# Algorithms for Data Mining and Machine Learning in BADA

BADA main study phase 1, Vinnova FFI 2015-00677

Ian Marsh, Bjorn Bjurling, Ahmad al-Shishtawy, Anders Holst

# 1 Introduction

This report gives an overview of the data mining (DM) and machine learning techniques (ML) employed in the BADA project at RISE. Our primary goal is to provide the project partners with the rationale behind choices, decisions, and experience from working with data connected to the road traffic and vehicle industry. The focus is on *techniques* and *technologies* that we have implemented in three use cases. We will, in addition cover further mining and machine learning techniques because of their importance in the field of data analysis.

The report provides an account for decisions to why we made certain technology choices, lessons learned, and important pointers to a deployment of the platforms and algorithms. In particular, the analysis of the use cases have been made in the HopsWorks[1] framework and deployed on the RISE SICS North data center (ICE). ICE is an infrastructure and cloud research and test environment owned and operated by RISE-SICS North. The facility is open to use primarily for European projects, universities, and companies. However, customers and partners from all over the world are welcome to use the data center for their testing and experiments.


**Fig. 1: The SICS RISE data center in Luleå**
https://www.sics.se/groups/rise-sics-north

HopsWorks is an open-source data processing platform developed and supported by researchers at SICS and the new startup Logical Clocks AB[2]. HopsWorks is one of the services provided at SICS ICE data center[3]. The following factors favored our choice of the platform:

1. Hops is available as a PaaS (Platform-as-a-Service) which means we can focus on the problem we are trying to solve instead of spending significant amounts of resources on installing and managing a highly complex infrastructure for building our own platform..
2. Hops includes all major Big Data platforms and tools that we need and more. Such as Hadoop HDFS, Spark, Flink, Kafka, Distributed TensorFlow with GPU support. All tools are available in a user-friendly web interface for accessing the services and Notebooks for interactive analytics.

[1] http://www.hops.io
[2] http://www.logicalclocks.com
[3] www.hops.site

3. Hops platform adds Multi-tenancy to Big Data platforms. This allows us to share data sets and data processing pipelines with several tenants without exposing data and processing from one tenant to another.
4. Hops has built-in security. It adds encryption to all communications inside and outside the data center and applies fine-grained access control and isolation between tenants.
5. Hops follows and contributes to the state-of-the-art and open source development of Big Data Platform. For example, Hops HDFS contributions has won the prestigious CCGRID 2017 Scale Challenge[4] and Hops has significant contributions for enabling distributed TensorFlow with GPUs on top of Spark[5].

There are many resources available, such as courses, books, and videos, giving excellent introductions to Data Mining, Machine Learning, and Big Data. One can find good introductions both for the mathematically inclined and for those with other professions for example at Coursera [**Coursera**] or in Oreilly's book series [**Oreilly**]. We shall therefore keep this document to a bite-sized format with very few references to mathematics. The audience for this document is the technical person but without specific knowledge in either data mining or machine learning.

[4] https://www.sics.se/media/news/rise-sics-and-kth-winners-of-ieee-scale-challenge-award-2017
[5] https://spark-summit.org/eu-2017/events/apache-spark-and-tensorflow-as-a-service

# 2 Preliminaries

Before we commence on the techniques within BADA, we introduce Data Mining and Machine Learning, first as entities on their own, then within road traffic scenarios.

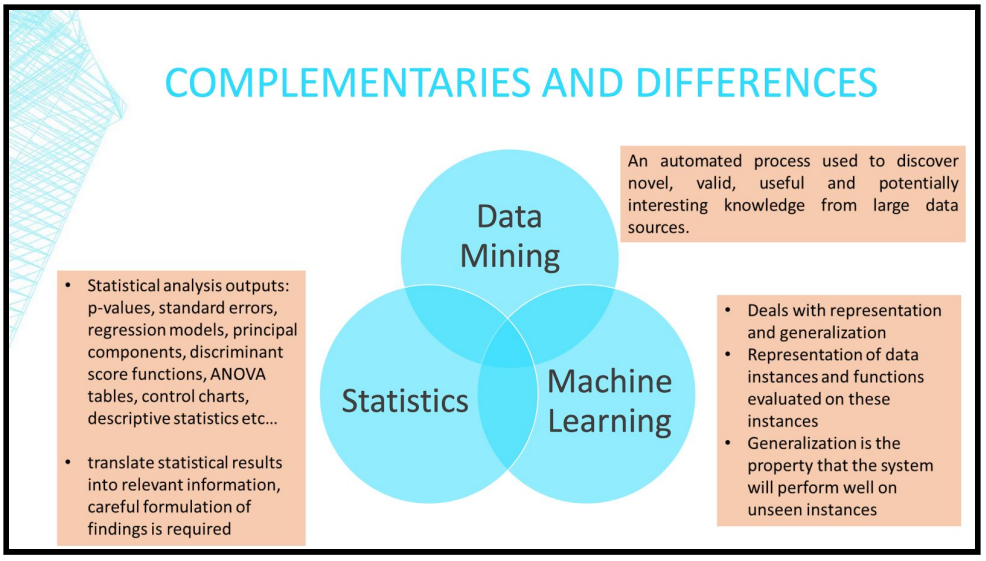



**Fig. 2: Differences and similarities between statistics, data mining and machine learning**

**Data mining** is about finding patterns in data, typically so that one can explain some phenomenon. Data mining is usually carried out by a *person*, in a specific situation, on a particular data set, with a set goal in mind. Quite often, the data set is massive and complicated. Moreover, data mining procedures are either *unsupervised*, 'we don't know the answer yet' or *supervised* 'we know the answer' [Table 1]. As an example, in BADA, data mining was performed to find common congestion bottlenecks from 11 years of traffic data. Data mining techniques include cluster analysis, classification and regression trees, and neural networks.

**Machine Learning** uses algorithms to build models of what is happening behind some data so that it can predict future outcomes. What distinguishes these algorithms is that predictions based on the model improve as amount of data processed by the algorithm grows. In other words, machine learning involves the study of algorithms that can extract information automatically typically without on-line human guidance. Training of ML algorithms benefits greatly from Big data. in BADA, we used machine learning for training a model to recognize which conditions lead to unexpected queue accumulation on road networks. Common machine learning techniques include cluster analysis, classification and regression trees, and neural networks.

The two areas of DM and ML overlap to a great extent with each other and also with Statistics (See Figure 2). Most algorithms used in Data analysis, and in particularly in BADA, belong typically to all three areas. In the end of queue detection use case, we could have chosen a neural network to learn the characteristics of queue-ends, or equally well we could have mined the data for certain patterns characterizing queue-ends (see Fig. 6), which we also ended up doing. The confusion sometimes surrounding the terms Data Mining and Machine Learning stems from historical use.

## 2.1 Data mining and machine learning in a road traffic context

Data mining can be used on road accident data to analyse road accidents. With Data mining techniques, one can for example find accident prone locations by clustering. In the Hazard Warning Use Case, clustering can be used to build a model that subsequently can be used for classifying the type of hazard warning issued by a driver by pressing the hazard warning button. In BADA, we used the location of a vehicle when the hazard warning button was pressed to determine the likely cause for the issue. Our simplistic model classified ‘rural presses’ as accident related and ‘urban presses’ as parking indicators.

