids clustering, respectively. For more information, see Introduction to *k*-Means Clustering and *k*-Medoids Clustering.

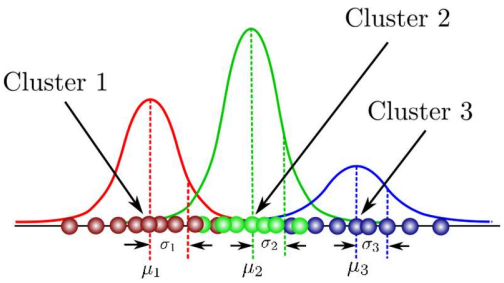
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



DBSCAN is a density-based algorithm that identifies arbitrarily shaped clusters and outliers (noise) in data. During clustering, DBSCAN identifies points that do not belong to any cluster, which makes this method useful for density-based outlier detection. Unlike *k*-means and *k*-medoids clustering, DBSCAN does not require prior knowledge of the number of clusters.

Use dbscan to perform clustering on an input data matrix or on pairwise distances between observations. For more information, see Introduction to DBSCAN.

Gaussian Mixture Model

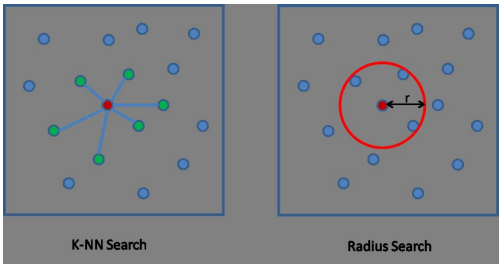


A Gaussian mixture model (GMM) forms clusters as a mixture of multivariate normal density components. For a given observation, the GMM assigns posterior probabilities to each component density (or cluster). The posterior probabilities indicate that the observation has some probability of belonging to each cluster. A GMM can perform *hard* clustering by selecting the component that maximizes the posterior probability as the assigned cluster for the observation. You can also

use a GMM to perform *soft*, or *fuzzy*, clustering by assigning the observation to multiple clusters based on the scores or posterior probabilities of the observation for the clusters. A GMM can be a more appropriate method than *k*-means clustering when clusters have different sizes and different correlation structures within them.

Use fitgmdist to fit a gmdistribution object to your data. You can also use gmdistribution to create a GMM object by specifying the distribution parameters. When you have a fitted GMM, you can cluster query data by using the cluster function. For more information, see Cluster Using Gaussian Mixture Model.

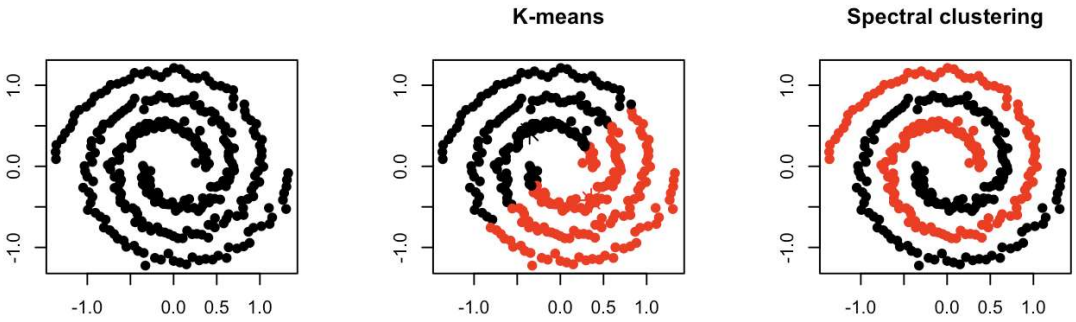
k-Nearest Neighbor Search and Radius Search



k-nearest neighbor search finds the *k* closest points in your data to a query point or set of query points. In contrast, radius search finds all points in your data that are within a specified distance from a query point or set of query points. The results of these methods depend on the distance metric that you specify.

Use the knnsearch function to find *k*-nearest neighbors or the rangesearch function to find all neighbors within a specified distance of your input data. You can also create a searcher object using a training data set, and pass the object and query data sets to the object functions (knnsearch and rangesearch). For more information, see Classification Using Nearest Neighbors.

Spectral Clustering



Spectral clustering is a graph-based algorithm for finding *k* arbitrarily shaped clusters in data. The technique involves representing the data in a low dimension. In the low dimension, clusters in the data are more widely separated, enabling you to use algorithms such as *k*-means or *k*-medoids clustering. This low dimension is based on eigenvectors of a Laplacian matrix. A Laplacian matrix is one way of representing a similarity graph that models the local neighborhood relationships between data points as an undirected graph.

Use spectralcluster to perform spectral clustering on an input data matrix or on a similarity matrix of a similarity graph. spectralcluster requires that you specify the number of clusters. However,

the algorithm for spectral clustering also provides a way to estimate the number of clusters in your data. For more information, see Partition Data Using Spectral Clustering.

Comparison of Clustering Methods

This table compares the features of available clustering methods in Statistics and Machine Learning Toolbox.

Method	Basis of Algorithm	Input to Algorithm	Requires Specified Number of Clusters	Cluster Shapes Identified	Useful for Outlier Detection
Hierarchical Clustering	Distance between objects	Pairwise distances between observations	No	Arbitrarily shaped clusters, depending on the specified `'Linkage'` algorithm	No
k-Means Clustering and k-Medoids Clustering	Distance between objects and centroids	Actual observations	Yes	Spheroidal clusters with equal diagonal covariance	No
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)	Density of regions in the data	Actual observations or pairwise distances between observations	No	Arbitrarily shaped clusters	Yes
Gaussian Mixture Models	Mixture of Gaussian distributions	Actual observations	Yes	Spheroidal clusters with different covariance structures	Yes
Nearest Neighbors	Distance between objects	Actual observations	No	Arbitrarily shaped clusters	Yes, depending on the specified number of neighbors
Spectral Clustering (Partition Data Using Spectral Clustering)	Graph representing connections between data points	Actual observations or similarity matrix	Yes, but the algorithm also provides a way to estimate the number of clusters	Arbitrarily shaped clusters	No

11. Αναζήτηση Εφαρμογών ομαδοποίησης στην ειδικότητα

Α) Αναζητήστε στο Διαδίκτυο*: Προηγμένες εφαρμογές Ομαδοποίησης, συναφείς με την ειδικότητα σας και παρουσιάστε μια σύντομη περιγραφή τους.*

Channel Clustering Software

Blue Yonder, formerly JDA Software, Channel Clustering uses data-mining techniques to form optimal retail store groupings based on user buying preferences which enables retailers to create targeted consumer strategies for store, assortment, pricing, and promotion planning.

What does it do?

- Data-mining technology helps you identify which stores exhibit similar consumer preference patterns and incorporates additional performance and attribute information to create dynamic store groupings that drive improved planning outcomes.
- Traditional approaches segment stores based on averages or assumptions, but Blue Yonder Channel Clustering groups stores based on facts such as POS data, individual store attributes, demographics, and local competitors.
- Data-mining technology allows you to confidently launch channel, assortment, and category plans and realize a fast return on investment with your merchandising programs.

How Can We Help

While traditional approaches segment stores based on averages or assumptions, Blue Yonder Channel Clustering groups stores based on facts such as POS data, individual store attributes, demographics, and local competitors.

With data-mining technology backing your efforts, you can confidently launch channel, assortment and category plans and realize a fast return on investment with your merchandising programs.

https://cantactix.com/channel-clustering/

12. Αναζήτηση Εφαρμογών Συσχέτισης στην ειδικότητα

Α) Αναζητήστε στο Διαδίκτυο*: Προηγμένες εφαρμογές Συσχέτισης, συναφείς με την ειδικότητα σας και παρουσιάστε μια σύντομη περιγραφή τους.*

How Correlation Analysis Boosts the Efficacy of eCommerce Promotions

In the first part of the blog series, we discussed how correlation analysis can be leveraged to reduce time to detection (TTD) and time to remediation (TTR) by guiding mitigation efforts early. Further, correlation analysis helps to reduce alert fatigue by filtering out irrelevant anomalies and grouping multiple anomalies stemming from a single incident into one alert.

In this edition, we throw light on the applicability of correlation analysis in the realm of eCommerce, specifically, promotions.

As we head into the holiday season, it's a tough time for retailers, especially given the revenue hit incurred by the pandemic earlier in the year. For many, Q4 accounts for more than 50 percent of their annual sales. With the pandemic accelerating digitalization, businesses will face a heightened competitive landscape. One of the highlights of the holiday season, be it Black Friday, Cyber Monday or Christmas, are the online promotions. The figure below illustrates the longitudinal variation in sales with and without an advertisement, and the number of weeks that ad was in circulation.

Figure source: Prior work by Cooper et al.

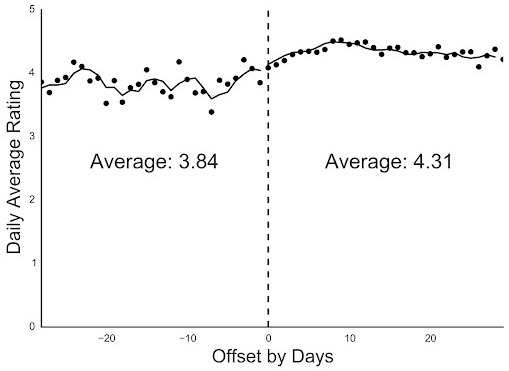
In general, there are many types of promotional campaigns (see below) and each involve a multitude of factors:

- Percentage discounts
- "ABC dollars off"
- Buy One Get One (BOGO)
- Multi-buys
- Holiday promotions
- Loyalty points
- Multi-save (ala, buy and save off the entire sale)

Optimization typically requires a deep dive to correlate the KPI(s) of interest with the various promotional levers.

For instance, which promotion(s) do you offer for which product? For how long? How do you assess the ROI and optimize efficiently? How do you avoid losses? Besides promotion type, another key factor to consider is the *surface*, whether the ad is running on traditional channels (TV, print, billboards) or on digital (desktop vs. mobile vs. in-app). Further, promotions not only help in boosting sales but also the ratings of mobile apps (it is well known that a higher rating in an app store compared to one's competitor helps boost install volume).

Consider the figure below, borrowed from the work by Askalidis 2018, which illustrates a changepoint induced in the daily average rating by a promotion. Detecting such a changepoint early and surfacing the highly correlated promotions can help inform ad spend (e.g., dynamically adjust budget amongst different promotions).



Before we dive further, one may wonder whether promotions actually help boost the top line as well as the bottom line of the business. A natural line of questioning given how much is being spent on these campaigns:

- According to Forrester, more than 20 percent of gross revenues are invested in trade promotions by global consumer packaged goods (CPG) brands. According the report, promotions help to:
    - Defend market share
    - Grow the customer base
    - Improve success rates for new product introduction
- BCG reports that, every year, fashion retailers globally invest more than $1 trillion in their markdown programs. Further, it was reported that discounts "*can boost gross margins by 10% to 20% for in-season and end-of-season sales programs.*"

Getting the promotion strategy right is not trivial. A recent report from McKinsey highlighted, "While promotions provide a short-term sales boost, they cannot generate long-term growth because they fail to address new customers, new shopping habits and preferences, or a retail environment undergoing a profound transformation."

Further, eMarketer found that promotional reliance can actually erode brand equity. To that end, many retailers are harnessing advanced analytics — big data and artificial intelligence (AI) — to develop effective promotion strategies that are appealing to consumers and lucrative for retailers. An AI-centric approach provides a more accurate way to address the dynamics owing to market seasonality, secular trends, personal preferences, inventory constraints and, consequently, enable hyper-targeting in a scalable fashion.

