Lappeenranta University of Technology
School of Business and Management

A365A0320 Computational Data Analytics in Business Management

Samuli Korpelainen

Use of Predictive Analytics for Marketing in Bank

25th May, 2018

Ajesh Sharma 0525504
Kimmo Flykt 0438358
Mikael Sommarberg 0420191
Sami Kabir 0528747

Table of Contents

1. Introduction

Human brain is a predictive engine due to its underlying construction where the bundles of cells that support perception and action, constantly attempts to match incoming sensory inputs with top-down expectations or predictions, which is achieved using a hierarchical generative model that aims to minimize prediction error (Clark, A. 2013). However, when it comes to analysis of very large set of data, human brain may not be very effective as well as fast in comparison to computational technologies, which in many cases, may be inspired by human intelligence principles, but can be more accurate than human, if problem statements are modeled correctly and if parameters are defined unequivocally (Rainie and Anderson, 2018).

Predictive Analytics is a multi-step process with multitude of statistical models and techniques under the hood which analyzes recent as well as historical data to make predictions about future. The goal in predictive analysis is to develop such models which can predict future events and behaviors of variables or parameters (Kumar and Garg, 2017). In most of the predictive models a score is computed, where a higher score indicates a higher likelihood of a given behaviour or event occurring or vice-versa. In today’s day and age, predictive analytics methods are widely being used in business, science and other problems areas. In business problems, predictive analytics model exploits patterns of historical and transactional data to derive valuable insight that can add value to the business (Ricardo Buettner, 2016).

Predictive Analysis process involves pulling information from different data repositories, which are then cleaned and organized in a relevant and meaningful way, which may then be combined with external data and mined to develop a predictive model, which results into a predictive model that can help a business gain new perspective or answer some very critical business questions (Charles Nyce, 2007). Figure 1 below shows the steps of predictive Analytics Process.

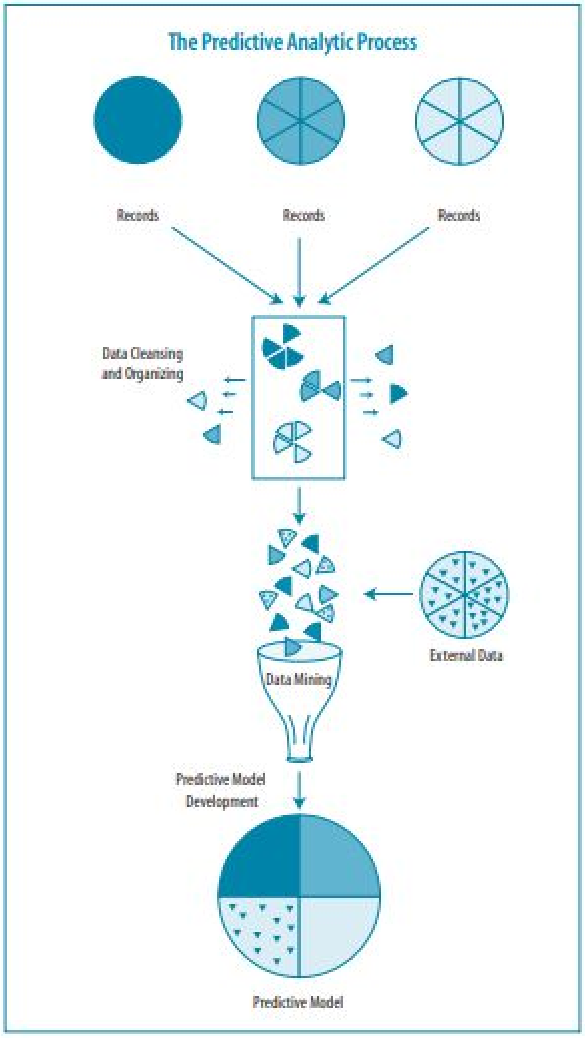


Figure 1. Predictive Analytics process (Charles Nyce,2007)

In the age of Big-Data, it is nearly impossible for a business to scale without data-driven techniques and solutions. The Portuguese bank, in our case study scenario, is considering how to optimize their marketing campaign in future by analyzing customer data. Marketing managers can make data-driven decision that may suggest about effective client selection, which would increase the success rate. Moreover, direct marketing could be effective, but it has some drawbacks, such as causing negative attitude towards banks due to the intrusion of privacy. A more targeted campaign helps business in creating a pool of more satisfied customer.

This 'Use of Predictive Analytics for Marketing in Bank' project has been carried out as part of a course work requirement at Lappeenranta University of Technology (LUT)'s "Computational Data Analytics in Business Management" course. In this project, we will be studying and comparing different machine learning models by using them to classify and predict how a targeted potential customer behaves in a direct bank marketing campaign. The prediction model predicts whether a client will subscribe to a term deposit product offering by the bank or not. The data used in this project, is a publicly available data set for research purposes.

The main goal for this project is to compare prediction accuracy of different machine learning models with the given historical bank marketing campaign data. If classifier has a very high accuracy, it can help the manager to filter clients and use available resources more efficiently to achieve the campaign goal. Proper strategy would reduce cost and improve long term relations with the clients. We have chosen to compare five different machine learning algorithms: K-Nearest Neighbours, Logistic Regression, Linear Discriminant Analysis, Decision Tree and Gaussian Naive Bayes.