A common way of approaching the problem of finding the classes in the Hazard Warning UC is to use a clustering algorithm (eg K-means) and count the button presses in each cell in grid laid out over a map. This would give K groups based on their frequency counts. Then an association rule mining could be applied to reveal the correlation between different attributes in the press data and location characteristics. The same technique could be applied to other traffic scenarios.

```
If weather = warm
     then slippery = normal
If slippery = normal and windy = false
     then driving = safe
If visibility = clear and driving = safe
     then weather = hot
If windy = false and driving = safe
     then visibility = clear and slippery = false
```

**Fig 3: Association rule: predicts values of arbitrary attribute (or a combination)**

Rather than mining large datasets, machine learning builds models of the data. An example is to learn density variations of road traffic. Here, the data could be observations of the number of cars at certains locations and time-points. ML could be used to build a model of the normal behaviour. New observations of densities deviating from the normal behaviour can subsequently be detected by deploying

statistical measures and inference. Such readings can be used for inferring for example traffic jams or accidents. For example, high densities of vehicles 1am on the E4 around Kista may constitute a deviation from the normal.

# 3 The use cases

In this section we discuss the work with the three BADA use cases. The work was guided by the following principles.

1. *Simplest algorithm and processing choice available*. This was to take incremental, well understood steps, and also to decide if that solution would be the best to deploy within industry.

2. *Open platforms*. In order to convey result and experiences confidently we need to know how the tools are implemented. Moreover, most state-of-the-art data analysis tools are open source (Kibana, ElasticSearch, Spark, Flink, Python,…)

3. *Synthetic data and open data*. In some cases we needed more data than was provided.

4. *Use of Python.* Now many programmers are available, PySpark is available, as well as many libraries we need, SciPy, NumPy and plotting with Matplotlib

## 3.1 Hazard warning : classification of streaming data

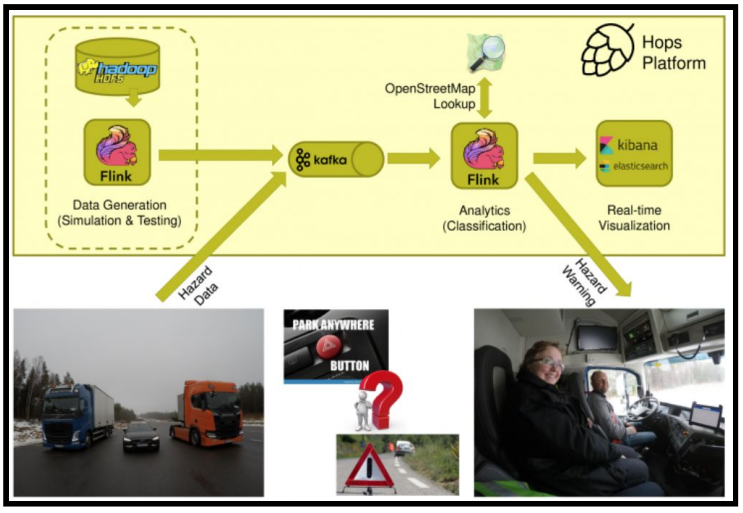



**Fig. 3: A streaming analytics platform for traffic signaling.**

The architecture designed for the hazard warning BADA scenario. The datasource is a HDFS file containing a list of geolocations. They are read by a Flink process that applies a time delay to simulate an event stream and writes the data back to a `kafka` topic. Another Flink process consumes the topic. For each record, it contacts an external `openstreetmap` server to find the language surrounding the geocoordinate. Based on the result it flags the record to be in a rural, residential or if not found, unknown area. The results are pushed into `elasticsearch`. The results can then be queried and visualized using `kibana`. All

components, besides the external server, are executed on our Hopworks cluster (SICS-RISE technology).

In this case, we used *generated* data. We needed large amounts of data for testing the streaming platform. We used the GPS locations to simulate that hazard warning button was pressed. Then used OpenStreetMaps to check if this location is either rural/urban and used that to classify the event into hazard/parking. While the classification task is simplistic, we used the use case to demonstrate deployment of a state-of-the-art streaming platform and the use of scalable data analysis algorithms. Essentially, to find the classes one can use clustering algorithms. These clusters can subsequently be used for defining the classes in the classification task. this is a clustering problem, to ascertain the location of the hazard press.

An extension of this use case could be to repeat the same experiments with larger datasets, for example using speed, steering, and suspension data, using the platforms developed in BADA. Occasionally privacy issues are encountered, however, systems can be built and then used in company's internal systems. Open source software tends to follow this pattern, the code is developed in the public domain, and then used and modified internally. Similar systems are running in Berlin, using a recent API developed for Flink[6] implementing the connected car event stream[7], within the EnviroCar project[8].

## 3.2 Traffic safety : *Fusing* location and accident information

In this use case we make use of several dataset by fusing them. By fusion, we can enrich analysis and interpretation of a dataset. For example, accident data can be enriched by weather and GPS. This may lead to improved the quality of the analysis.

The datasets used for this use case are acquired from STRADA information system. Every recorded accident in STRADA has attributes such as `report id, type of accident, the location` of the accident by coordinates, accident description, date of registration, etc. These accident reports are recorded by police and hospital from 2003 to 2015. There are two datasets be analysed in BADA. The first dataset is accident descriptions recorded by the police, it contains 224,574 descriptions. The second dataset is accident description recorded by hospitals, and it contains 448,476 descriptions. All accident descriptions are short, ranging from 1-5 sentences.

[6] https://goo.gl/5mo5Tb (Flink 0.10.0)
[7] https://goo.gl/GE9Sup (Connected car stream)
[8] https://goo.gl/7nDQuL (EnviroCar project)

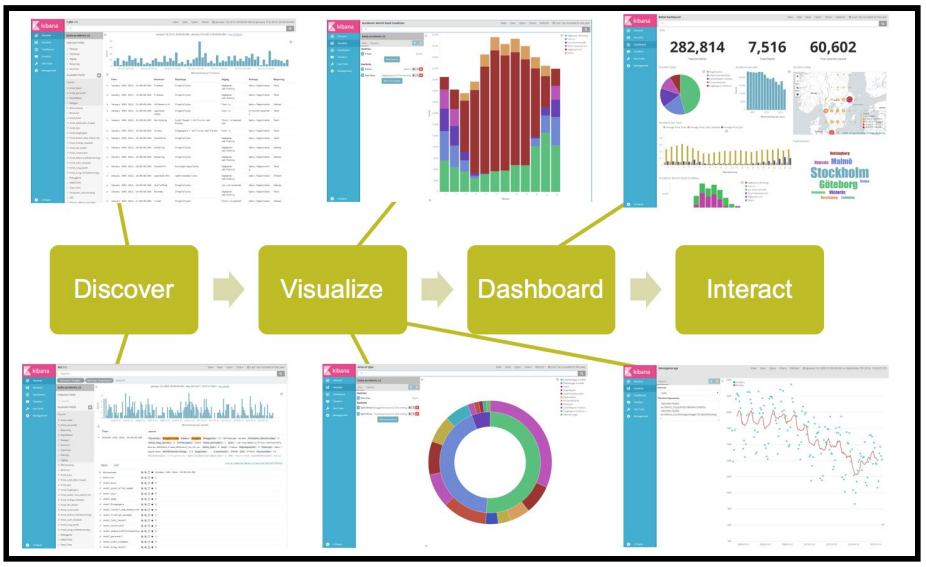



**Fig. 4: Data mining approach for vehicle safety in Sweden.**

We mined the national information system of road accidents in Sweden (STRADA). Once the data is read in and indexed using `logstash`, it can be searched with `Elasticsearch`, results were visualised with `Kibana`. We made use of Geolocation, where accidents were plotted on a map, aggregate statistics too, where stacked bar charts and pie-like summaries of the data as in Fig. 4.

In big data projects (as of course in any data analysis project), it is important to be able to explore the data as a part of the data analysis work. This use case shows how to visualize with Kibana and how to mine a distributed dataset using ElasticSearch. Mining is made via database like queries concerning the stored accident data. ElasticSearch allows queries to be answered in very short time whatever the query and whatever the size of the dataset. ElasticSearch distributes the query to the (geographically potentially dispersed) servers and then collects the answers into a format that can be displayed for the user.