Owing to the factors mentioned above, correlation between different metrics may evolve over time. Detecting a change in correlation patterns, as illustrated in the figure below, can potentially help course-correct and fine-tune an ongoing promotion.

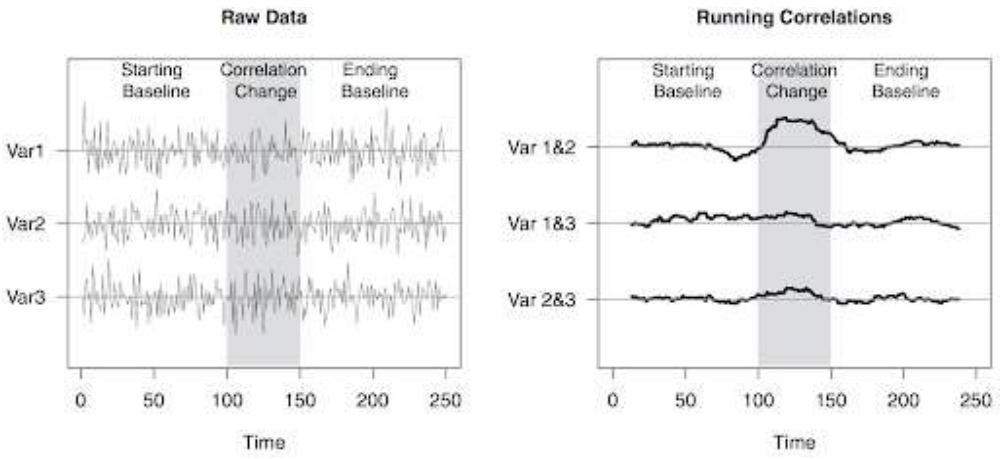


Figure source: Prior work by Cabrieto et al.

No wonder then that Global Market Insights estimates the retail analytics market size will grow at a more than 20 percent compound annual growth rate between 2020 and 2026 and will rake in more than $20 billion.

Promotions can influence consumer purchases in multiple ways, from when and what to buy, to how much to buy, and these are just a few of the dimensions to consider in decomposing campaign efficacy:

- Brand switching
- Purchase acceleration (ala, interpurchase time)
- Boosting shopping momentum
- Boosting impulse purchases
- Stockpiling
- Past promotional purchases may have a negative impact on repurchase probability of the brand\\

In a similar vein, effects of promotions differ between hedonic versus utilitarian purchases. In prior work, it is argued that promotions may help remedy hyperopia ("namely, excessive farsightedness and overcontrol") for hedonic purchases.

The above mostly capture the impact of promotions in the short term. However, as discussed in the Journal of Marketing Research, there are other factors at play that may result in delayed effects. For instance:

- *Delayed response effect:* A consumer may not buy a product immediately after seeing a promotion.

- *Customer holdover effect:* This corresponds to protracted increase in purchase rate of current customers or addition of new customers.
- *Anticipatory response effect (or lead effect):* Consumers may hold back purchases in anticipation of promotions.

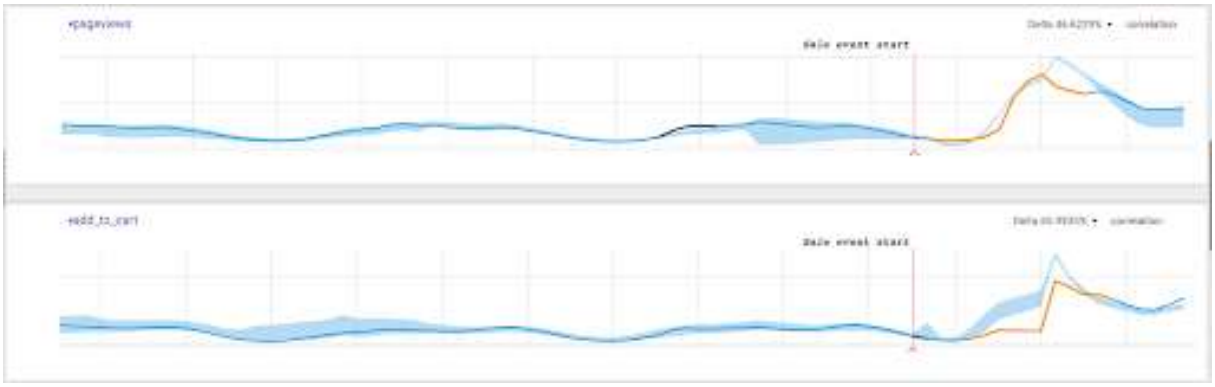
Alternatively, price promotions can have long-term effects on category incidence, brand choice and purchase quantity. The different data points needed for the analysis and the potential concerns that may skew the analysis are discussed below.

- Promotion type
- Promotion surface
- Promotional pricing
- Targeting
- Event, if any (e.g., Thanksgiving, Cyber Monday, Christmas, New Year Eve, Birthday, Anniversary)
- Incremental sales/engagement
- Intent (e.g., is the purchase meant to be a gift)
- Potential concerns
- Timeliness
- Bias, for example, by-region
- Lack of competitive promotions data can limit the efficacy of promotion response models – for instance, it would be difficult to assess the promotion-price elasticity
- Data fidelity
    - Sampling and specification error
    - Missing data

From above, we note that there are multiple factors at play and, hence, correlation analysis finds a natural fit to determine, as a first-cut, which factor(s) play a key role in driving the top and bottom lines. The ability to identify strong correlations would help marketers to double down on the corresponding promotions. On the other hand, identifying no correlation, or a weaker correlation, would help marketers dive deeper to carry out root cause analysis in a more targeted way.

Source: Mike Seddon

To illustrate, consider the figure below, which shows how two correlated anomalies – a spike in page views and add to carts – for an eCommerce site points to an anomalous sales pattern. The shaded area (the baseline) is the normal pattern of sales for a promotional event of this nature.



Clearly, the add to cart metric is underperforming. Correlating the relevant event (the sale) and the related metrics (page views and add to cart) together, it underscores the irregularity of a drop in both those metrics.

When the event started, the team was alerted about the fact that the sales event did not yield the expected increase in both the correlated metrics; in fact, page views actually dropped(!) 46 percent compared to the expected spike, leading to a drop of 66 percent in add_to_cart. These drops were identified because the effect of the sales event (an "external" variable to the metric), was correlated to the values of the metric. If the correlation between the metrics and the event was not taken into account, the drop would have seemed like an increase.

Multivariate correlation analysis helps to throw light on how the dependent variable (say, incremental sales) is associated with other independent variables. In the current context, the challenge is that some of the variables may be discrete (they may arise owing to, for instance, one-hot encoding of categorical variables) and some may be continuous. To this end, we suggest employing techniques such as those proposed by Olkin and Tate, and Cox, along with several others:

- "A Test for Serial Correlation in Multivariate Data", The Annals of Statistics
- "On the Use of the Inverse of the Correlation Matrix in Multivariate Data Analysis", The American Statistician
- "A Linear Combination Test for Detecting Serial Correlation in Multivariate Samples", Institute of Mathematical Statistics
- "Correlation Analysis for Exploring Multivariate Data Sets", IEEE

Some of the measures for capturing multivariate correlation include multiple correlation, unsigned correlation coefficient (UCC) and the unsigned in correlation coefficient (UIC).

https://www.anodot.com/blog/correlation-analysis-ecommerce-promotions/

13. Εργαλεία υλοποίησης Νευρωνικών Δικτύων σε MatLab/Octave

Α) Αναζητήστε στο Διαδίκτυο*: Εργαλεία υλοποίησης ΤΝΔ σε MatLab/Octave. Δώστε μια σύντομη περιγραφή κάθε μεθόδου και τον τρόπο χρήσης και απεικονίστε τα σε διάγραμμα*

Neural Net Clustering

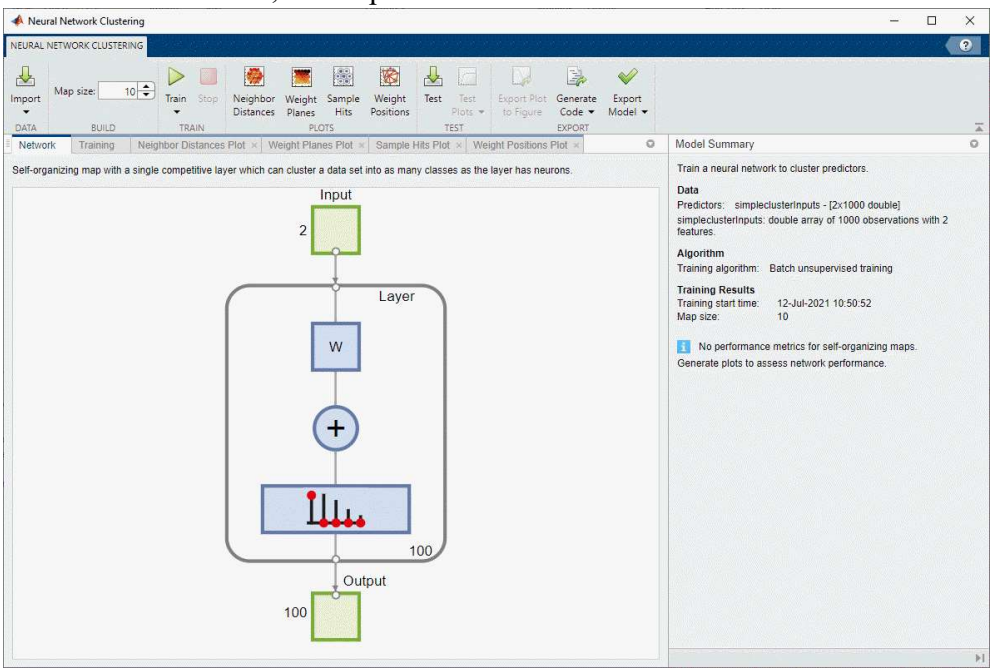
Solve clustering problem using self-organizing map (SOM) networks

Description

The **Neural Net Clustering** app lets you create, visualize, and train self-organizing map networks to solve clustering problems.

Using this app, you can:

- Import data from file, the MATLAB® workspace, or use one of the example data sets.
- Define and train a neural network.
- Analyze results using visualization plots, such as neighbor distance, weight planes, sample hits, and weight position.
- Generate MATLAB scripts to reproduce results and customize the training process.
- Generate functions suitable for deployment with MATLAB Compiler™ and MATLAB Coder™ tools, and export to Simulink® for use with Simulink Coder.