2. Background and motivation

The aim of the marketing campaign was to get customers to subscribe to one of the bank’s term deposit product. Whether a customer chose to subscribed to the offered product or not is represented by variable ‘y’ in the data set. The bank in question is considering how to optimize this campaign in future by analyzing this dataset.

There are two main approaches for enterprises to promote products and/or services: through mass campaigns, targeting general indiscriminate public or directed marketing, targeting a specific set of contacts (Ling and Li 1998). Nowadays, in a global competitive world, positive responses to mass campaigns are typically very low, less than 1%, according to the same study. Alternatively, directed marketing focus on those targets that assumably will be keener to avail that specific product/service, making this kind of campaigns more attractive due to its efficiency (Ou et al. 2003). Nevertheless, directed marketing has some drawbacks, for instance it may trigger a negative attitude towards banks due to the intrusion of privacy (Page and Luding 2003).

It should be stressed that due to intensive competition and new financial avenues such as crypto-currencies available to the general public, there is a huge pressures on banks world over to increase their financial assets. To meet this challenge, one adopted strategy is to offer attractive long-term deposit products with good interest rates, in particular by using directed marketing campaigns to make the exercise effective. Moreover, the same drivers that are stressing the bank’s financial health, are also creating a case for reduction in operational costs and time. Thus, there is a need for an improvement in efficiency: lesser contacts should be done, but an approximately number of successes (clients subscribing the deposit) should be kept.

3. Data

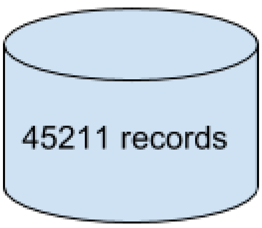
The dataset used in this project is a publicly available dataset that was used by S. Moro, R. Laureano and P. Cortez in their project titled ‘Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology’ (Moro et al., 2011). This dataset contains the information related to bank’s clients and bank’s previous marketing campaign. The table below describes the different fields in this dataset, their data types and expected values.

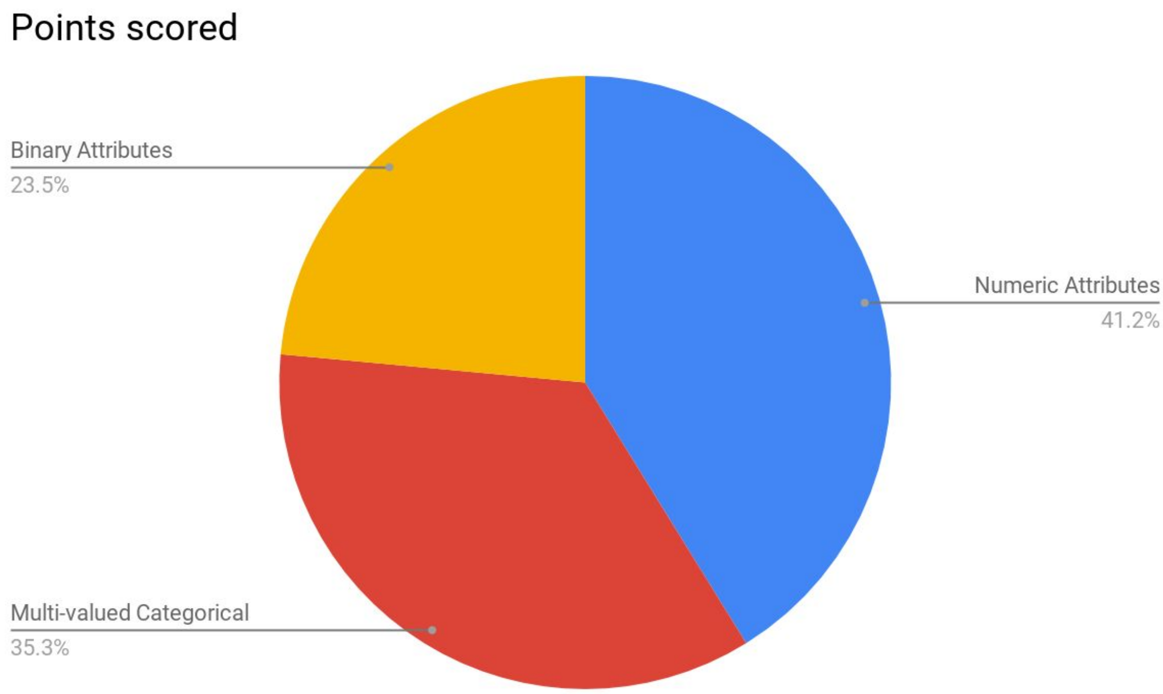
Sl. no.	**Variable Name**	**Data Type**	**Expected Values**	**Significance**
01	Age	Numeric	18, 30, 60, 90 ...	Describes the age of the customer.
02	Job	Character	‘Student’, ‘unemployed’ ‘manager’ etc.	This is a categorical information describing the employment scenario of a customer.
03	Marital	Character	‘Married’, ‘Single’ etc.	This describes the marital status of the customer.
04	Education	Character	‘Unknown’, ‘secondary’, etc	This describes the education qualification of the customer.
05	Default	Binary	‘Yes’ or ‘No’	This describes if the customer has credit in default or not.
06	Balance	Numeric	10000, 1000, 999 ...	This describes the average yearly balance in euros of a customer.
07	Housing	Binary	‘Yes’ or ‘No’	This describes if the customer has housing loan or not.
08	Loan	Binary	‘Yes’ or ‘No’	This describes if the customer has any personal loan or not.