Kibana also has a time series analysis feature called `Timelion`, for unsupervised ML. Using time series analysis, it is possible to see how accidents occur over time intervals, e.g. the work days/weekends (short), summer/winter months (medium) or over years (long). The STRADA data, was analysed with respect to trends and seasonality.

Accident data is well analysed in the literature. See for example [**TopicModelling**]. The exploratory aspect of this use case can be extended by using additional data sets, for example road friction information together with the weather in appropriate time intervals. We recommend extending this work by cross-referencing weather and road surface data(bases) as well as using car sales information.

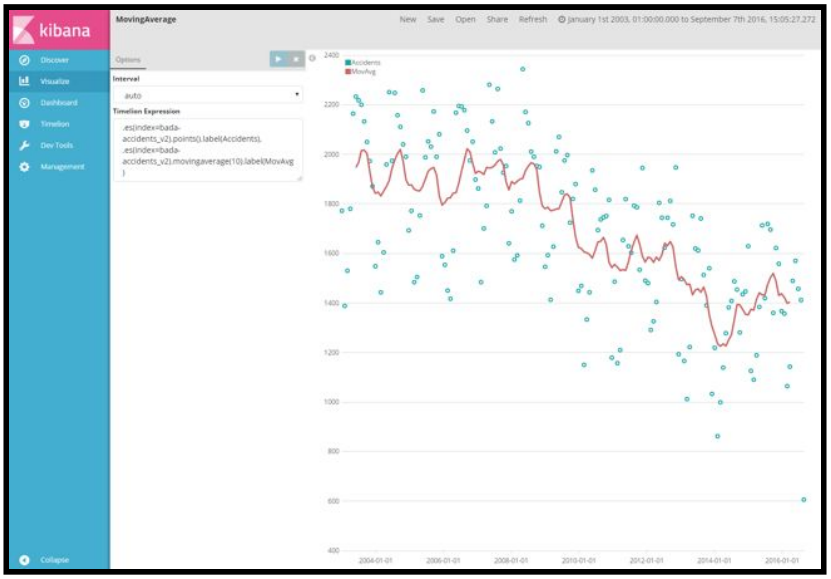



**Fig. 5: Time series visualisation of accidents in Sweden. The x-axis ranges from 2004-01-01 to 2016-01-01 (each label is 2 years). The y-range is [400, 2400].**

In the Traffic Safety use case, one can also make analyses of how the use of certain safety technologies can affect the risk of accidents. For this, we use *Causalimpact* designed by Google. It is a technique for working with time series data, to estimate the effect of a designed intervention of a process [**Causal**]. Thus, unlike typical time series analysis, Causalimpact may be used where one knows that some artifact was introduced in the process, typically producing a jump of some kind.

The advantage of using a Bayesian approach, which CausalImpact is an example of, is that one doesn't need to trade off a short lag in the Autoregressive (AR) and the Moving Average (MA) of a ARMA process. ARIMA is a variant of ARMA where the I refers to Integrated (the opposite of differencing). Either with short lags e.g. AR(1) and MA(1) one can follow a process quickly, but may miss the trends. Longer AR and MA processes (10-50) follow the trends well, but might miss jumps. The AR and MA can be different, but choosing them can be difficult without a model of the process. However, the CausalImpact approach achieves this without compromising either short or long term tradeoffs.

We currently are investigating this in BADA in connection with the introduction of ABS brakes influenced accidents frequency.

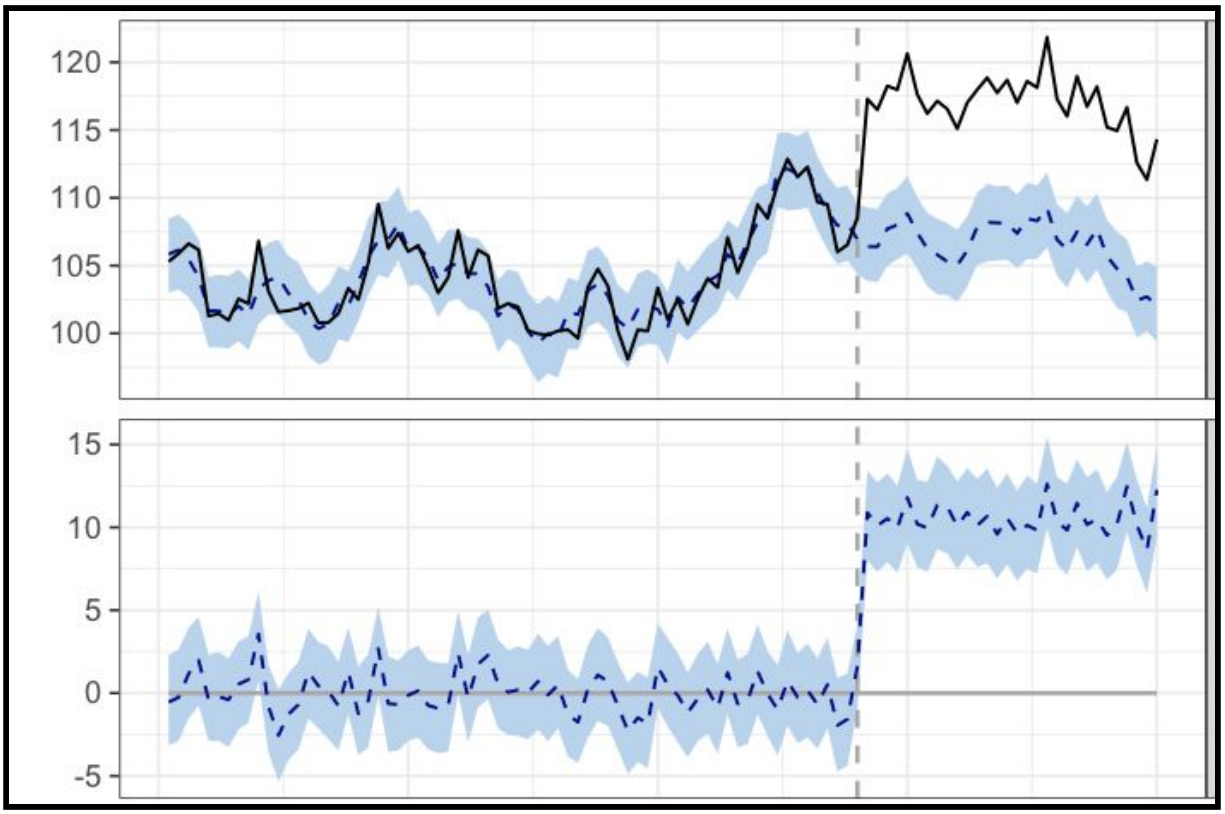



**Fig. 6: Estimation of a time series. The top figure shows prediction using a tradition (ARIMA) like model whilst the lower one using a Causal-approach. The data are the same in both plots, but the y-axes are not the on same range. Data is from a network delay trace taken from the [cheesepi.sics.se](cheesepi.sics.se) platform.**

## 3.3 End of queue (EoQ) detection : time series analysis

The EoQ problem uses all aspects of Data Mining, Machine Learning and statistics. The approach taken includes mathematical modelling based on a fundamental property of traffic flows, dense roads led to lower average speeds. EoQ also needs inference as the data available (Trafikverket), although broad in time, 11 years and national coverage, is still not individual vehicles, making EoQ detection difficult. The data is aggregated into 1 minute intervals and 300 meter road sections (the city detector separation).

A slightly different issue, is that there is no real ground truth to be tested against. As in some cases the end of the queue is quite obvious, whereas in other cases, a mild congestion event, i.e. concertining vehicles, might not be perceived as a queue.