Neural Net Fitting

Solve fitting problem using two-layer feed-forward networks

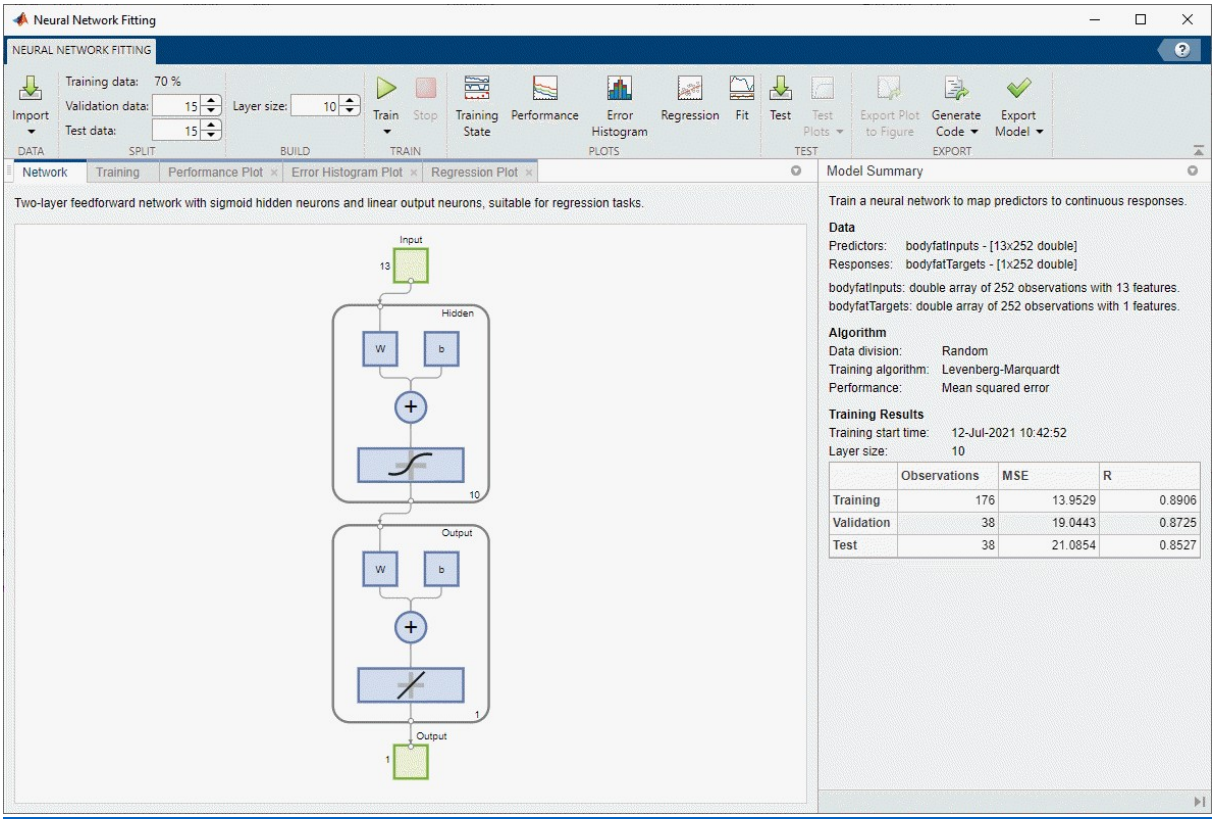
Description

The **Neural Net Fitting** app lets you create, visualize, and train a two-layer feed-forward network to solve data fitting problems.

Using this app, you can:

- Import data from file, the MATLAB® workspace, or use one of the example data sets.

- Split data into training, validation, and test sets.
- Define and train a neural network.
- Evaluate network performance using mean squared error and regression analysis.
- Analyze results using visualization plots, such as regression fit or histogram of errors.
- Generate MATLAB scripts to reproduce results and customize the training process.
- Generate functions suitable for deployment with MATLAB Compiler™ and MATLAB Coder™ tools, and export to Simulink® for use with Simulink Coder.



Neural Net Pattern Recognition

Solve pattern recognition problem using two-layer feed-forward networks

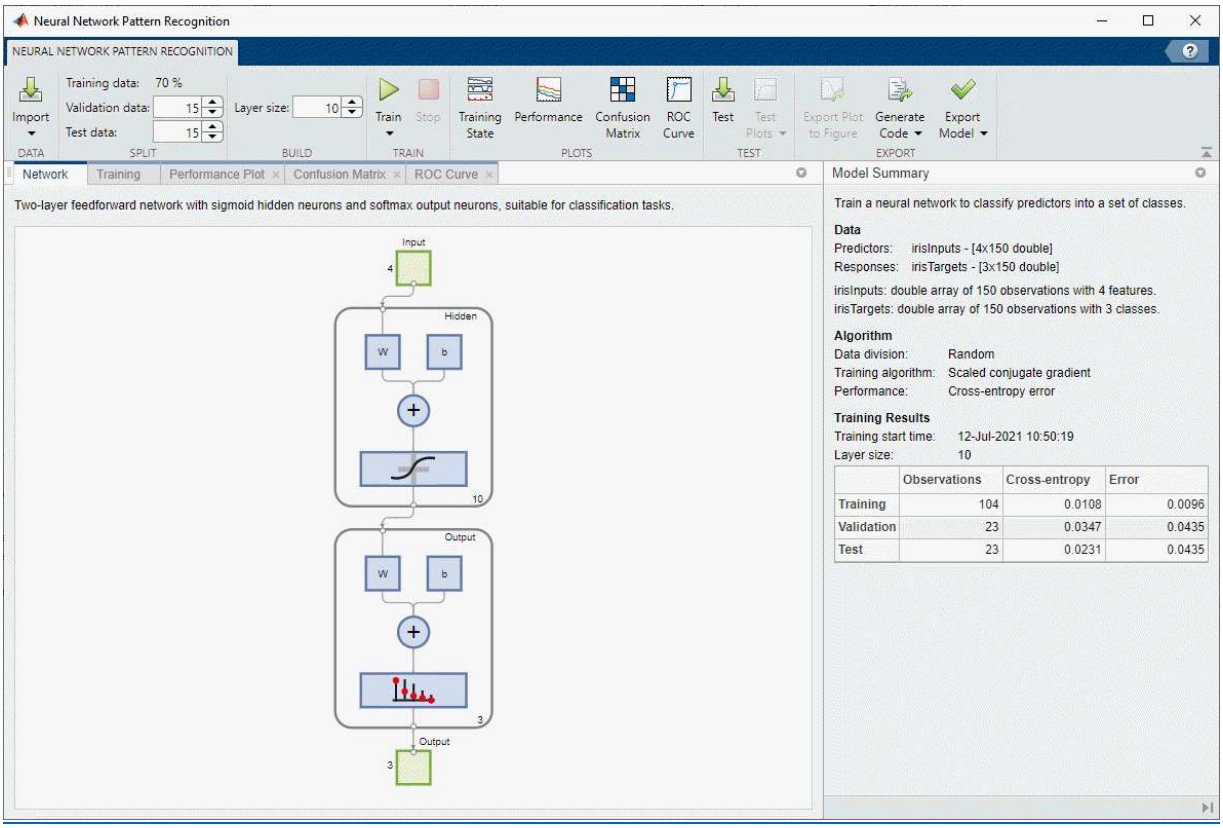
Description

The **Neural Net Pattern Recognition** app lets you create, visualize, and train two-layer feed-forward networks to solve data classification problems.

Using this app, you can:

- Import data from file, the MATLAB® workspace, or use one of the example data sets.
- Split data into training, validation, and test sets.
- Define and train a neural network.
- Evaluate network performance using cross-entropy error and misclassification error.

- Analyze results using visualization plots, such as confusion matrices and receiver operating characteristic curves.
- Generate MATLAB scripts to reproduce results and customize the training process.
- Generate functions suitable for deployment with MATLAB Compiler™ and MATLAB Coder™ tools, and export to Simulink® for use with Simulink Coder.



Neural Net Time Series

Solve nonlinear time series problem using dynamic neural networks

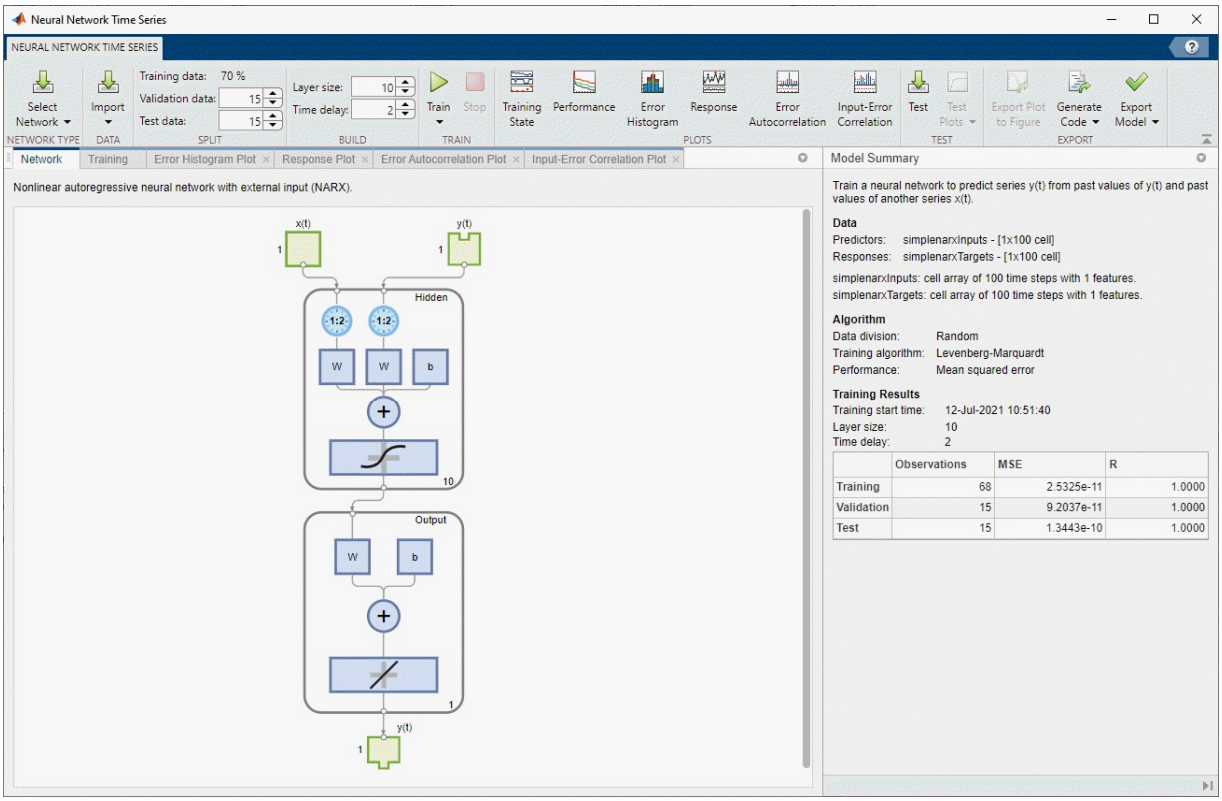
Description

The **Neural Net Time Series** app lets you create, visualize, and train dynamic neural networks to solve three different kinds of nonlinear time series problems.