09	Contact	Character	‘Unknown’, ‘Cellular’ etc.	This describes the categorical information about the contact communication type of a customer.
10	Day	Numeric	01, 15, 20, 30 ...	This describes the day of the month when the last contact was made with the customer.
11	Month	Character	‘Jan’, ‘Feb’, ‘Mar’, etc.	This describes the month of the year when the last contact was made with the customer.
12	Duration	Numeric	10, 19, 300, ...	This describes the last contact duration in seconds.
13	Campaign	Numeric	1, 3, 5 ….	This describes the number of times contacts were made for this campaign and for a customer.
14	Pdays	Numeric	-1, 10, 15 ...	This describes the number of days that passed by after the client was last contacted from a previous campaign.
15	Previous	Numeric	1, 10, 14 ...	This describes the number of contacts performed before this campaign and for this client.
16	Poutcome	Character	‘Unknown’, ‘Success’, Failure etc.	This describes the outcome of the previous marketing campaign.
17	Y	Binary	‘Yes’ or ‘No’	This describes if a client has subscribed a term deposit or not

Here our main target variable is ‘Y’ which describes if a client has subscribed a term deposit or not.

Following initial impression that can be created using the dataset.





Other observations that can be made about the data are:

- No missing values: Preprocessing should be easier.
- 81.74% of the time outcome of previous marketing campaign is unknown.
- Data is very imbalanced, only 11.69% yes in outcome ‘y’.

We use python programming language for this analysis with Atom editor.

In this benchmark, the historical data of last call was left out, and employment status was simplified (3 categories: unemployed, student or neither == employed). The main reason for this was to gain some difference when comparing to the study created by the original data providers S. Moro et al.

4. Methodology

Predictive modeling, as we now know, is a process that uses data mining and probability to forecast outcomes, where each model is made up of a number of predictors, which are variables that are likely to influence results. After data has been collected for relevant predictors, a statistical model is required to be formulated. The model may employ a simple linear equation, or it may be a complex neural network, mapped out by sophisticated software. As additional data becomes available, the statistical analysis model is validated or revised. There are many techniques that can be used for a given data set, however, we have chosen the following five techniques on the available data set and have compared their results.