**Fig. 7: An illustration of vehicles arriving at a busy road section.**

Given the large amounts of data, we need a big data framework in which to execute a solution. We use the Hadoop Ecosystem, for distributed storage and Spark, an in-memory data processing engine (see below)[9]. We used SQL-Spark queries on the data, which allows us to select and join fields in a database like manner.

We estimate the density of the traffic from the relation `flow = density * speed`, as we have the flow and speed in the original data. Density is a key indicator for queues, so this process is performed on the data. Sudden increases in density are cues for potential queues, as shown above in figure 7. Therefore, we have implemented a heuristic, threshold-based method. By plotting events over a day on Google maps, we obtain human feedback as an intuitive visualisation test, but it is not an automated EoQ algorithm, that can mine all large variations over the nation of traffic flows. This is work in progress.

We used an EoQ-detection algorithm, based on the fundamental law of traffic theory, which is essentially when density changes, but tracks when flow, density and speed *all change* to indicate a queue accumulation. This is probably the best indicator of queue accumulation.

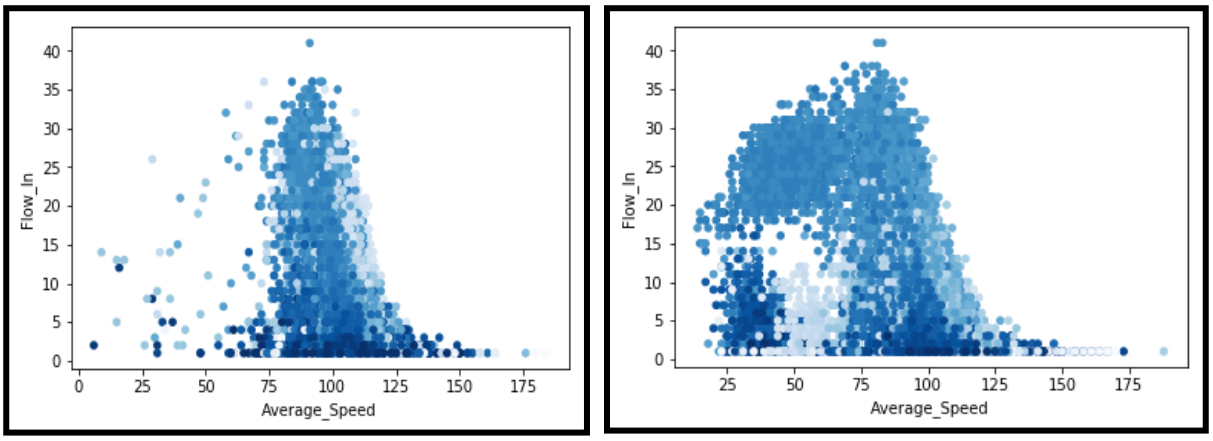



**Fig. 8: Illustrations of the relationship between traffic flow (y) against the average speed (x) of vehicles from the Trafikverket data.**

[9] https://goo.gl/C66Noy (Easy introduction to Spark data processing)

The processing procedure is we read flow data and importantly, transform the data into a format we can cross reference. Specifically, the detector coordinates were not in a format we could use (Ds_Reference for Trafikverket versus the commonly used GPS). Once converted, we can use the position with other systems, such as the SMHI weather database.

The traffic flow data from Trafikverket has been gathered over 11 years. It comprises about 7 billion entries, 0.4TB and covers 2060 detectors around Sweden. So as to obtain faster results in the 3rd use case, we used one month of data only, which is around 0.4GB, 8 million entries and covers November 2016. We chose November for the inclement weather in Sweden at this time of year, to see the result of varying weather on traffic flows. We also cross-references the flow data with GPS and made a start with the SMHI database.

# 4 Algorithms for data mining & machine learning

Methods for machine learning and data mining can be divided into five broad areas:

1. **Statistical methods**
2. **Case-based methods**
3. **Artificial neural networks**
4. **Logical-based methods**
5. **Heuristic search**

We know go through some of the important algorithms for data mining and machine learning in these categories. Opinions vary on the most important, one somewhat older source is [**Top10**] from 2016, but the usefulness of the algorithms is well described. We end this chapter with some some other important aspects of machine learning, i.e representation, validation, and machine learning packages.

| | ***Supervised Learning*** | ***Unsupervised Learning*** |
|---|---|---|
| ***Discrete*** | **Classification** | **Clustering** |
| ***Continuous*** | **Regression** | **Dimensionality Reduction** |

**Table 1: Tabular representation of data mining algorithms**

## 4.1 Statistical methods

Figure 2 shows the differences and similarities between Statistics, Data Mining and Machine Learning. *Classic statistics* is very important to understand how the input data varies, e.g. road traffic speeds, as well as the results from some mining or learning. Typical quantities such as the mean and variance are important summary statistics for the data, where many values can be literally summarised by a single value. Moving onto distributions of data, allows further analysis to be made,

quantiles, and how the data is distributed, e.g. towards the lower or higher ends of the range. Statistical tests (`Chi-squared, t-tests`, …) often need all the data.

Another simple example is using today's temperature to predict tomorrow's given what we know about today, we have *prior* information and a model of the likelihood. It is known as a *Bayesian approach* and uses the conditional probability law, in a slightly different form.

*Statistical learning* includes some of the most important modeling and prediction techniques. Statistical learning is based on probability theory and statistical modelling. The CausalImpact approach we briefly described above uses the Bayesian approach, updating the model. Other methods include Mixture models, hidden Markov models, kernel density estimators and particle filters. Statistical methods are further divided into parametric versus nonparametric methods – depending on whether the forms of the class distributions are known or not.

### 4.1.1 Regression (prediction)

Essentially regression is a prediction procedure. From Table 1 regression outputs continuous values. In statistical modeling, **regression** analysis is a set of statistical processes for estimating the relationships among variables. The inputs are also known as predictors or features. So classification and regression are similar processes but with different types of output, for example in a BADA context, the average speed through Gothenburg central.

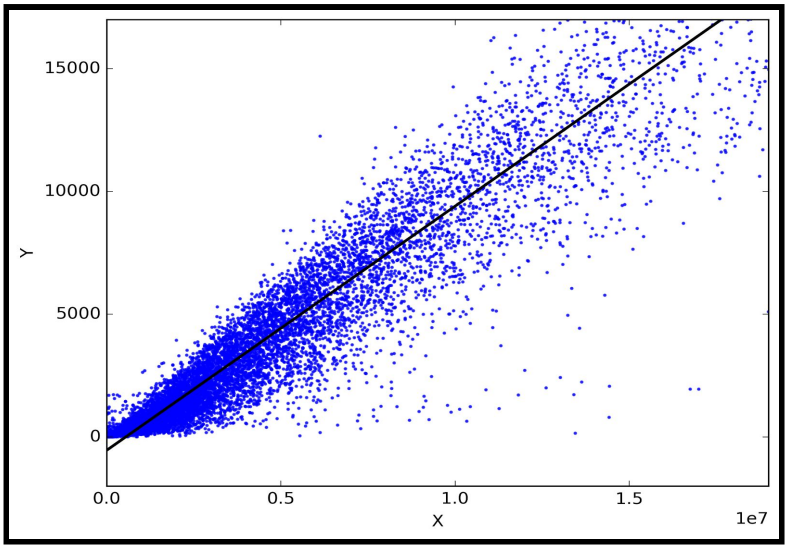



**Fig. 9: Finding a regression line between input variable (x) and an observed output (y).**

Finding relationships between the `input` and `output` variables, is the key idea behind regression. For example, how does vehicle density influence the average

flow or velocity, and by how much? Once such a relation is known, then one can make predictions based on this relationship. In mathematics, this relation is often known as a function `y = f(x)`. In this case, `x` is the input and `y` is the output and the job of regression is to find the function f.