Using this app, you can:

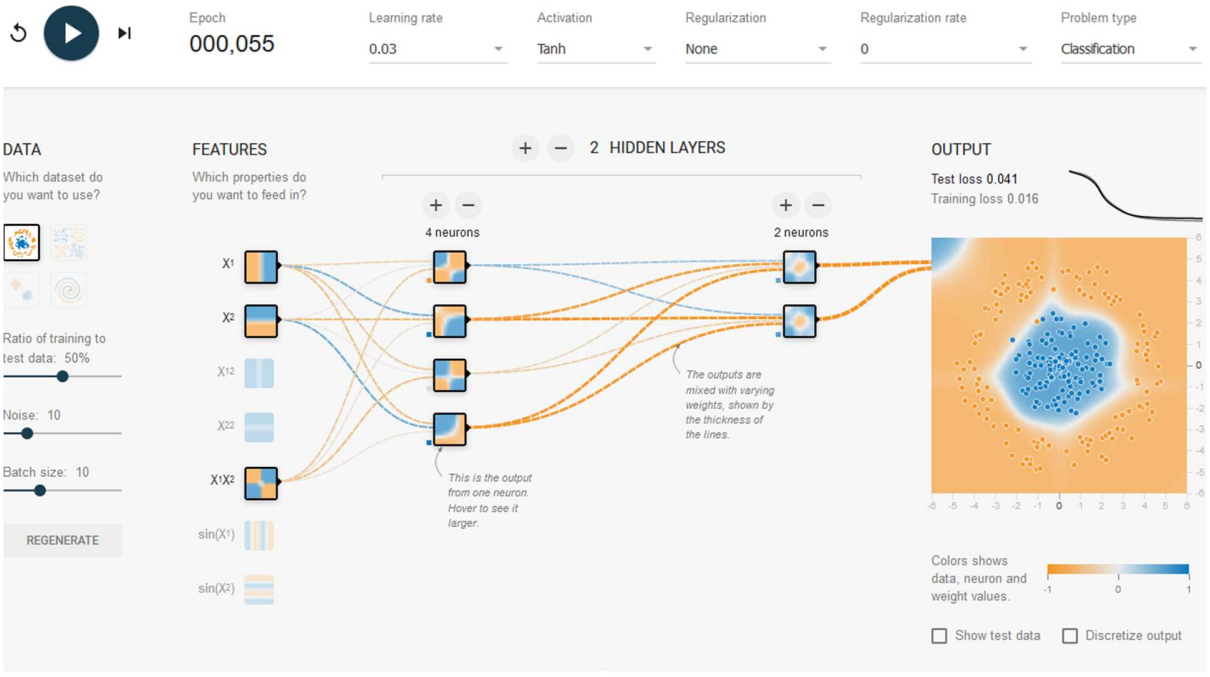
- Create three types of neural networks: NARX networks, NAR networks, and nonlinear input-output networks.
- Import data from file, the MATLAB® workspace, or use one of the example data sets.
- Split data into training, validation, and test sets.
- Define and train a neural network.
- Evaluate network performance using mean squared error and regression analysis.

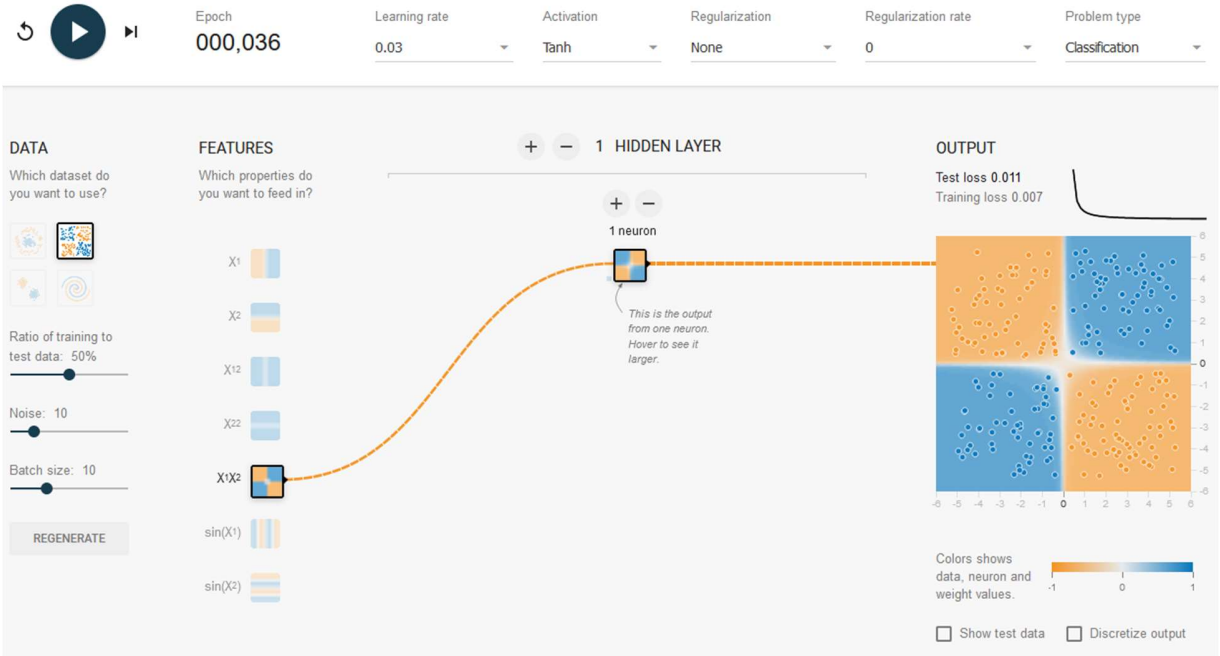
- Analyze results using visualization plots, such as autocorrelation plots or a histogram of errors.
- Generate MATLAB scripts to reproduce results and customize the training process.
- Generate functions suitable for deployment with MATLAB Compiler™ and MATLAB Coder™ tools, and export to Simulink® for use with Simulink Coder.

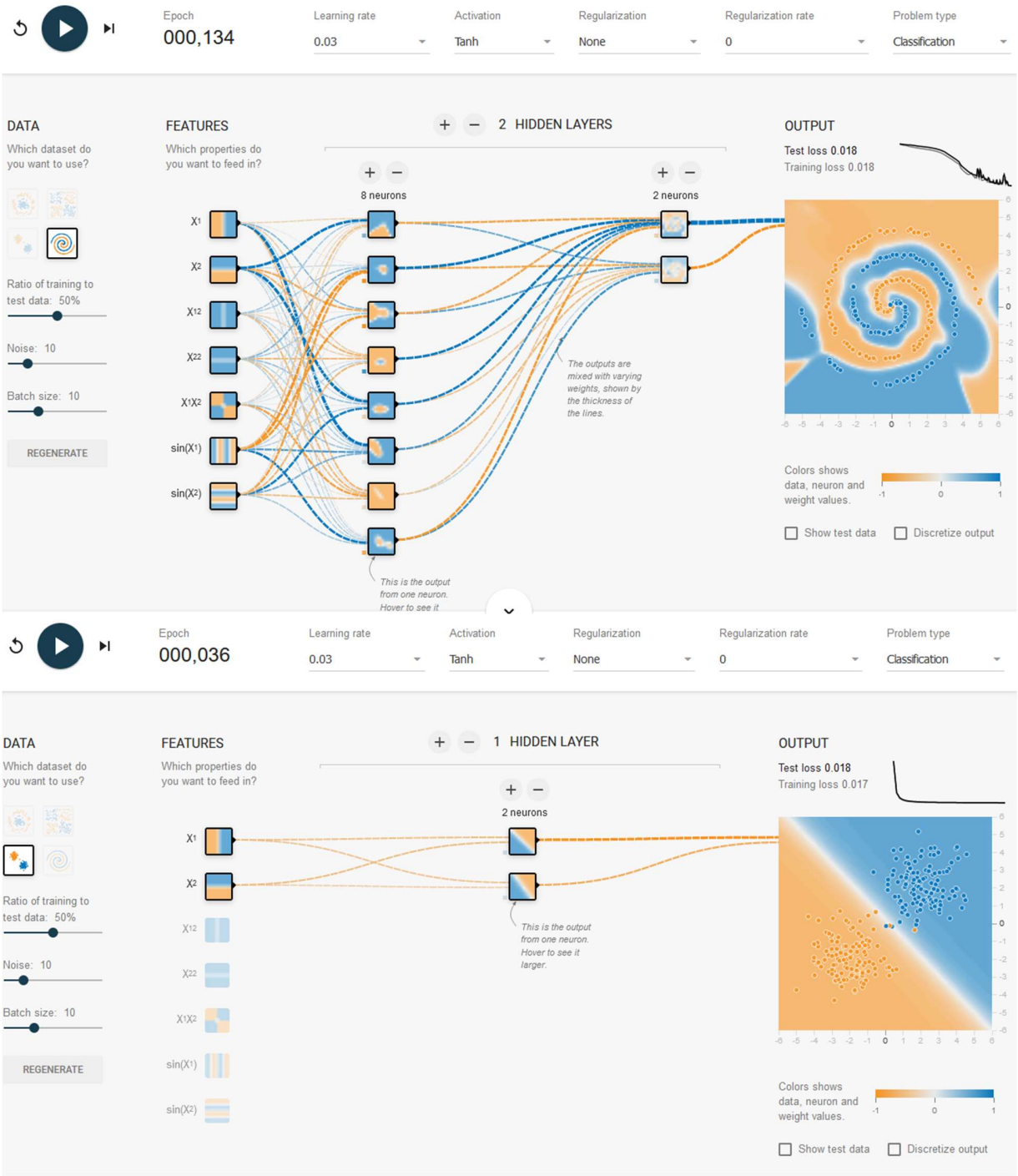


14. Online εφαρμογές και διαδραστική υλοποίηση/επίδειξη ΤΝΔ

Να εκτελέσετε/δοκιμάσετε την Online Demo Εφαρμογή: https://playground.tensorflow.org/ για τις 4 περιπτώσεις δεδομένων με διαφορετικούς συνδυασμούς επιπέδων & νευρώνων (βλ. σχετικές οδηγίες περιπτώσεις από τη διάλεξη)







15. Αναζήτηση Εφαρμογών ΤΝΔ Εκτίμησης στην ειδικότητα

Α) Αναζητήστε στο Διαδίκτυο*: Προηγμένες εφαρμογές ΤΝΔ Εκτίμησης, συναφείς με την ειδικότητα σας και παρουσιάστε μια σύντομη περιγραφή τους.*

Retail segmentation using artificial neural networks

Derrick S. Boone*, Michelle Roehm

Advances in information technology (e.g., scanner data, cookies, and other electronically based data collection methodologies) have enabled researchers to collect unprecedented amounts of individual-level customer data. As a result, customer databases are becoming increasingly larger and more complex, and may tax the capabilities and exacerbate the shortcomings of the techniques currently used to analyze them. To address this challenge, we examine the use of artificial neural networks (ANNs) as an alternative means of segmenting retail databases. In particular, we investigate the Hopfield–Kagmar (HK) clustering algorithm, an ANN technique based on Hopfield networks, and empirically compare it to K-means and mixture model clustering algorithms. Our results indicate that ANNs may be more useful to retailers for segmenting markets because they provide more homogeneous segmentation solutions than mixture model and K-means clustering algorithms, and are less sensitive to initial starting conditions

https://www.researchgate.net/publication/222902597_Retail_segmentation_using_artificial_neural_networks

An Application of Artificial Neural Networks to Estimate the Performance of High-Energy Laser Weapons in Maritime Environments

Antonios Lionis , Andreas Tsigopoulos and Keith Cohn

Abstract: Efforts to develop high-energy laser (HEL) weapons that are capable of being integrated and operated aboard naval platforms have gained an increased interest, partially due to the proliferation of various kinds of unmanned systems that pose a critical asymmetric threat to them, both operationally and financially. HEL weapons allow for an unconstrained depth of magazine and cost exchange ratio, both of which are essential characteristics to effectively oppose small unmanned systems, compared to their kinetic weapons counterparts. However, HEL performance is heavily affected by atmospheric conditions between the weapon and the target; therefore, the more precise and accurate the atmospheric characterization, the more accurate the performance estimation of the HEL weapon. To that end, the Directed Energy Group of the Naval Postgraduate School (NPS) is conducting experimental, theoretical and computational research on the effects of atmospheric conditions on HEL weapon efficacy. This paper proposes a new approach to the NPS laser performance code scheme, which leverages artificial neural networks (ANNs) for the prediction of optical turbulence strength. This improvement could allow for near real-time and location-independent HEL weapon performance estimation. Two experimental datasets, which were obtained from the NPS facilities, were utilized to perform regression modeling using an ANN, which achieved a decent fit (R2 = 0.75 for the first dataset and R2 = 0.78 for the second dataset).