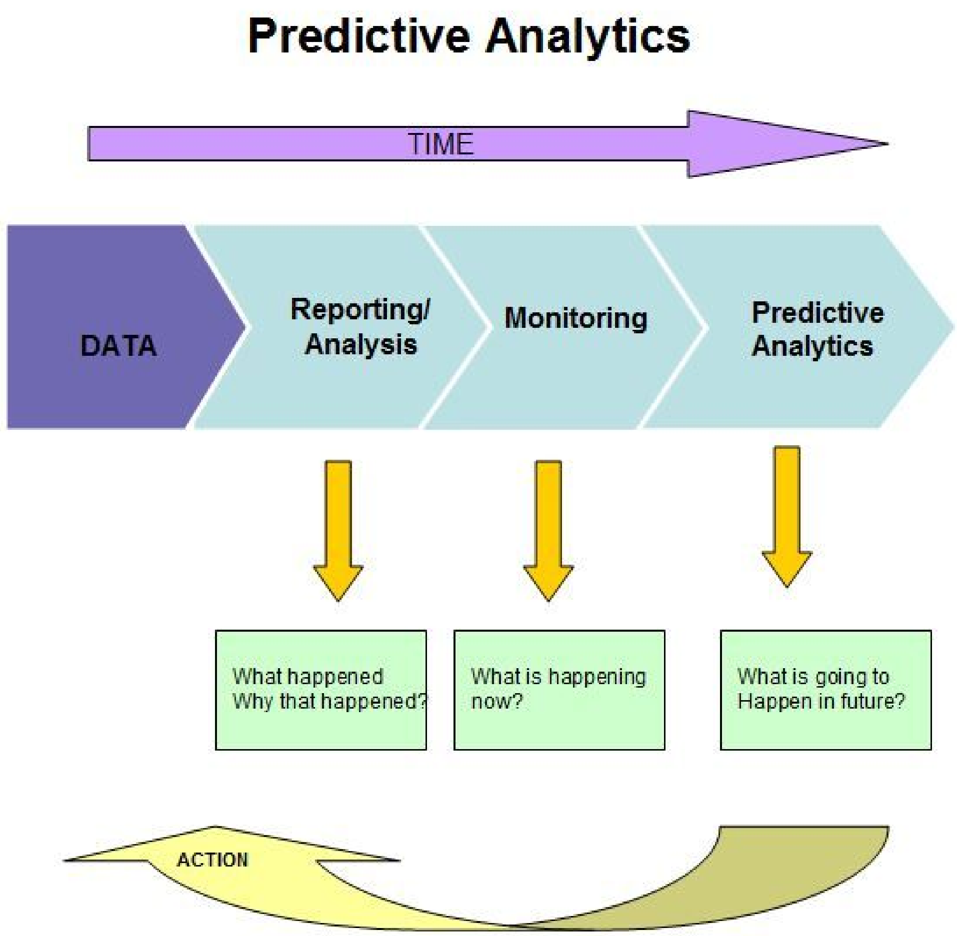


Figure 2. Predictive Analytics as an Iterative methodology

4.1. K-Nearest Neighbors

The purpose of the k-nearest neighbor prediction function is to predict a target variable from a predictor vector. Commonly, the target is a categorical variable, a label identifying the group from which the observation was drawn. The analyst has no knowledge of the membership label but does have the information coded in the attributes of the predictor vector. The predictor vector and the k-nearest neighbor prediction function generate a prediction of membership. In addition to qualitative attributes, the k-nearest neighbor prediction function may be used to predict quantitative target variables.(Steele, B. 2016)

The basic concept of k-NN is to associate a record or sample into a class based on how close its "predictors" are to the predictors of that class. For example, if we consider the task of classifying owners of a charter plane. The predictors for ownership might be income level , home ownership and age of children (if any). For example, high income business man with no kids or adult children may be owners, whereas medium income, homeowners with school age kids may not be (they may choose to fly in low cost airline). The question that a k-NN exercise might be called to answer is, given the predictor information for a new prospect, could they be considered as an owner or not?

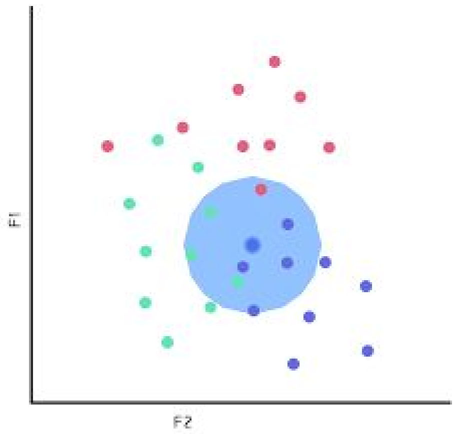


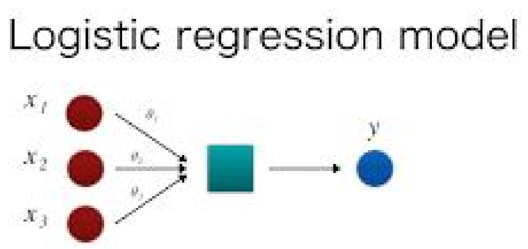
Figure 3. K-NN methodology

The simplest case is when k=1. In other words, if we find an existing record (or data point) in the available data which is the "closest" to our test data point. If we find one, then we put our test data point in the same class as the nearest record so found. K-NN works well if there are only few classes. For data where there may be multiple classes, we run the risk of "over-fitting" the model and the chances of mis-classification are higher. When k > 1, the algorithm determines a cluster of records to which our test sample is closest to. It then binds the test sample in the same class as the majority of records in that cluster. Figure 3 represents K-NN method graphically. Using a very high "k" value results in over-smoothing of the results and is no different than simply binning the test sample in a class with the most number of records! Computing time also increases as k increases, however the advantage is that higher values of k provide smoothing that reduces vulnerability due to data quality issues in the Training Set. Typically, k is in units of tens of units, rather than in hundreds or thousands.

4.2. Logistic Regression

Many social phenomena are discrete or qualitative rather than continuous or quantitative in nature. For example this kind of phenomenon can be marriage. From one viewpoint we can draw line where person is either married or not. Same logic can be used to determinate if person is dead or not. (Pampel, F. C., 2000)

Logistic regression is so called supervised learning model. In supervised learning we are given a set of input-output pairs that we can call training set. Training set is used to learn a function that can be applied to previously unseen data. For many applications, we would like to estimate the probability that new instance belongs to the class of interest denoted as (0,1). The model produces numeric value and is one differentiating characteristic between classification and regression. Logistic regression is a class probability estimation model and not a regression model.(Morreale A, 2018)



In logistic regression the response variable is binary. the response can be either true (success) or false (failure), usually coded as 1 and 0. logistic regression models are quite useful for classifying new cases into one of two outcome categories (“success” or “failure”). the estimated logistic model, applied to new cases of a test (evaluation) data set, provides predictions of success probabilities. with a certain cutoff on the predicted success probabilities, the logistic regression provides a rule for classifying new cases. this chapter looks at examples of death penalty data, data of delayed airplanes, loan acceptance data, and german credit data. it uses the statistical software r for the estimation, but also uses minitab, a popular spreadsheet-based software program, in the first illustrative example. (Ledolter, J. 2013)