### 4.1.2 Dimensionality reduction

Dimensionality reduction has been widely applied in many scientific fields. Reduction techniques are used to lower the amount of data from the original dataset and thus leave only the statistically relevant components in the processed data. Dimension reduction also improves the performance of classification algorithms, by removing noisy irrelevant data. In the ML community, dimensionality reduction is known as feature extraction. Principal Component Analysis (PCA) is ubiquitous in dimensionality reduction, it is computationally efficient, and parameterless. PCA has many variants: kernel PCA, nonlinear PCA, linear discriminant analysis see [**Jolliffe**].

PCA is a variance-based method for reducing the dimensionality of data. It uses an transformation (see below) to convert a dataset of correlated variables, into a set of uncorrelated variables, these are the principal components as mentioned above. The transformation ensures that the 1st component has largest variance, the 2nd the next highest variance, as can be seen in green in the right figure. The number of principal components will be always be less than or equal to the number of original components. Dimensionality reduction allows features that contribute little to the dataset to be removed [**Ian2017**].

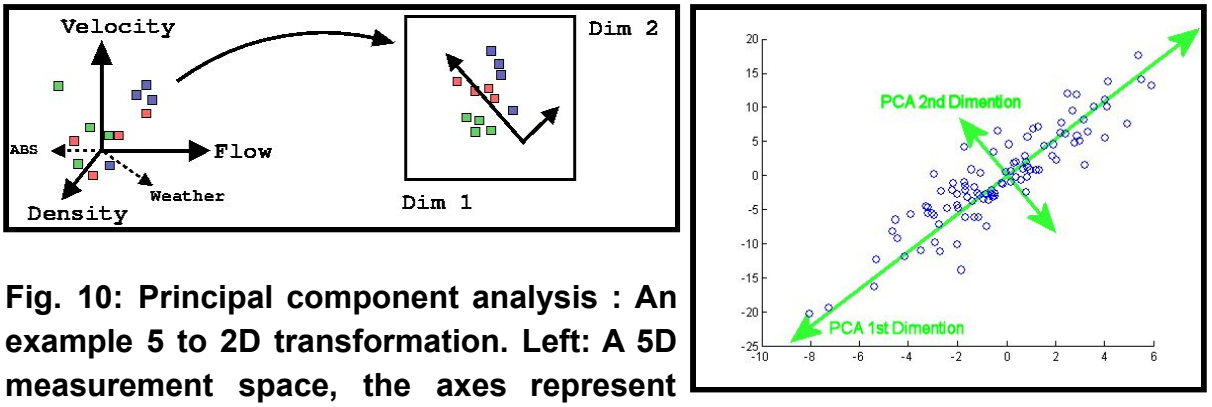



**Fig. 10: Principal component analysis : An example 5 to 2D transformation. Left: A 5D measurement space, the axes represent measured attributes, dotted lines represent labeled data, and coloured points represent measurement points. Each colour (RGB) represents one measurement campaign. Right: A 2D visualization of the same points.**

## 4.2 Case-based methods

Classification is the best known case-based methods. It is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. The basic idea is that similar patterns belong to the same class. They are easy to train, as one has to just save every pattern seen. The disadvantages include the model size increases with the number of examples seen and some notion of a distance (metric) is needed. For example, a classification model could be used to identify vehicles from a flow of vehicles where each is not known as a categorisation issue like a car or a truck. The hazard warning use case is an example of a binary classification problem.In a graphical representation (Fig 11), classification can be illustrated as trying to find a line (line for 2D and hyperplane in higher dimensions) separating the data point into classes.

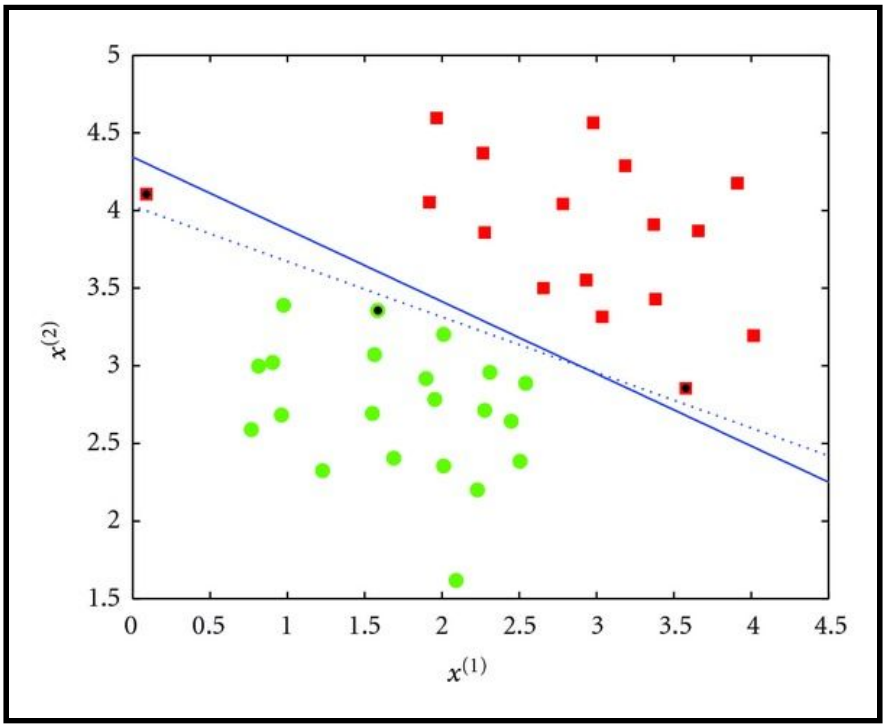



**Fig. 11: Classification problem, separating events into one of two classes or categories. Finding the decision boundary is one core issue as well as deciding there are 2 classes.**

K-means is the most popular *clustering* algorithm, as it is very efficient and easy to implement. The k-means algorithm belongs to a category of prototype-based clustering approaches. Prototype-based clustering means that each cluster has a representative called a centroid [**k-means**]. The procedures is ***1)*** the cluster centroids are initialized randomly and data points are assigned to the closest

centroid, *2)* the mean of current clusters is computed and the chosen centroids are moved to the mean point. *3)* This procedure is repeated until the centroids do not change, and the final state is called convergence. The main problem with vanilla k-means is usually selecting the initial number of clusters. Alternatives to k-means include nearest neighbour, k-nearest neighbour, and Gaussian Mixture Models.

## 4.3 Artificial neural networks

Inspired by the neural structure of the brain, neural networks (NN) are units connected by weights. Artificial refers to the neurons not being biological. Weights are adjusted to produce a mapping between inputs and outputs. In neural networks, one often sees the terms layers, where each layer represents and learns one feature. ‘Deep’ in deep learning typically refers to multiple layers, arranged in layers, so that the output of one layer is fed to the next one. Typically, lower layers process simpler features, and higher layers more complex features.

As a very simple example, in facial recognition in images, the lowest layer could recognise a line in a image, the next layer a facial outline, the next facial features, and the top layer an individual person.  Typically no manual work is done extracting the features, each is learned by the neural network. We discuss two deep neural network approaches, a recurrent network for time series-like data and a convolutional network for image-like data.

### 4.3.1 Long-short, term memory : a recurrent neural network example

Long short-term memory (LSTM) is a recurrent neural network (RNN) architecture that remembers values over arbitrary intervals, typically time. Stored values are not modified as learning proceeds. RNNs allow forward and backward connections between neurons. An LSTM is well-suited to classify, process and predict time series given time lags of unknown size and duration between important events. *This is almost a perfect match for traffic flow*, which needs only a time series approach in time, but also in space, as road networks operate in 2, sometimes 3 dimensions. Further, dimensions present in the data could include: weather, road conditions, tolls, accidents, speed limits and so on. Including dimensionality reduction might be needed, to reduce dimensions to only those relevant in the data set under examination.