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiEupSTgoj8AhU6h_0HHXkVAMwQFnoECBMQAQ&url=https%3A%2F%2Fwww.researchgate.net%2Fpublication%2F227727199_Advanced_Applications_of_Neural_Networks_and_Artificial_Intelligence_A_Review&usg=AOvVaw02WudCgwb4SFN6ERyFz1p2

16. Αναζήτηση Εφαρμογών ΤΝΔ Ταξινόμησης στην ειδικότητα

Α) Αναζητήστε στο Διαδίκτυο*: Προηγμένες εφαρμογές ΤΝΔ Ταξινόμησης, συναφείς με την ειδικότητα σας και παρουσιάστε μια σύντομη περιγραφή τους.*

Towards Identification of Packaged Products via Computer Vision: Convolutional Neural Networks for Object Detection and Image Classification in Retail Environments

Identification of packaged products in retail environments still relies on barcodes, requiring active user input and limited to one product at a time. Computer vision (CV) has already enabled many applications, but has so far been under-discussed in the retail domain, albeit allowing for faster, hands-free, more natural human-object interaction (e.g. via mixed reality headsets). To assess the potential of current convolutional neural network (CNN) architectures to reliably identify packaged products within a retail environment, we created and open-source a dataset of 300 images of vending machines with 15k labeled instances of 90 products. We assessed observed accuracies from transfer learning for image-based product classification (IC) and multi-product object detection (OD) on multiple CNN architectures, and the number of images instances required per product to achieve meaningful predictions. Results show that as little as six images are enough for 90% IC accuracy, but around 30 images are needed for 95% IC accuracy. For simultaneous OD, 42 instances per product are necessary and far more than 100 instances to produce robust results. Thus, this study demonstrates that even in realistic, fast-paced retail environments, image-based product identification provides an alternative to barcodes, especially for use-cases that do not require perfect 100% accuracy.

https://dl.acm.org/doi/abs/10.1145/3365871.3365899

Quantitative Analysis of Fish Microbiological Quality Using Electronic Tongue Coupled with Nonlinear Pattern Recognition Algorithms

Abstract

The objective of this study was to establish quantitative evaluation models for fish microbiological quality analysis based on electronic tongue technique coupled with nonlinear pattern recognition algorithms. Crucian carp stored at 4C were used. A commercial electronic tongue system was employed. The total viable counts (TVCs) of fish samples were measured by the classical microbiological plating method. Partial least square regression, support vector regression (SVR) and back propagation neural network (BP-NN) were applied comparatively to predict TVC values. The multivariate regression models were evaluated by the root mean square error of prediction (RMSEP) and the correlation coefficient in prediction set (R_{pre}). Results revealed that the performance of BP-NN model was superior to that of PLS model and SVR model. The RMSEP and R_{pre} of the BP-NN model for TVC prediction were 0.211 ln colony-forming unit (cfu)/g and 0.993, respectively. This study showed that electronic tongue together with BP-NN model could be a reliable technique for the detection of fish microbiological quality.

Practical Applications

Fish is a highly perishable commodity after harvesting and postmortem as a consequence of microbial breakdown mechanisms. Total viable count (TVC) method is the most widely used microbiological indicator for the evaluation of fish microbiological quality. However, the conventional analytical methods for the determination of TVC are cumbersome and time wasting. This work provides a practical and efficient way for rapid, accurate and convenient determination of TVC in fish using electronic tongue combined with regression algorithms to address these limitations.

https://onlinelibrary.wiley.com/doi/abs/10.1111/jfs.12180

17. Αναζήτηση Εφαρμογών ΤΝΔ Ομαδοποίησης στην ειδικότητα

Α) Αναζητήστε στο Διαδίκτυο*: Προηγμένες εφαρμογές ΤΝΔ Ομαδοποίησης, συναφείς με την ειδικότητα σας και παρουσιάστε μια σύντομη περιγραφή τους.*

Retail segmentation using artificial neural networks

Abstract

Advances in information technology (e.g., scanner data, cookies, and other electronically based data collection methodologies) have enabled researchers to collect unprecedented amounts of individual-level customer data. As a result, customer databases are becoming increasingly larger and more complex, and may tax the capabilities and exacerbate the shortcomings of the techniques currently used to analyze them. To address this challenge, we examine the use of artificial neural networks (ANNs) as an alternative means of segmenting retail databases. In particular, we investigate the Hopfield–Kagmar (HK) clustering algorithm, an ANN technique based on Hopfield networks, and empirically compare it to *K*-means and mixture model clustering algorithms. Our results indicate that ANNs may be more useful to retailers for segmenting markets because they provide more homogeneous segmentation solutions than mixture model and *K*-means clustering algorithms, and are less sensitive to initial starting conditions.

Introduction

Retailers have long recognized the importance of tailoring their marketing mixes to suit the specific needs and preferences of different customer groups. However, access to unprecedented amounts of individual-level customer data may make such targeted promotional efforts increasingly difficult to effectively implement. Recent growth in the use of loyalty programs, personal shopping programs, scanners, cookies, and other electronically based data collection methodologies has resulted in an “embarrassment of riches” that may only serve to complicate market segmentation and targeting.

Given the easy access to customer data afforded by information technology, retailer databases are becoming substantially larger and noisier. As a result, how to effectively handle, analyze, and interpret customer information will be one of the key challenges facing retailers who wish to

execute segment-specific marketing mixes. According to Shapiro and Varian (1999), those firms that master information about their customers will thrive by delivering customized, highly valued offerings, while those that do not will be at a competitive disadvantage.

In order to more effectively implement targeted marketing mixes, it is vitally important that retailers segment their customer bases such that sufficient commonalities exist within, and sufficient distinctions between, each segment to justify taking the time and expense to create separate marketing mixes. Furthermore, the correct assignment of customers to the resulting segments is critical because improper segmentation reduces the effectiveness of a segmentation strategy and squanders marketing resources. Consider, for example, a retailer that has information regarding several million customers in its database and wants to send targeted promotional messages (e.g., brochures, etc.) to the various segments. If customers are sent the wrong brochures because they are assigned to the wrong segments, the marketing efforts directed towards them are likely to be ineffective and for naught. Moreover, the financial costs associated with such mistakes can be nontrivial.

According to the Direct Marketing Association (1999), the cost of printing, handling, and mailing a single page letter and brochure is between US$.93 and US$1.10 per customer. Thus, when segmenting databases that contain millions of customers, the assignment of even a small fraction of customers to incorrect segments can waste tens, or even hundreds, of thousands of marketing dollars. Therefore, retailers must exercise care when selecting a clustering algorithm for segmenting their markets because the resulting solutions are highly dependent upon the algorithm being used. (See Arabie, Hubert, and De Soete (1996), Punj and Stewart (1983), and Wedel and Kamakura (2000) for a review of clustering algorithms.)

For example, prior researchers have noted that the computational burdens associated with hierarchical clustering techniques make them less appropriate than nonhierarchical techniques for segmenting large databases Milligan & Sokol, 1980, Punj & Stewart, 1983, Wedel & Kamakura, 2000. However, these same authors note that nonhierarchical techniques tend to perform poorly unless “rational” information regarding initial centroid locations (cluster seeds) is provided a priori. Commonly, such information is based on a hierarchical clustering technique and/or managerial expertise. We contend that for large data sets, the computational burdens associated with obtaining hierarchical-based cluster seeds, and the biases and inaccuracies that may be associated with managerially based seed selection, highlight the need for new methods that provide optimal segmentation solutions independent of a priori information.

The central theme of this paper follows from the above notions. We argue that analytical techniques that do not require a priori seed information are needed in order for retailers to more effectively segment and target noisy, data-rich marketplaces. Accordingly, we examine artificial neural networks (ANNs) as an alternative methodology for segmenting retail customer databases. Although ANN computational systems have been applied to a wide variety of problems in engineering, computer science, mathematics, and other areas, management scientists have only recently begun to investigate the potential of ANNs to solve business-related problems in general, and marketing-related problems in particular. (See Krycha and Wagner (1999) for a review of ANN applications in marketing and other business disciplines.)

We address this oversight by investigating the segmentation utility of ANNs for retailer databases. More specifically, we focus on an application of Hopfield (1982) networks, an ANN particularly well suited for solving segmentation problems. We begin by providing an overview of ANNs in general. We then review the Hopfield–Kagmar (HK) clustering algorithm and discuss why it should theoretically provide better segmentation solutions than other clustering techniques (Kagmar-Parsi, Gualtieri, Devaney, & Kagmar-Parsi, 1990). Following this discussion, we empirically test HK, *K*-means and mixture model algorithms using a real world retail data set and a variety of artificial data sets Dempster et al., 1977, MacQueen, 1967. Finally, we present a general discussion of our findings and offer observations about limitations and directions for future research.

Section snippets

Artificial neural networks (ANNs)

The term “artificial neural network” (ANN) describes a family of analytical models that is based on the physiological properties of animal nervous systems. As with animal nervous systems, ANNs are composed of interconnected “nodes” (neurons) that are capable of processing and transmitting information. How the interconnections are modeled (architecture) determines how information is transmitted in the network and its properties. (See Gurney (1997) for an overview of ANNs in general, and Arabie

Hopfield–Kagmar (HK) clustering

Fig. 1b depicts a Hopfield (1982) network. Unlike other ANNs, each node in a Hopfield network is connected to all other nodes, but not to itself, and information can flow from node to node in multiple directions. Thus, all information in a Hopfield network is propagated within the net, making it highly useful for combinatorial optimization applications in which the goal is to minimize some objective function.

Given its architecture, a Hopfield network has certain inherent advantages over other

Empirical study

In testing the HK algorithm, we used a two-pronged approach. First, we compared HK and traditional clustering algorithms using a real world data set. Second, because the efficacy with which a clustering algorithm returns a market's true structure can only be ascertained when the true cluster structure is known unambiguously, HK and traditional clustering algorithms were compared using a variety of artificial data sets.

Results

The results of our analyses for the real world data set, which consisted of 4317 customer-level observations of six purchase behavior variables, are presented in Table 3, Table 4.

Analysis of within-segment variation, collapsed across seed-type, revealed that there are differences in the abilities of HK, *K*-means (KM) and the mixture model (MM) to recover

homogenous solutions ($\bar{x}_{HK}$=19.26, $\bar{x}_{KM}$=20.07, $\bar{x}_{MM}$=28.26; HK versus KM: $F(1,996)=3.39$, $p=0.07$; HK versus MM: $F(1,996)=424.85$, $p<0.00$). Further

General discussion

Our research has been motivated by several simple observations. First, the use of segment-specific marketing mixes is widespread among retailers. Second, increasing access to individual-level customer information has led to the compilation of large databases that are likely to contain hundreds of variables and data complexities such as outliers, noise, and unequally sized clusters. Third, the current analytical techniques used to analyze such databases are strained under such conditions, making

Limitations and directions for future research

Although this research advances our understanding of the Hopfield–Kagmar (HK) clustering algorithm as an alternative segmentation methodology, it is not without limitations. Given that within-segment variation is the optimization function being minimized, HK may over-fit the data. Thus, in order to more fully assess the performance of HK, it should be evaluated across more real world data sets and validated by comparing HK-based segmentation schemes with actual segmentation outcomes

NEURAL NETWORKS IN DATA MINING:

In more practical terms neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. Using neural networks as a tool, data warehousing firms are harvesting information from datasets in the process known as data mining. The difference between these data warehouses and ordinary databases is that there is actual anipulation and cross-fertilization of the data helping users makes more informed decisions.