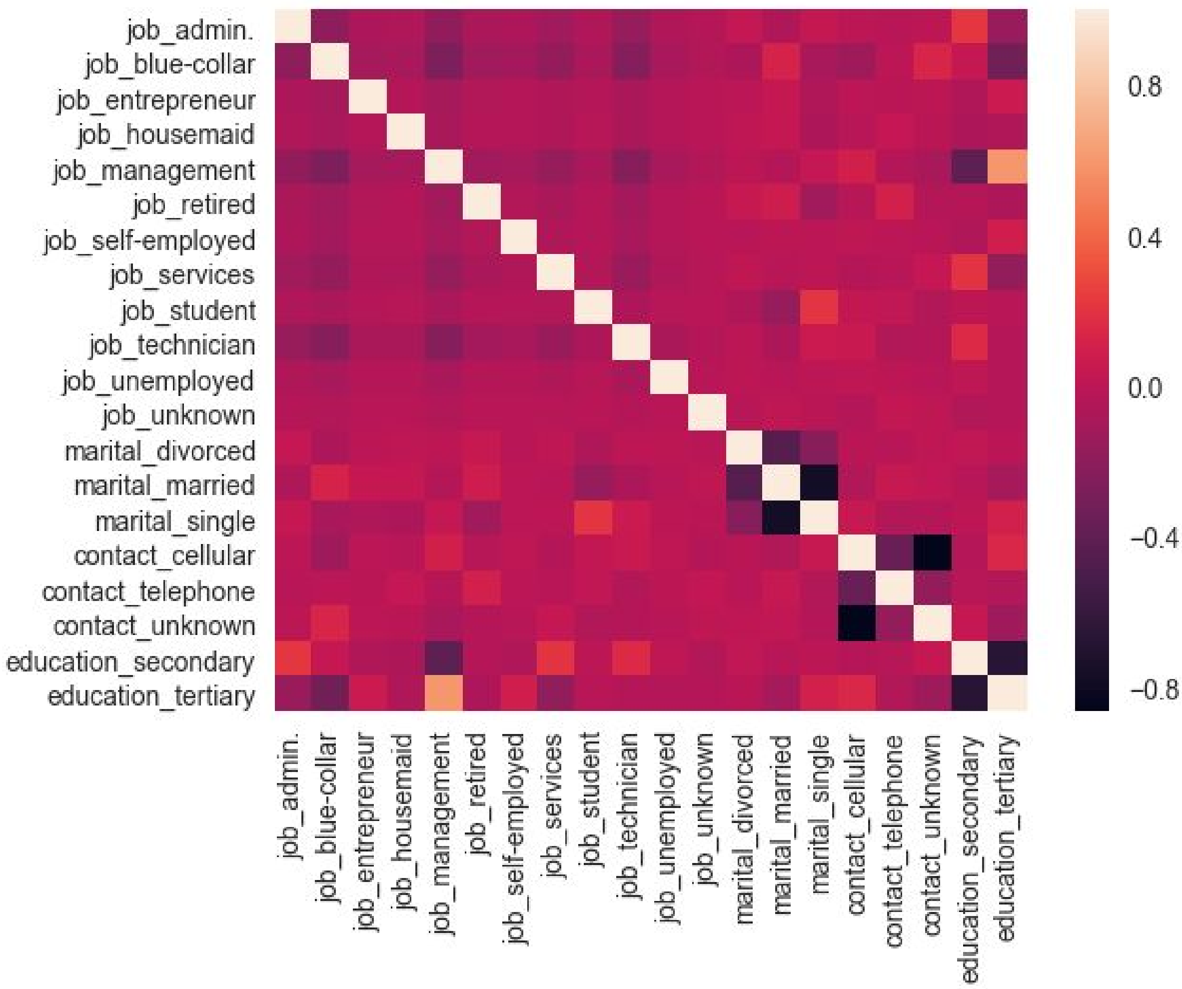


Figure 4, heatmap for logistic regression. Shows how ROC AUC varied between chosen parameters(education, marital, contact and job).

4.3. Linear Discriminant Analysis

Linear Discriminant Analysis was developed to address the shortcomings of Logistic Regression and is the go-to linear method for multi-class classification problems. Logistic regression is a simple and powerful linear classification algorithm. It also has limitations that suggest at the need for alternate linear classification algorithms.

- <u>Two-Class Problems</u>: Logistic regression is intended for two-class or binary classification problems. It can be extended for multi-class classification, but is rarely used for this purpose.
- <u>Unstable With Well Separated Classes</u>: Logistic regression can become unstable when the classes are well separated.

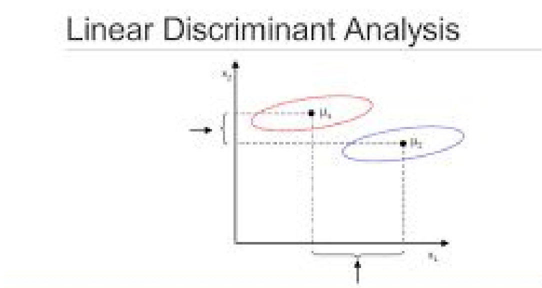
● Unstable With Few Examples: Logistic regression can become unstable when there are few examples from which to estimate the parameters.

The representation of LDA is straight forward as it consists of statistical properties of data, calculated for each class. For a single input variable (x) this is the mean and the variance of the variable for each class. For multiple variables, this is the same properties calculated over the multivariate Gaussian, namely the means and the covariance matrix. These statistical properties are estimated from data and plug into the LDA equation to make predictions.

LDA makes some simplifying assumptions about data:

- That data is Gaussian which means that each variable is shaped like a bell curve when plotted.
- That each attribute has the same variance, which means that values of each variable vary around the mean by the same amount on average.