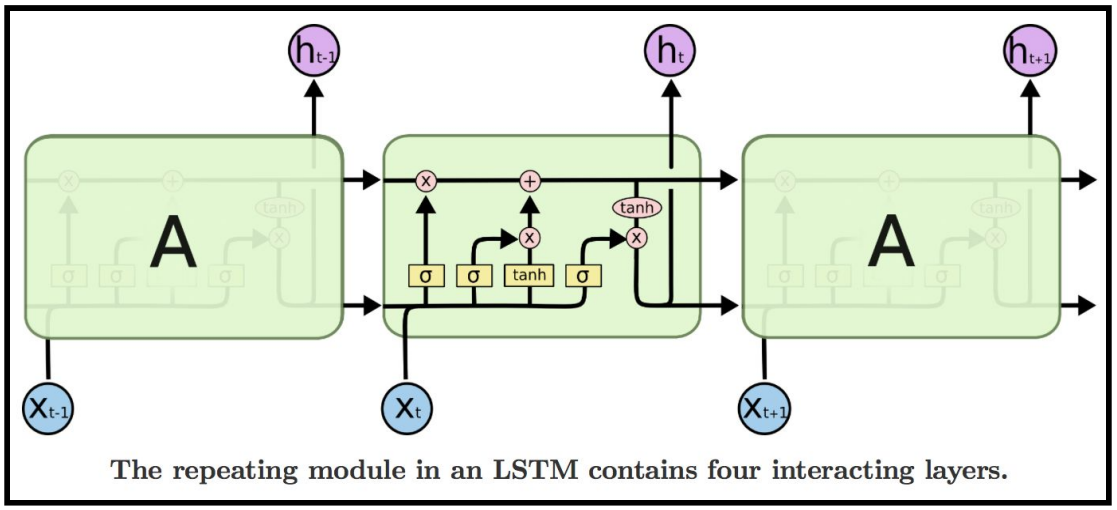
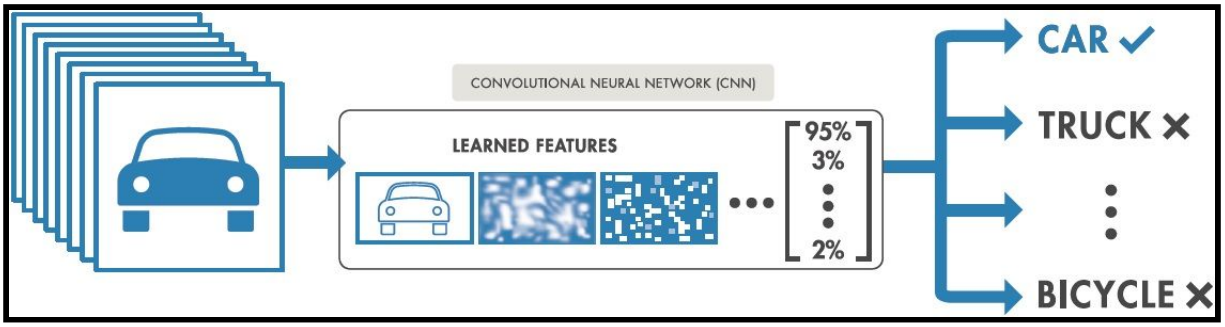



**Fig. 12: a LSTM representation**[10]

The relative insensitivity to gap length is an advantage for LSTM over RNNs and hidden Markov models and other sequence learning methods. The gap length in a BADA setting is the time intervals (min, hour, day, week) of the vehicle traffic.

### 4.3.2 Convolutional neural networks : image like data

Sometimes known as ConvNets, a convolutional neural network represents the data as a map. Some notion of distance is needed between the points on the map. An example is in the two images below, where the top figure left indicates each layer, of a vehicle. In the learned features, the first layer indicates a 93% probability of the image being a car. The other features in this case are noise, and probably not a car. Note, some noise in almost always present in all real-world images.




[10] https://goo.gl/fTU4u4 (Deep learning for sequences)

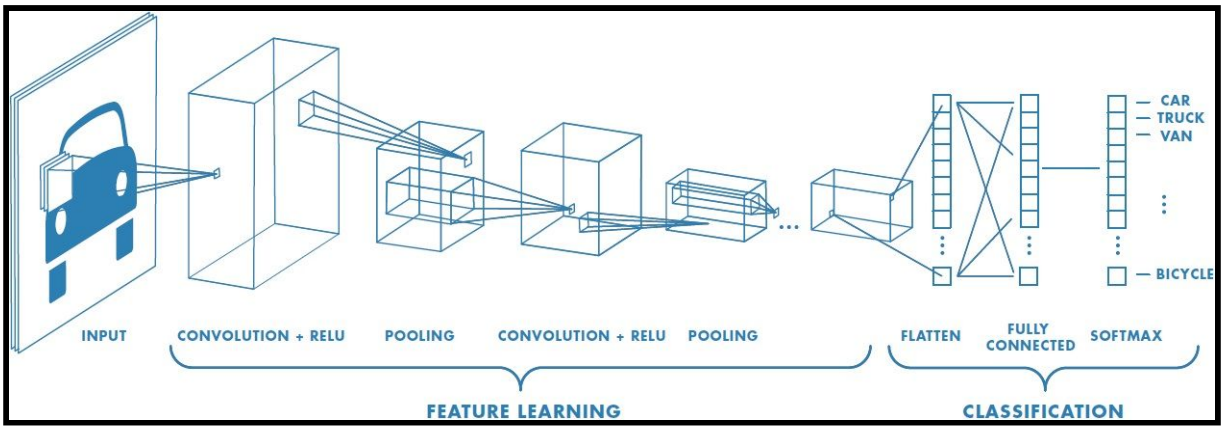



**Fig. 13: Image representation in a CNN[11].**

At each convolutional layer, a linear filter is applied to the image. After each convolution it is common to apply a nonlinear layer immediately afterward. The purpose of this layer is to introduce nonlinearity to a system that basically has just been computing linear operations during the conv. layers. After some ReLU layers, programmers may choose to apply a pooling layer. It is also referred to as a downsampling layer. In this category, there are also several layer options, with maxpooling being the most popular. In the figure above, this is done twice for feature learning. The classification process is to flatten the multidimensional representation (3 -> 1 in this case) and neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular neural networks. The softmax function, is a normalized exponential function (and a generalization of the logistic function) that reduces a K-dimensional vector of real values to a K-dimensional vector of real values in the range [0, 1] that add up to 1. *Other artificial neural network methods include Multilayer perceptrons, Self Organizing Maps (SOM) and Bolzmann machines*.

The disadvantages of neural networks are i) it can be difficult to know how the network calculates each layer and ii) a lot of training data is needed for deep neural networks. However, with Big data, training of deep neural networks have become feasible and the last few years have seen many formidable application of deep neural networks.

## 4.4 Logic-based methods

*Inductive logic programming* (ILP) is a branch of machine learning which uses logic programming as a representation. Examples, background knowledge and hypotheses can be represented in the language of choice. Prolog is a popular choice

[11] https://goo.gl/B3sRKj (Mathworks example of a CNN)

coding ILP-suitable problems. With an encoding of the known background knowledge and a set of examples represented as a logical database of facts, an ILP system will derive a hypothesised logic program which entails all the positive and none of the negative examples. Essentially, logical expressions are constructed to characterise the input classes. The theory of ILP is based on proof theory and model theory for the first order predicate calculus.

A *rule-based system* is a set of "if-then" statements that uses a set of assertions, to which rules on how to act upon those assertions are created. In software development, rule-based systems can be used to create software that will provide an answer to a problem in place of a human expert. We saw an example in the associative rules in Fig. 3.