Neural networks essentially comprise three pieces: the architecture or model; the learning algorithm; and the activation functions. Neural networks are programmed or "trained" to ". . . store, recognize, and associatively retrieve patterns or database entries; to solve combinatorial optimization problems; to filter noise from measurement data; to control ill-defined problems; in summary, to estimate sampled functions when we do not know the form of the functions." It is precisely these two abilities (pattern recognition and function estimation) which make artificial neural networks (ANN) so prevalent a utility in data mining. As data sets grow to massive sizes, the need for automated processing becomes clear. With their "model-free" estimators and their dual nature, neural networks serve data mining in a myriad of ways.

Data mining is the business of answering questions that you've not asked yet. Data mining reaches deep into databases. Data mining tasks can be classified into two categories: Descriptive and predictive data mining. Descriptive data mining provides information to understand what is happening inside the data without a predetermined idea. Predictive data mining allows the user to submit records with unknown field values, and the system will guess the unknown values based on previous patterns discovered form the database. Data mining models can be categorized according to the tasks they perform: Classification and Prediction, Clustering, Association Rules.

Classification and prediction is a predictive model, but clustering and association rules are descriptive models.

The most common action in data mining is classification. It recognizes patterns that describe the group to which an item belongs. It does this by examining existing items that already have been classified and inferring a set of rules. Similar to classification is clustering. The major difference being that no groups have been predefined. Prediction is the construction and use of a model to assess the class of an unlabeled object or to assess the value or value ranges of a given object is likely to have. The next application is forecasting. This is different from predictions because it estimates the future value of continuous variables based on patterns within the data. Neural networks, depending on the architecture, provide associations, classifications, clusters, prediction and forecasting to the data mining industry.

Financial forecasting is of considerable practical interest. Due to neural networks can mine valuable information from a mass of history information and be efficiently used in financial areas, so the applications of neural networks to financial forecasting have been very popular over the last few years. Some researches show that neural networks performed better than conventional statistical approaches in financial forecasting and are an excellent data mining tool. In data warehouses, neural networks are just one of the tools used in data mining. ANNs are used to find patterns in the data and to infer rules from them. Neural networks are useful in providing information on associations, classifications, clusters, and forecasting. The back propagation algorithm performs learning on a feed-forward neural network.

18. Μελέτη πρόσφατων δημοσιεύσεων (VM et al.)

Να μελετήσετε τις παρακάτω πρόσφατες δημοσιεύσεις και να παρουσιάσετε/σχολιάσετε τις μεθόδους που χρησιμοποιήθηκαν:

https://doi.org/10.3390/info12030118

Two-Level ANN

Η προτεινόμενη μέθοδος ANN δύο επιπέδων υπερέχει σε απλότητα, ακρίβεια και ταχύτητα, είναι εύκολη στην εφαρμογή και γρήγορη στην εκτέλεση, επομένως μπορεί να εφαρμοστεί σε λογισμικό προστασίας από ιούς, έξυπνα τείχη προστασίας, εφαρμογές Ιστού κ.λπ.

https://doi.org/10.1016/j.jrmge.2020.10.001

Multi-layer ANN model structure

Σε αυτή τη μελέτη, σχεδιάστηκε ένα τεχνητό νευρωνικό δίκτυο (ANN) με στόχο την έμμεση πρόβλεψη του uniaxial compressive strength (UCS) μέσω του ποσοστού σερπεντινοποίησης (serpentinization) και των φυσικών, δυναμικών και μηχανικών χαρακτηριστικών των σερπεντινιτών. Ο προτεινόμενος τύπος βασισμένος σε ANN βρέθηκε ότι είναι πολύ αποτελεσματικός στην πρόβλεψη τιμών UCS και τα δείγματα μπορούσαν να ταξινομηθούν με

απλές φυσικές, δυναμικές και μηχανικές δοκιμές, έτσι ώστε να αποφευχθούν οι ακριβές, δύσκολες, χρονοβόρες και καταστροφικές μηχανικές δοκιμές.

https://www.naun.org/main/NAUN/neural/2019/a102016-060.pdf

Multiple ANN modules

Περίληψη - Αυτή η εργασία προτείνει μια νέα ανίχνευση εισβολής Σύστημα (IDS) βασισμένο σε τεχνητά νευρωνικά δίκτυα (ANN). ο Το προτεινόμενο σύστημα πολλαπλών ANN είναι αρθρωτό, παράλληλο και εύκολο επεκτάσιμη για τον εντοπισμό πρόσθετων τύπων επιθέσεων. Τρεις τύποι από επιθέσεις έχουν δοκιμαστεί μέχρι στιγμής: επιθέσεις DDoS, PortScan και Web. Τα πειραματικά αποτελέσματα προέκυψαν με ανάλυση και δοκιμή του Τα προτεινόμενα IDS χρησιμοποιώντας το σύνολο δεδομένων CICIDS2017, δείχνουν ικανοποιητικά απόδοση και υπεροχή ως προς την ακρίβεια, το ποσοστό ανίχνευσης, το ψευδές ρυθμός συναγερμού και επιβάρυνση χρόνου σε σύγκριση με το υπάρχον single-ANN συστήματα.

http://users.uniwa.gr/vmouss/papers/P54.pdf

Genetic Algorithms

Η βελτιστοποίηση ροής εργασιών προσομοίωσης έχει γίνει ένας σημαντικός τομέας έρευνας, καθώς επιτρέπει στους χρήστες να επεξεργάζονται μεγάλης κλίμακας και ετερογενή προβλήματα σε κατανεμημένα περιβάλλοντα με πιο ευέλικτο τρόπο. Οι περισσότερο χαρακτηριστικές κατηγορίες τέτοιων προβλημάτων προέρχονται από την αεροδιαστημική και την αυτοκινητοβιομηχανία. Σε αυτό χρησιμοποίηθηκε ένας ειδικά αναπτυγμένος αλγόριθμος που βασίζεται σε **τεχνικές ευρετικής βελτιστοποίησης** (Γενετικοί Αλγόριθμοι) εφαρμόζεται για την παροχή μιας βελτιστοποιημένης υλοποίησης ροής εργασιών ενός αρχικού χρονοδιαγράμματος ροής εργασιών (PERT). Για να για να καταδείξει τις δυνατότητές του, ο αλγόριθμος εφαρμόζεται σε ένα δείγμα προβλήματος σχεδιασμού κατασκευαστικού προϊόντος που απαιτεί πολύ χρονοβόρες προσομοιώσεις και ανάλυση πεπερασμένων στοιχείων υπό περιορισμένη διαθεσιμότητα πόρους υπολογιστών.

Στην εργασία αυτή παρουσιάστηκε μια εξελικτική μέθοδος για τη βελτιστοποίηση της εκτέλεσης της προσομοίωσης ροές εργασιών. Η μέθοδος βασίζεται σε μια ευρετική τεχνική βελτιστοποίησης (Γενετικοί Αλγόριθμοι) δηλαδή εφαρμόζεται για την παροχή βελτιστοποιημένης υλοποίησης ροής εργασίας ενός αρχικού χρονοδιαγράμματος ροής εργασιών. ο Η μέθοδος μπορεί να χρησιμοποιηθεί on-line και σε συνδυασμό με μια παγκόσμια τεχνική βελτιστοποίησης ενημερώνει συνεχώς τα χρονοδιαγράμματα εκτέλεσης των υπολογιστικών εργασιών κάθε περίπλοκης διαδικασίας. Η προτεινόμενη μέθοδος δείχνει τις δυνατότητές της όταν το πρόβλημα σχεδιασμού προϊόντος απαιτεί πολλά χρονοβόρων προσομοιώσεων ή ανάλυσης πεπερασμένων στοιχείων υπό περιορισμένη διαθεσιμότητα πόρους υπολογιστών.

19. Αναζήτηση μεγάλων dataset

Α) Αναζητήστε στο Διαδίκτυο*: Big dataset από βάσεις δεδομένων όπως το Kaggle. Κατεβάστε τα στον Η/Υ και δώστε μια σύντομη περιγραφή*

Employee Performance Prediction

About Dataset

The garment industry is one of the most dominating industries in this era of industrial globalization. It is a highly labor-intensive industry that requires a large number of human resources to produce its goods and fill up the global demand for garment products. Because of the dependency on human labor, the production of a garment company comprehensively relies on the productivity of the employees who are working in different departments of the company. A common problem in this industry is that the actual productivity of the garment employees sometimes does not meet the targeted productivity that was set for them by the authorities to meet the production goals in due time. When the productivity gap occurs, the company faces a huge loss in production.

Dataset Description

This dataset includes important attributes of the garment manufacturing process and the productivity of the employees which had been collected manually and also been validated by the industry experts.

day : Day of the Week
quarter : A portion of the month. A month was divided into four quarters
department : Associated department with the instance
team*no :* *Associated team number with the instance no*of*workers : Number of workers in each*
***team no*of*style*change** : Number of changes in the style of a particular product
targeted*productivity :* *Targeted productivity set by the Authority for each team for each day.*
***smv** : Standard Minute Value, it is the allocated time for a task wip : Work in progress. Includes the number of unfinished items for products over*time : Represents the amount of overtime by each team in minutes
incentive : Represents the amount of financial incentive (in BDT) that enables or motivates a particular course of action.
idle*time :* *The amount of time when the production was interrupted due to several reasons*
***idle*men :** The number of workers who were idle due to production interruption
actual_productivity : The actual % of productivity that was delivered by the workers. It ranges from 0-1.

College Performance, Debt and Earnings

About this dataset

This College Scorecard dataset offers a comprehensive look into the performance, cost, and outcomes of U.S. colleges and universities. It contains an extensive amount of data detailing

information related to cost of attendance and various outcomes such as average salary after graduation, loan repayment rates, and gainful employment rates for graduates. The datasets also provides information on program level demographics such as gender breakdowns among enrolled students and faculty diversity in programs attended by students. This is an invaluable source of information for anyone who wants to make informed choices about their college education experience in terms of both costs and expected returns after graduation. Going beyond financial metrics, this dataset give insight into the cultural climate at each college or university so that users can analyze whether their unique backgrounds or experiences will fit into the campus ethos at those institutions. With this data set available to everyone interested in higher education options, individuals have a powerful tool to compare options from many perspectives including financial investment returns economics , educational quality measures , graduate success rate indices , faculty diversity break-ups etc

20. Αναζήτηση online εφαρμογών για χρήση/υλοποίηση ΤΝΔ

Α) Αναζητήστε στο Διαδίκτυο*: ONLINE εφαρμογές για υλοποίηση ΤΝΔ και παρουσιάστε μια σύντομη περιγραφή τους και τρόπο χρήσης*

https://playground.tensorflow.org

Um, What Is a Neural Network?

It's a technique for building a computer program that learns from data. It is based very loosely on how we think the human brain works. First, a collection of software "neurons" are created and connected together, allowing them to send messages to each other. Next, the network is asked to solve a problem, which it attempts to do over and over, each time strengthening the connections that lead to success and diminishing those that lead to failure. For a more detailed introduction to neural networks, Michael Nielsen's Neural Networks and Deep Learning is a good place to start. For a more technical overview, try Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.

This Is Cool, Can I Repurpose It?

Please do! We've open sourced it on GitHub with the hope that it can make neural networks a little more accessible and easier to learn. You're free to use it in any way that follows our Apache License. And if you have any suggestions for additions or changes, please let us know.

We've also provided some controls below to enable you tailor the playground to a specific topic or lesson. Just choose which features you'd like to be visible below then save this link, or refresh the page.

What Do All the Colors Mean?

Orange and blue are used throughout the visualization in slightly different ways, but in general orange shows negative values while blue shows positive values.

The data points (represented by small circles) are initially colored orange or blue, which correspond to positive one and negative one.