With these assumptions, the LDA model estimates the mean and variance from data for each class. It is easy to think about this in the univariate (single input variable) case with two classes.



The mean (mu) value of each input (x) for each class (k) can be estimated in the normal way by dividing the sum of values by the total number of values.

muk = 1/nk * sum(x)

Where muk is the mean value of x for the class k, nk is the number of instances with class k. The variance is calculated across all classes as the average squared difference of each value from the mean.

sigma^2 = 1 / (n-K) * sum((x – mu)^2)

Where sigma^2 is the variance across all inputs (x), n is the number of instances, K is the number of classes and mu is the mean for input x

LDA makes predictions by estimating the probability that a new set of inputs belongs to each class. The class that gets the highest probability is the output class and a prediction is made.

The model uses Bayes Theorem to estimate the probabilities. Briefly Bayes' Theorem can be used to estimate the probability of the output class (k) given the input (x) using the probability of each class and the probability of the data belonging to each class:

$$P(Y=x|X=x) = (PIk * fk(x)) / sum(PIl * fl(x))$$

Where PIk refers to the base probability of each class (k) observed in your training data (e.g. 0.5 for a 50-50 split in a two class problem). In Bayes' Theorem this is called the prior probability.

$$PIk = nk/n$$

The f(x) above is the estimated probability of x belonging to the class. A Gaussian distribution function is used for f(x). Plugging the Gaussian into the above equation and simplifying we end up with the equation below. This is called a discriminant function and the class is calculated as having the largest value will be the output classification (y):

$$Dk(x) = x * (muk/siga^2) – (muk^2/(2*sigma^2)) + \ln(PIk)$$

Dk(x) is the discriminant function for class k given input x, the muk, sigma^2 and PIk are all estimated from your data.

4.4. Decision Tree

Decision trees in scikit-learn is a non-parametric supervised machine learning method used for classification and regression analysis. The goal in creating decision tree algorithm is to create a model that predicts the value of a target by creating simple decision rules inferred from the data. The main benefits of using decision tree algorithm are the simplicity to understand and interpret and it requires rather little data preparation. (Pedregosa et al., 2011)

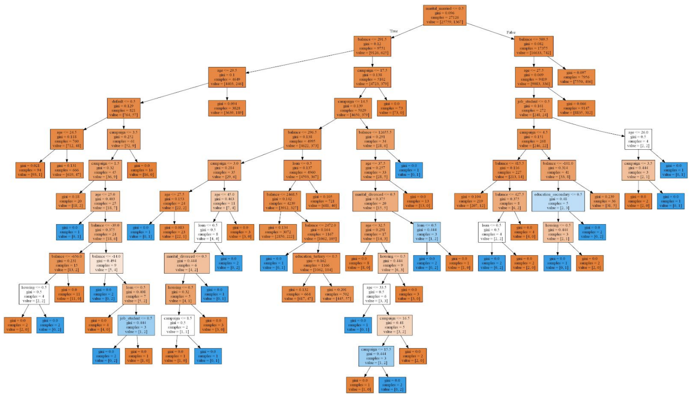
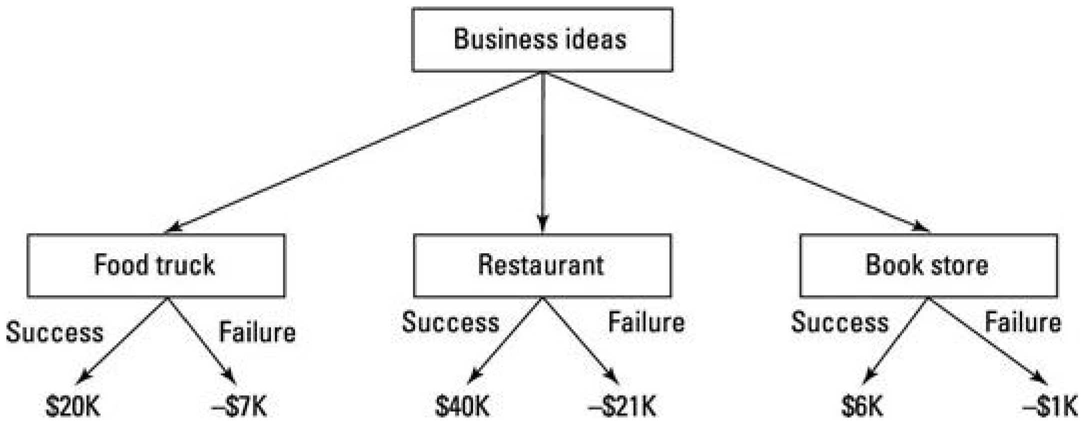


Figure 5, visualization of a limited width decision tree

Scikit-learn as a tool supports two modes in decision tree classification: gini impurity and entropy (=”information gain”)(Pedregosa et al., 2011). The software creator recommended setting, gini impurity, was used in our benchmark.

A decision tree is an approach to predictive analysis that can help business owners make decisions. Suppose, for example, that you need to decide whether to invest a certain amount of money in one of three business projects: a food-truck business, a restaurant, or a bookstore. A decision tree to decide on a business venture begins with calculating the expected value for each alternative — a numbered rank that helps select the best one.