A *decision tree* is a support tool that uses a tree-like graph structure to represent possible consequences, chance event outcomes, resource costs, and utility of the problem. DTs are one way to display an algorithm that only contains conditional control statements. Decision trees are commonly used in many systems to identify a strategy for reaching a specific goal, they are a popular tool in machine learning as they are simple to implement, and many efficient implementations are available. A random forest is a collection of decision trees, and have proven to give good results. One drawback, is the limited transparency of the solution, it difficult to know how the algorithm, particularly the random forest reaches a solution. Sometimes, lower accuracy is traded off for transparency of the algorithm (it is clear how it worked). A decision true can be constructed using, information theory, a field introduced by Shannon in the 1940's who formalised a measure of randomness, so called Entropy.

## 4.5 Heuristic search

Search is well established, research and deployment within computer science. The idea is to search through a number of different models, or parameters in a model expression, to find something that matches the data and can be used during training of other machine learning models.

Algorithms, many stemming from optimisation, genetic algorithms and reinforcement learning, and simulated annealing are well known. Typically one wants to find a maximum or minimum point, in a set of data without actually know the true lie of the data. In some cases one would like to find more than 1 maximum or minimum, e.g. "the maximum value and the minimum cost" when building a device. So called, multi-parameter optimisation is commonplace in industry. Where one cannot find the minimum and maximum points through calculus, usually we are forced into some form of search algorithm.

The Minimum Description Length (MDL) principle says that given a limited set of observed data the one that permits the *greatest compression of the data* is the best one. MDL is based on any regularity in the data can be used to compress the data, i.e. to describe it using fewer symbols than the number of symbols needed to describe the data literally. The more regularities there are, the more the data can be compressed. MDL methods are particularly well-suited for dealing with model selection, prediction, and estimation where the models can be complex, and hence overfitting the data is a problem. Overfitting is one situation where the solution fits the data well, like a curve going through all the data points, but is a complex function with many terms and higher order exponents. A simpler line might not be as accurate, but is much simpler `y = mx + c`. MDL was introduced by Jorma Rissanen, a well-known Finnish statistician in 1978 [**MDL**].

## 4.6 Data representation

How data is represented is an important facet in any system. If the system is to be designed from the start, a concise, standard, secure representation is usually a good starting point. *Concise* due to processing millions of records, *standardised* so packages such as Python/Ruby/C++ can read and write them, and of course *secure*, which means the data stream can be encrypted when sent over an open network and secured end-to-end if not. Alas, large organisations go with non-standard solutions and formats, which makes using cheap, fast, open and importantly constantly evolving solutions, e.g. from Apache more difficult.

## 4.7 Data validation

In machine learning, model validation is referred to as the process where a trained model is evaluated using a testing data set. The main purpose of using the testing data set is to test the generalization ability of a trained model. Model validation is carried out after model training, and typically 25% of the data is used for training, 25% for validation and 50% for running the experiments.

Undertraining obviously leads to poor results, as insufficient data means the model has not seen the full range of situations that will happen in the real world. Basically this is simply an error in planning and procedure. However, overtraining is a more common problem, and less easily detected. Typically errors include having a too complex model, having too similar training, validation and test samples, fine tuning and evaluating different models with the same data [**BIDAF**].

## 4.8 Free packages for data mining and machine learning

Many packages exist, including ones in R, C++ however we focus on the most popular ones in Python are `numpy` and `scipy`. Additional toolkits such as

`scikit-learn` and `pandas` are worth having installed. For plotting and output `matplotlib` is a popular choice, but more involved visualisations can be made with Processing, Javascript, or `d3`. For the data processing, `Spark` and Flink provide data mining and machine learning libraries as well as for graph processing and stream processing with front ends for `Python, Scala, and R`. Further, ElasticSearch and Kibana are free. Hopsworks is also free as mentioned in the introduction.

# 5 Significance of Big Data frameworks

One of the important goals with BADA has been to investigate the viability of investments in Big data technology.

The data made available for BADA has not been of the scales that would have allowed a direct illustration how big data technology improves on traditional analysis. However, the scales have been sufficient for developing and testing platforms and algorithms. Moreover, we have had access to streaming data and data of various formats.

During the lifetime of BADA, the recent years' development in big data and analytics have surpassed every expectation on the fields and its applications. We have seen the proliferation of Deep Neural Networks, GPU-based computation, Scaling, Elasticity, and Ubiquitous connectivity. The possibility of deploying machine learning algorithms in industrial applications is to a huge extent dependent on the use of big data for training. Convolutional neural networks for object recognition and Reinforcement learning used in Google's Alpha Go are two practical examples of the success of Machine learning. In 2017, several technology companies are deploying mobile AI-chips, which will undoubtedly take the development of the fields further.

While the usefulness of Big data has been shown in many other contexts, BADA has shown that it is feasible for the Swedish Automotive industry to make use of the new technologies. For the use cases in BADA:

- We have implemented scalable algorithms that can be used for solving big data analysis tasks
- We have built data analytics platforms using state-of-the-art freely available software and frameworks
- Some of the investments in technology transfer and building up knowhow in Swedish industry has been made as a part of BADA during seminars and workshops

# 6 Summary

This document has presented the algorithms used in BADA during the work with the Use Cases. We have included insight and background to these algorithms. We have selected and covered a number of further algorithms that we believe should also be covered.
The focus of the work was to investigate and illustrate a selection of algorithms that can be used for big data analytics on automotive data.

Working with the use cases with platforms, data, algorithms with industry has led to four lessons learnt.

1. *Data Readiness issues*, wrangling, working with data takes time. Often, more than 80% of analysis work is spent on preparing the data.

2. *Ground truth*. In some cases we do not really have a ground truth to test against. An example is from the end of queue detection where it is not obviously clear where the real end of queue is, as there might be sections of cars moving along, or some vehicles close, but not in a queue.

3. *Dissemination*. Apart from F2F meetings, email and phone meetings, we have documents in Google docs, code on a Git server, so we decided to create a single site, at least to hold the some of the results, code and pointers, this is BADA.

4. *Liaison*. Start early on the tasks and provide feedback to the partners as soon as possible. Inevitably clarification are needed, and to show the work in progress is very important, so-called “Jenkins” development in the software industry.

# 7 An annotated reading list

This reading list is deliberately short and to the point of the document. We include referenced article with short notes. Some other material of interest was in the footnotes, which are active links to relevant material.

**[TopicModelling]** Agazi Mekonnen and Shamsi Abdullayev, “Topic modelling and clustering for analysis in road traffic accidents”
https://goo.gl/ydfWSs
Department of Applied Mechanics, Chalmers University of Technology, 2017.