In the hidden layers, the lines are colored by the weights of the connections between neurons. Blue shows a positive weight, which means the network is using that output of the neuron as given. An orange line shows that the network is assiging a negative weight.

In the output layer, the dots are colored orange or blue depending on their original values. The background color shows what the network is predicting for a particular area. The intensity of the color shows how confident that prediction is.

What Library Are You Using?

We wrote a tiny neural network library that meets the demands of this educational visualization. For real-world applications, consider the TensorFlow library.

Credits

This was created by Daniel Smilkov and Shan Carter. This is a continuation of many people's previous work — most notably Andrej Karpathy's convnet.js demo and Chris Olah's articles about neural networks. Many thanks also to D. Sculley for help with the original idea and to Fernanda Viégas and Martin Wattenberg and the rest of the Big Picture and Google Brain teams for feedback and guidance.

https://www.nest-simulator.org/

NEST is a simulator for spiking neural network models that focuses on the dynamics, size and structure of neural systems rather than on the exact morphology of individual neurons. The development of NEST is coordinated by the NEST Initiative.

NEST is ideal for networks of spiking neurons of any size, for example:

1. Models of information processing e.g. in the visual or auditory cortex of mammals,
2. Models of network activity dynamics, e.g. laminar cortical networks or balanced random networks,
3. Models of learning and plasticity.

https://cs.stanford.edu/people/karpathy/convnetjs/

ConvNetJS is a Javascript library for training Deep Learning models (Neural Networks) entirely in your browser. Open a tab and you're training. No software requirements, no compilers, no installations, no GPUs, no sweat.

The library allows you to formulate and solve Neural Networks in Javascript, and was originally written by @karpathy (I am a PhD student at Stanford). However, the library has since been extended by contributions from the community and more are warmly welcome. Current support includes:

- **Common Neural Network modules** (fully connected layers, non-linearities)
- Classification (SVM/Softmax) and Regression (L2) **cost functions**
- Ability to specify and train **Convolutional Networks** that process images

- An experimental **Reinforcement Learning module**, based on Deep Q Learning.

Head over to Getting Started for a tutorial that lets you get up and running quickly, and discuss Documentation for all specifics.

21. Αναζήτηση σύνθετων εφαρμογών ΤΝΔ στην ειδικότητα

Α) Αναζητήστε στο Διαδίκτυο*: Σύνθετες υλοποιήσεις ΤΝΔ σε συνδυασμό με άλλες μεθόδους, σε εφαρμογές συναφείς με την ειδικότητα σας και παρουσιάστε μια σύντομη περιγραφή τους.*

Automatic Retail Product Identification System for Cashierless Stores

The purpose of this research in this project is to design and build an end-to-end retail product identification system. This system will provide cashierless stores with a smart checkout in which the purchases are automatically recognized without the need for scanning the barcodes or neither standing in long queues. Besides, this system will help self-checking stores to identify fraudulent customer behaviors, such as putting false barcodes on expensive products in order to scan them and get the price difference.

The research questions that this project aims to answer are mainly four issues. Firstly, the appropriate architecture of an end-to-end retail product identification system was intended to be determined by using text analysis and classification. Afterwards, using text classification as basis of the system, an investigation was done about the suitable representation and features of the text to use as input of the deep learning model. Then, to evaluate the designed system,there is a push to select the performance metrics, and finally finding out how to boost the performance of the system and achieve high efficiency. This final issue involves the problem of retraining GloVe embeddings to get new embeddings including all the vocabulary of dataset in this project.

In this project, design science research method was chosen as basic method for study. Furthermore, text classification was used as the fundamental technique for resolving the automatic retail product identification task. Besides, a pretrained GloVe model was also used as embedding method. Word2Vec and Mittens are the retraining tools that used to get full representation of text dataset. In general, the proposed solution includes text extraction from OCR, text preprocessing, model building and training and the main component which is the classifier. In the test and inference stage, the trained model is used for prediction of test dataset samples and confusion matrix was used, which are precision, recall, F1-score and accuracy, to evaluate the performance of the system. In addition, the LSTM-based model was compared with RMDL from three perspectives: runtime, memory usage and accuracy.

Finally, an end of conclusions was drawn that RMDL has the better performance for prediction. However, the solution based on LSTM is obviously better than RMDL in terms of runtime and memory requirements. To boost the performance of the system and to increase the accuracy, the GloVe are retrained using Word2Vec and Mittens

The solution for ARPIS based on a deep learning model

While considering the main problem of recognition of retail products with similar appearance for ARPIS, an idea is come up with of converting image recognition into text classification, for that even if retail products can have similar appearance, they always have different descriptions in texts on their packagings as long as they are different retail products. Based on this idea, Optical character recognition (OCR) is adopted as assistance to scan character and to extract them from retail products packaging so that these texts can be stored in editable form for further use. Modi and Parikh (2017) made an exhaustive review on optical character recognition and the conclusion showed a variety of reliable techniques for OCR with good performance for either texts in handwriting or natural scene images.

Once texts are obtained from retail products packaging and are stored in editable form, they can be used as datasets for a deep learning model. Since text classification is the main idea of classifying these description from retail products packaging into distinctive labels so that the retail product can be recognized, it is important to choose proper techniques for text classification.

https://www.diva-portal.org/smash/get/diva2:1559911/FULLTEXT01.pdf

22. Εφαρμογές CNN στην ειδικότητα

Α) Αναζητήστε στο Διαδίκτυο*: Προηγμένες εφαρμογές CNN, συναφείς με την ειδικότητα σας και παρουσιάστε μια σύντομη περιγραφή τους.*

Application of new advanced CNN structure with adaptive thresholds to color edge detection

Color edge detection is much more efficient than gray scale detection when edges exist at the boundary between regions of different colors with no change in intensity. This paper presents adaptive templates, which are capable of detecting various color and intensity changes in color image. To avoid conception of multilayer proposed in literatures, modification has been done to the CNN structure. This modified structure allows a matrix *C*, which carries the change information of pixels, to replace the control parts in the basic CNN equation. This modification is necessary because in multilayer structure, it faces the challenge of how to represent the intrinsic relationship among each primary layer. Additionally, in order to enhance the accuracy of edge detection, adaptive detection threshold is employed. The adaptive thresholds are considered to be alterable criteria in designing matrix *C*. The proposed synthetic system not only avoids the problem which is engendered by multi-layers but also exploits full information of pixels themselves. Experimental results prove that the proposed method is efficient.

Highlights

► We model a new CNN structure which avoids multi-layers in color edge detection. ► Adaptive thresholds are used in designing CNN templates. ► Experimental results are better than classic edge detectors.

Comments and conclusions

A new synthetic system for color edge detection based on CNN has been presented. The proposed system avoids the constraint of multilayer structure in color image detection based on CNN. It has avoided directly inputting multi-dimension data by using change information of pixels to be control parts. Therefore, it transforms multi-dimensional data into one dimension. Additionally, thresholds have been determined according to human vision system. Therefore, it makes the templates more suitable for pixels. Experimental results have shown that the proposed method performs better than classic methods such as Sobel and Prewitt in color image detection. The designing method of templates in this paper also gives us inspiration: in some special tasks using CNN, modification to CNN structure can be done according to specific task. In this way, we need not directly designing proper A, B templates.

Sustainability Analysis and Market Demand Estimation in the Retail Industry through a Convolutional Neural Network

The Chinese retail industry is expected to grow dramatically over the next few years, owing to the rapid increase in purchasing power of Chinese consumers. Retail managers should analyze the market demands and avoid dull sales to promote the sustainable development of the retail industry. Economic sustainability in the retail industry, which refers to a suitable return of investment, requires the implementation of precise product allocation strategies in different regions. This study proposed a hybrid model to evaluate economic sustainability in the preparation of goods of retail shops on the basis of market demand evaluation. Through a grid-based convolutional neural network, a regression model was first established to model the relationship between consumer distribution and the potential market demand. Then, another model was proposed to evaluate the sustainability among regions based on their supply-demand analysis. An experiment was conducted based on the actual sales data of retail shops in Guiyang, China. Results showed an immense diversity of sustainability in the entire city and three classes of regions were distinguished, namely, high, moderate, and limited. Our model was proven to be effective in the sustainability evaluation of supply and demand in the retail industry after validation showed that its accuracy reached 92.8%.

23. Ανοιχτά/Διαθέσιμα datasets για ΤΝ

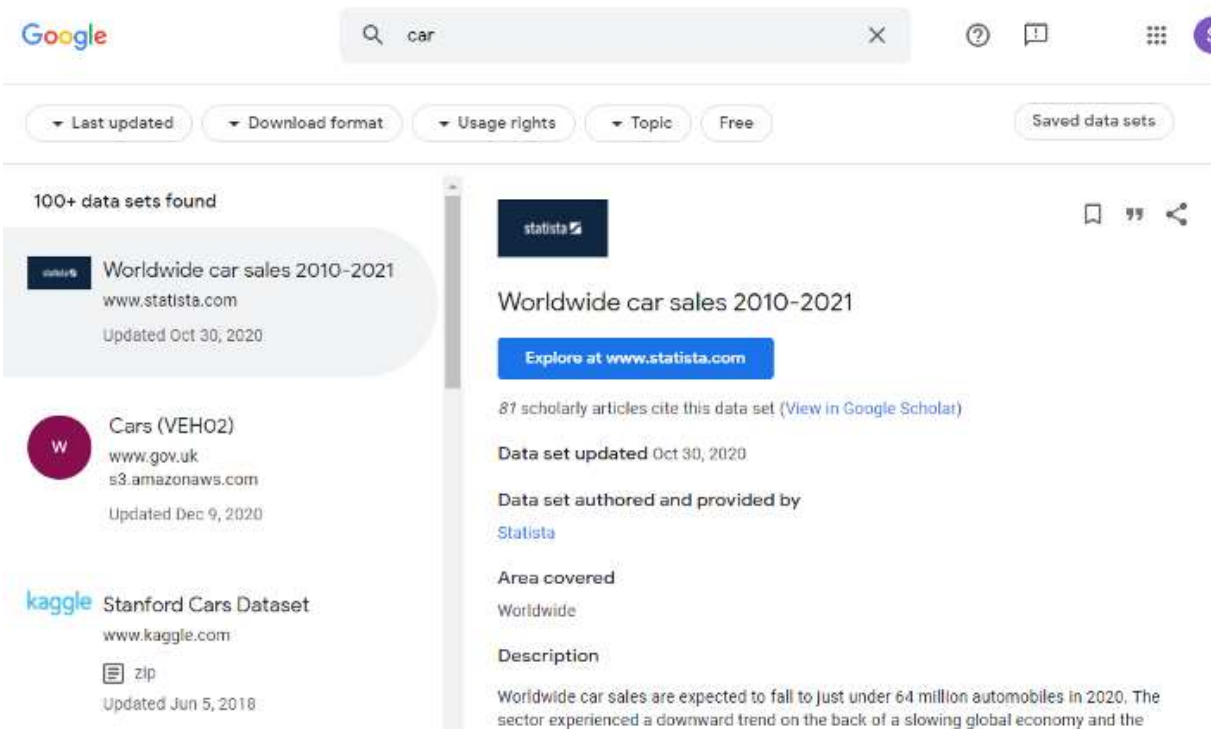
Α) Αναζητήστε στο Διαδίκτυο*: Βάσεις Δεδομένων με datasets που είναι διαθέσιμα για επίλυση προβλημάτων με μεθόδους ΤΝ και παρουσιάστε μια σύντομη περιγραφή τους.*

1. Google's Datasets Search Engine

Domain: https://datasetsearch.research.google.com/

As with Google's core product, you can easily search for the datasets using text. Additionally, you can filter the query by date, data format, and usage rights. The datasets on this website range from real-life datasets provided by companies for a price to free to use datasets for personal projects.