The expected value is calculated in such a way that includes all possible outcomes for a decision. Here the expected value reflects the average gain from investing in a business.Therefore the expected value becomes one of the criteria you figure into business decision-making.

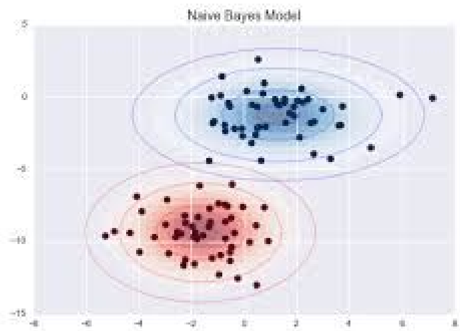
Decision trees can also be used to visualize classification rules. A decision algorithm generates a decision tree that represents classification rules, where the decision tree will be, essentially, a flow chart: Each node of the decision tree represents an attribute identified in the data matrix. The leaves of the tree are the predicted decisions.

4.5. Gaussian Naive Bayes

Naive Bayes classifier is a straightforward and powerful algorithm for the classification task. Even if we are working on a data set with millions of records with some attributes, it is suggested to try Naive Bayes approach. Naive Bayes classifier gives great results when we use it for textual data analysis. Such as Natural Language Processing.

Naive Bayes methods are supervised learning algorithms based on the Bayes' theorem. The naive prefix here means, that there is an optimistic assumption of independence between every pair of features. (Pedregosa et al., 2011)

Gaussian Naive Bayes is a classification algorithm in scikit-learn. In the documentation it is said to be extremely fast when comparing to more sophisticated methods, but on the flipside it is a bad estimator (and a decent classifier). (Pedregosa et al., 2011)
The simplicity of Bayes' theorem and its applied algorithms is one of the reasons Bayes based mathematical models work well. Simplicity in this context means low variance in its probability estimates and the results can be biased. This may not matter in classification use as long as the order is preserved. (Hand, D.J., Yu, K, 2001)



Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. This is also known as Maximum A Posteriori (MAP).

The MAP for a hypothesis is:

MAP(H)

= max(P(H|E))

= max((P(E|H)*P(H))/P(E))

= max(P(E|H)*P(H))

P(E) is evidence probability, and it is used to normalize the result.

Naive Bayes classifier assumes that all the features are unrelated to each other. Presence or absence of a feature does not influence the presence or absence of any other feature.

5. Implementation

The implementation of the machine learning models was in python language. We have used Atom Editor for this purpose.

The imported libraries/packages are as follow.

```python
# Load libraries
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
import pandas as pd #Import pandas library
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

read_data() function was used to interpret the bank-full.csv file. In the feature_to_dummy() function, we have called get_dummies() to convert categorical variables of this .csv file into indicator variables.

```python
# Load dataset
def read_data(): #Read data function
    DF = pd.read_csv('bank-full.csv', delimiter=";", \
    true_values=["success","yes"], false_values=["failure","no"])
    DF = feature_to_dummy(DF, "education", True)
    DF = feature_to_dummy(DF, "marital", True)
    DF = feature_to_dummy(DF, "job", True)
    DF = feature_to_dummy(DF, "contact", True)
    return DF

def feature_to_dummy(DF, column, drop=False):
    tmp = pd.get_dummies(DF[column], prefix=column, prefix_sep='_')
    DF = pd.concat([DF, tmp], axis=1, join_axes=[DF.index])
    if drop:
        del DF[column]
    return DF
```

In the implement_machine_learning() function, we have divided the whole dataset into two parts: a) training dataset (80%), b) testing dataset (20%) using train_test_split() function.

```
def implement_machine_learning(DF): #defined function implement_machine_learning from data frame
    features = ["age", 'job_student', 'job_unemployed',
            "marital_divorced", "marital_married", "marital_single","education_secondary",
               "education_secondary", "education_tertiary", "default","balance","housing", "loan","campaign"]
    validation_size = 0.20
    seed = 7
    scoring = 'accuracy'
    Y = DF['y'] #Selected the y from the column Result
    X = DF[features] #Selected the x to be the features
    X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y, test_size=validation_size,
                                                    random_state=seed)
```

cross_val_score() function evaluates all the five machine learning models.

```
    # Test options and evaluation metric
    models = []
    models.append(('LR', LogisticRegression()))
    models.append(('LDA', LinearDiscriminantAnalysis()))
    models.append(('KNN', KNeighborsClassifier()))
    models.append(('CART', DecisionTreeClassifier()))
    models.append(('NB', GaussianNB()))
    # evaluate each model in turn

    for name, model in models:
        kfold = KFold(n_splits=10, random_state=seed)
        cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring=scoring)
        results.append(cv_results)
        names.append(name)
        msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
        print(msg)
```

The prediction accuracy of all the five models have been shown graphically using the compare_algorithms() function.

```
def compare_algorithms():
    # Compare Algorithms
    fig = plt.figure()
    fig.suptitle('Algorithm Comparison')
    ax = fig.add_subplot(111)
    plt.boxplot(results)
    ax.set_xticklabels(names)
    plt.show()
```

Finally, here is the main function of our python file.