[**YouTube1**] Machine Learning (Introduction + Data Mining VS ML)
https://www.youtube.com/watch?v=NP2-M3qRqAU
*Simple 8 minute introduction to the differences between ML and DM.*

[**Perner**] Complementarities and differences between machine learning, data mining and statistics., Petra Perner, 2015, IBM.
http://data-mining-forum.de/complementarities_and_differences_between_machine_learning_and_data.pdf

*Keynote slideset on the differences between ML and DM presented at a conference.*

**[Coursera]** Machine Learning, Andrew Ng**.**
https://www.coursera.org/learn/machine-learning

*The course provides a broad introduction to machine learning, datamining, and statistical pattern recognition. Topics include: (i) Supervised learning (parametric/non-parametric algorithms, support vector machines, kernels, neural networks). (ii) Unsupervised learning (clustering, dimensionality reduction, recommender systems, deep learning). (iii) Best practices in machine learning (bias/variance theory; innovation process in machine learning and AI). Many people at SICS-RISE have taken this course and consider it worthwhile.*

[**Jolliffe**] “Principal Component Analysis”, I.T. Jolliffe, 2004.
http://www.springer.com/gp/book/9780387954424

*A comprehensive, well thought out, clear book on PCA and its many derivatives.*

[**k-means**], Prof. Erik Sudderth, 2012
http://cs.brown.edu/courses/cs195-5/spring2012/lectures/2012-04-10_clustering.pdf

*Nice slideset on clustering algorithms.*

**[Causal]** CausalImpact by Google.
https://google.github.io/CausalImpact/

*The CausalImpact algorithm implements an approach to estimating the causal effect of a designed intervention on a time series. For example, how did ABS brakes affect road safety, The algorithm aims to address this using a structural Bayesian time-series model to estimate how the response metric might have evolved after the intervention if the intervention had not occurred. The CausalImpact package assumes that the outcome time series can be explained in terms of a set of control time series that were themselves not affected by the intervention. Furthermore, the relation between treated series and control series is assumed to be stable during the post-intervention period.*

[**Statistical learning**] Foundations of Machine Learning, Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar, 2012.
https://mitpress.mit.edu/books/foundations-machine-learning

*A graduate-level textbook focussing on fundamental concepts and methods in machine learning. Several important modern algorithms are underpinned by the theory (and some applications). Rather theoretical, but with good appendices this is a well written book for those interested in diving a little deeper into statistics and mathematics. Essentially functional analysis, which means finding mappings between inputs and outputs, in different 'spaces' than what we used to, (which is a Euclidean space). By allowing infinite series and complex numbers in the 'space' one can find mappings and help learn the connections between inputs and outputs.*

[**Kumar2016**] Kumar, Sachin, Toshniwal, Durga, "A data mining approach to characterize road accident locations", Journal of Modern Transportation, 2016.

*Introduces data mining and its usefulness in analysing road accidents. The focus on identifying factors that affect the severity of an accident, and look propensity at certain specific locations. They first apply k-means algorithm to group the accident locations into 3 categories, i) high ii) moderate and low frequency accident locations. k-means algorithm takes accident frequency count as a parameter to cluster the locations. Then they used association rule mining to characterize these locations.*

[**RoadModels**] "Statistical Models for Road Traffic Forecasting", 2015, Mediamobile & Insitut Mathématique de Toulouse.
https://www.math.univ-toulouse.fr/~agarivie/bigdata/slides/goudal_bigdata_Toulouse.pdf

*Insightful slide set on the problems (and some) solutions on data mining and to a less extent machine learning in a road traffic setting. Mediamobile – a subsidiary of the TDF Group – is*

*one of the leading providers of real-time traffic and mobility information services in Europe. Clustering of jams, as well as statistical modelling of vehicle dynamics is covered. Mediamobile joined the Chalmers-hosted SAFER consortium, https://www.saferresearch.com/about.*

[**Top10**] "Top 10 algorithms in data mining" Xindong Wu et al.
http://www.cs.uvm.edu/~icdm/algorithms/10Algorithms-08.pdf

*Although a little old as of 2017, this paper presents the top 10 data mining algorithms identified by the IEEE International Conference on Data Mining (ICDM) in December 2006: C4.5, k-Means, SVM, Apriori, EM, PageRank, AdaBoost, kNN, Naive Bayes, and CART. These top 10 algorithms are among the most influential data mining algorithms in the research community. With each algorithm, they provide a description of the algorithm, discuss the impact of the algorithm, and review current and further research on the algorithm.*

[**Hackers**] "Machine Learning for Hackers", Oreilly, D. Conway, J, White, 2012.
http://shop.oreilly.com/product/0636920018483.do

*For the programmer interested in crunching data, this book gets hands on practitioners started with machine learning. The focus is on building toolkits of algorithms in R to train themselves to automate tasks. Hands-on case studies are the approach to learn ML. Chapters focus on a specific problem in machine learning, such as classification, prediction, optimization, and recommendation systems.*

1. *Develop a naïve Bayesian classifier to determine if an email is spam*
2. *Use linear regression to predict #pageviews for the top 1000 websites*
3. *Learn optimization techniques by attempting to break a simple letter cipher*
4. *Compare and contrast U.S. Senators statistically, based on their voting*
5. *Build a "whom to follow" recommendation system from Twitter data*

[**Bishop**] Pattern Recognition and Machine Learning, Christopher Bishop, 2006.
https://www.microsoft.com/en-us/research/people/cmbishop/

*For beginners who need to understand Bayesian perspective on Machine Learning, this book is a decent choice. The author makes a good attempt to explain complicated theories in simplified manner by giving examples/applications. The best part of the book are chapters on graphical models (chap. 8), mixture model EM (chap. 9) and approximate inference (chap. 10). Bayesian approaches feature prominently. Statistical learning and non-Bayesian perspective on machine learning are not really covered. A complement to Tom Mitchell's Machine Learning book.*

[**Murphy**] "Machine Learning, A Probabilistic Perspective", Kevin P. Murphy
https://mitpress.mit.edu/books/machine-learning-0

*This substantial book is a deep and detailed introduction to the field of machine learning, using probabilistic methods. It is aimed at a graduate-level readership and assumes a mathematical background that includes calculus, statistics and linear algebra.*

[**TOPIC**] "Topic Modeling and Clustering for Analysis of Road Traffic Accidents", Agazi Mekonnen, Shamsi Abdullayev, Master's thesis in Computer Science: Algorithms Languages and Logic. http://publications.lib.chalmers.se/records/fulltext/250497/250497.pdf

*They examined different approaches on how to cluster, summarise and search accident descriptions in Swedish Traffic Accident Data Acquisition (STRADA) dataset. One of the central questions in this project was that how to retrieve similar documents if a query does not have any common words with relevant documents. Another question is how to increase similarity between documents which describe the same or similar scenarios in different words. They designed a new pre-processing technique using keyword extraction and word embeddings to address these issues. Theoretical and empirical results show the pre-processing technique employed improved the results of the examined topic modeling, clustering and document ranking methods.*

[**CNNintro**] A Friendly Introduction to Convolutional Neural Networks, Ifu Aniemeka, 2017. https://hashrocket.com/blog/posts/a-friendly-introduction-to-convolutional-neural-networks
*An 8 page presentation of CNNs with examples of the layer construction.*

[**NNintro**] Quick introduction to neural networks, Ujjwal Karn, 2017. https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks

*Nice intro to data science. With plenty of links to other material. Suitable for beginners to machine learning, data mining and data science. Link to data school http://www.dataschool.io from this site.*

[**Ian2017**] "Dimensionality reduction in (large) measurement datasets" , Submitted to Sigcomm Big Data Conference. *2017.* *http://ianmarsh.org/wp-content/uploads/2017/01/bigdama_ianmarsh_25.pdf*

*6 page conference paper written by Ian Marsh on using dimensionality reduction in large amounts of telecommunications data using 2 techniques, covers PCA as described in this report.*

[**MDL**] "A tutorial introduction to the minimum description length principle", Peter Grunwald, 2004. *https://arxiv.org/abs/math/0406077*

*This tutorial paper provides a tutorial on MDL. It takes on how does one decide among competing explanations of data given limited observations. It is a problem of model selection. The MDL principle is a method for inductive inference that provides a generic solution to the model selection problem. MDL is based on any regularity in the data can be used to compress the data, i.e. to describe it using fewer symbols than the number of*

*symbols needed to describe the data literally. The more regularities there are, the more the data can be compressed. Equating 'learning' with 'finding regularity', we can therefore say that the more we are able to compress the data, the more we have learned about the data. Formalizing this idea leads to a general theory of inductive inference with several attractive properties, such as Occam's Razor, no overfitting,, no need for 'underlying truth' and predictive interpretation.*