If you are looking for a great overview of all datasets available without any specific constraints google is the best place to start.



Screenshot from Google Dataset Search Engine

2. Kaggle Datasets

Domain: https://www.kaggle.com/datasets

If you have ever done any data science-related courses or hackathons you probably came across Kaggle. Kaggle is the world-leading platform for all Data Science related programming. It also allows users to find and publish data sets, and more importantly work and compete with other data-science people on how to extract value from them.

If you are trying to learn more about a specific type of problem and want to discuss the learning with Data Scientists all around the world kaggle is the place for you.

3. Earth Data

Domain: https://earthdata.nasa.gov/

For those of you who like to have a high-level overview Earth Data from Nasa is the right place. It features the probably largest collection of geo-related datasets about the earth, climate and water bodies.

The datasets are provided and created by researchers and institutions around the world and surely of the highest quality available in the respective fields. If you are looking for a project with a focus on time series or geospatial data, this surely is the best place to start looking.

Screenshot from Earth Data

4. Amazon and Microsoft Datasets, Azure and AWS

Domain AWS: https://registry.opendata.aws/

Domain Azure: https://azure.microsoft.com/en-us/services/open-datasets/catalog/?q=

The big tech giants feature datasets from all around the world in their open data registries. I made it a joint place because while they do not feature a large variety of datasets, they feature some especially big datasets.

Their experience in cloud and big data storage surely comes in handy when making such datasets available to the public. Currently AWS features around 200 datasets and Azure around 20.

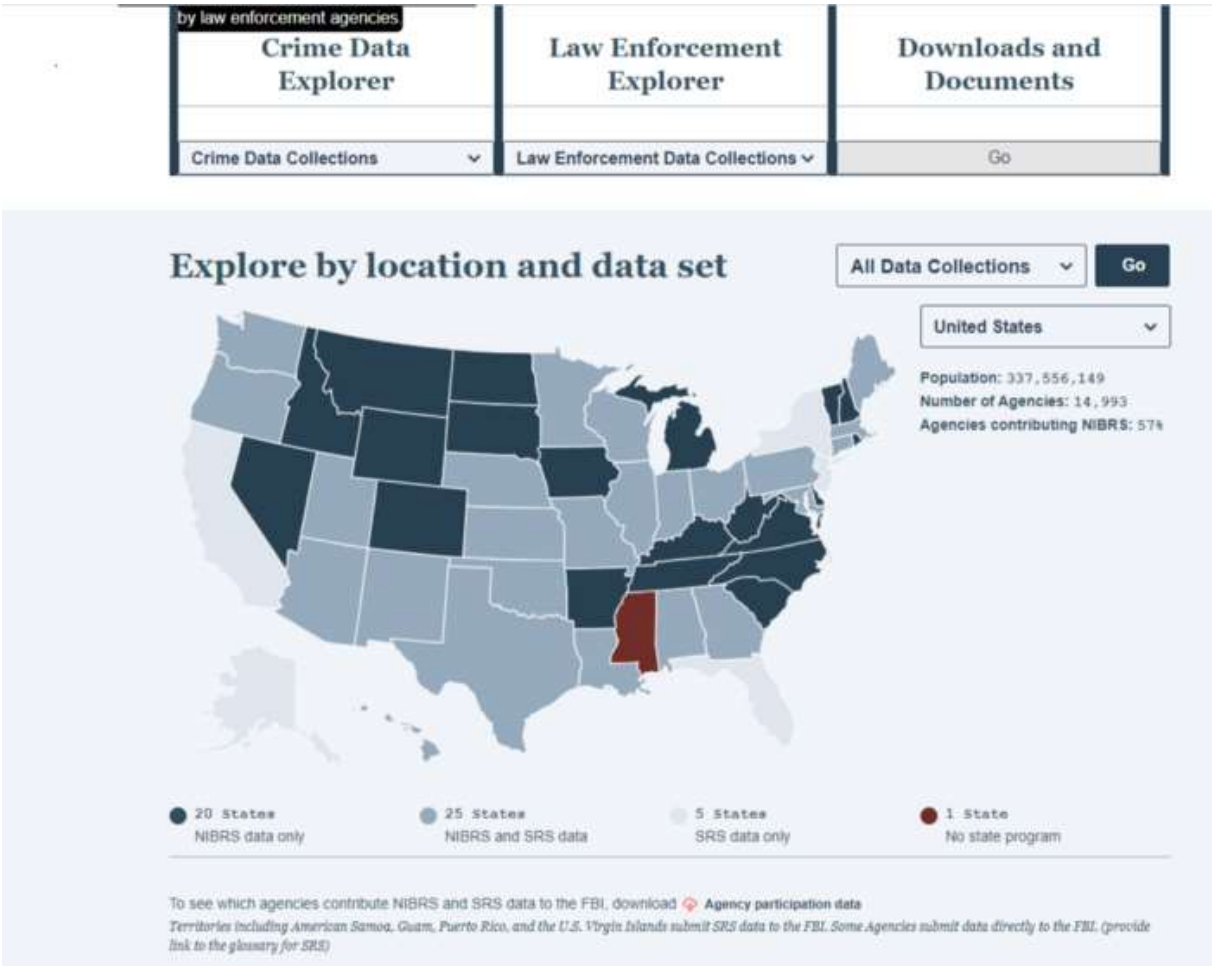
These places are the best if you are looking for a project in the Big Data realm and want to work with huge amounts of data.

5. FBI Crime Data Explorer

Domain: https://crime-data-explorer.fr.cloud.gov/downloads-and-docs

If you ever wonder what happens to those that do not comment their code well, the FBI crime data explorer might give you a hint. Probably the biggest data collection around criminal, and noncriminal, law enforcement data. It features data from state based crime up to human traffic related data.

While this generally is a sad story it is also one of the most exciting types of data. If you are looking for a change and a new exciting project that is a little bit different, it surely is a gold mine.



Screenshot from FBI Data Explorer licensed as CC0

6. Data World

Domain: https://data.world/

A collection that is rarely mentioned is Data world. It's very similar to the Google dataset search engine. What I however find very pleasant about this implementation is the search depth, when entering a query it does not only show the dataset itself but also subfiles that might contain the desired data. This can of course be particularly useful when looking for secondary data such as demographics and geographic location collections.

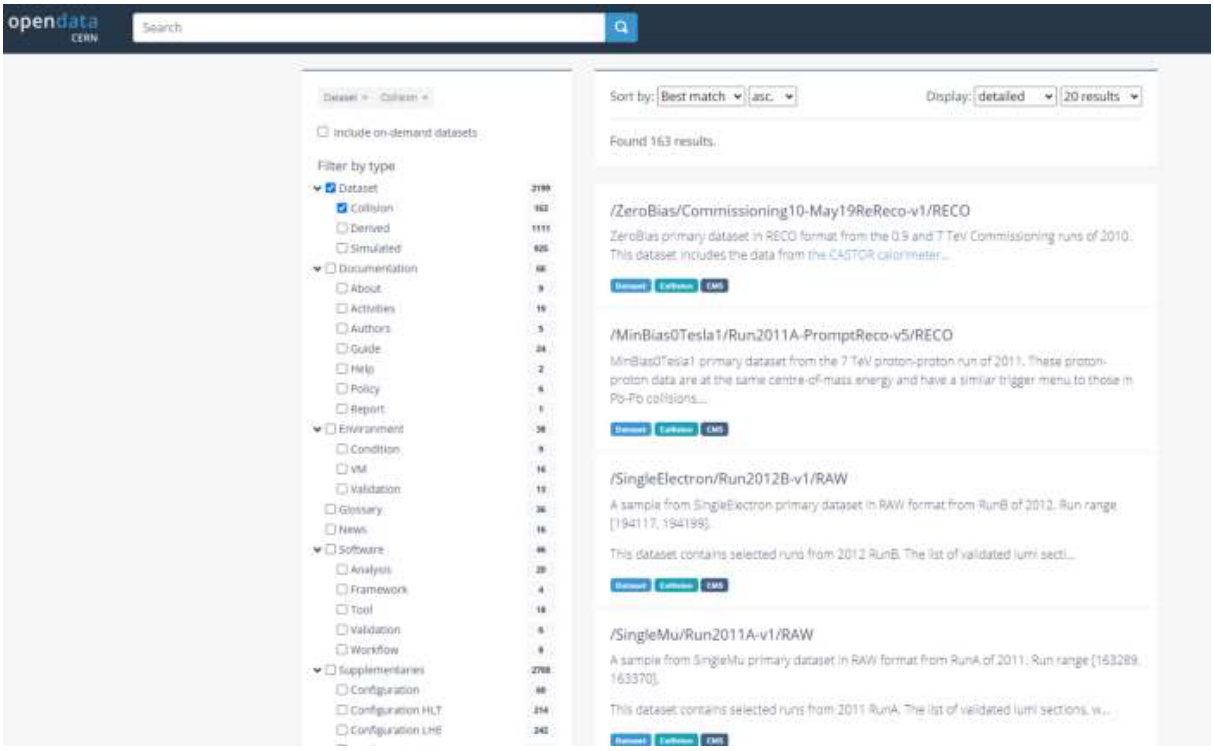
If you are looking for a dedicated website that has data in their name, Data World comes highly recommended.

7. CERN Open Data Portal

Domain: http://opendata.cern.ch/

The European Organization for Nuclear Research(CERN) located close to Geneva has made many of their incredible research data available to the public.

CERN's Open Data portal is fascinating. They collected and made available over two petabytes of data on the smallest things possible, particle physics. This is one of Europes most prestigious research institutions, and their data quality on particle collisions can't be met by anyone.



Screenshot from Open Data Cern licensed as CC0

8. Lionbridge AI Datasets:

Domain: https://lionbridge.ai/datasets/

Lionbridge is a company that provides services around data collection, annotation, and validation. Among other things custom labeling environments and what we are interested in today a variety of dataset you can find through their website.

On their dataset section they show you several articles containing various sources. Such as the '11 Best Climate Change Datasets for Machine Learning' and 'The 50 Best Free Datasets for Machine Learning'. Since they are a company build around datasets their recommendations are surely great.

Best place if you are looking for a comparison between specialized datasets.

9. UCI Machine Learning Repository

Domain: https://archive.ics.uci.edu/ml/index.php

The University of California, Irvine maintains over 550 datasets which are free for you to use. I find this website to be particularly interesting for educational purposes since it offers filtering by

the problem. So classification, regression, and clustering, you can easily find a dataset that would work well with the technologies that you are currently exploring.

Apart from knowing how to educate people their team surely knows a lot about Machine Learning datasets and how to evaluate them.

24. Εργασία – Project:

Δίνεται από μια εργασία (dataset) ανά ομάδα εκπαιδευομένων (ομοειδούς ειδικότητας) όπου θα υλοποιηθούν οι κυριότεροι αλγόριθμοι/μέθοδοι (prepprocessing, regression, classification, clustering κλασικές και με ΤΝΔ).

https://archive.ics.uci.edu/ml/datasets.php

https://archive.ics.uci.edu/ml/machine-learning-databases/census1990-mld/

file:///C:/Users/jasproudis/Downloads/USCensus1990-desc.html

Θέλω να εκτιμήσω συσχετίσεις μεταξύ πολιτειών και μέσου όρου εισοδημάτων να κάνω ταξινόμηση και στην συνέχεια να δοκιμάσω ομαδοποίηση με βάση το εισόδημα χωρίς επίβλεψη με χρήση τεχνητών νευρωνικών δικτύων.