```
def main(): #main program function
    DF = read_data() #call read data and save data frame to variable
    implement_machine_learning(DF)
    compare_algorithms()

main() #Call the main function
```

6. Results

The undermentioned table demonstrates the prediction accuracy of the five machine learning models we have used in our project using python language.

Machine Learning Model	Prediction Accuracy
Logistic Regression (LR)	88.02%
Linear Discriminant Analysis (LDA)	88.11%
K-Nearest Neighbour (KNN)	86.90%
Decision Tree Classifier (CART)	81.88%
Gaussian gaussian Naive Bayes (NB)	84.85%

From the table, it can be stated that Linear Discriminant Analysis (LDA) has highest level of prediction accuracy with respect to bank marketing data.

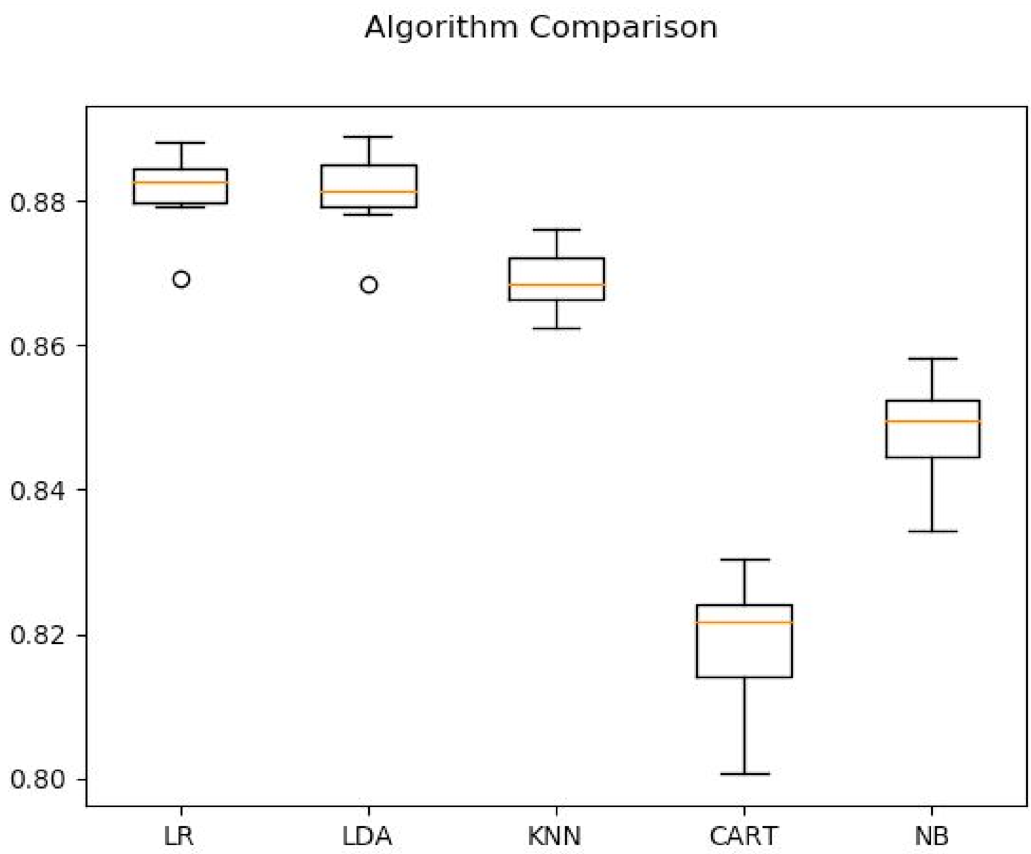


Figure 6: Prediction accuracy comparison of Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbour (KNN), Decision Tree Classifier (CART) and Gaussian gaussian Naive Bayes (NB) models.

7. Conclusions

The results explained in the previous chapter concludes the study. We managed to get results from the given data set and compare algorithms, which was our primary goal for this project. While programming, interpreting the results and studying the results, whole research group got a hands-on experience with modern tools for computational data analytics with a real business case.

The next step for this project would be the optimization of each algorithm. In this project, default settings of each algorithm was used. Adjusting settings to get better results would require a more in-depth study of each algorithm to know how the performance could be improved.

8. References

[01] Clark, A. (2013) “Whatever next? Predictive brains, situated agents, and the future of cognitive science,” Behavioral and Brain Sciences. Cambridge University Press, 36(3), pp. 181–204. doi: 10.1017/S0140525X12000477.

[02] Rainie, L. and Anderson, J. (2018). Theme 3: Humanity and human judgment are lost when data and predictive modeling become paramount. [online] Pew Research Center: Internet, Science & Tech. Available at: http://www.pewinternet.org/2017/02/08/theme-3-humanity-and-human-judgment-are-lost-when-data-and-predictive-modeling-become-paramount/ [Accessed 24 May 2018].

[03] V. Kumar and M. L. Garg, "Deep learning in predictive analytics: A survey," 2017 International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT), Dehradun, 2017, pp. 1-6. doi: 10.1109/ICETCCT.2017.8280331

[04] Ricardo Buettner, "Predicting user behavior in electronic markets based on personality-mining in large online social networks: A personality-based product recommender framework" in Electronic Markets: The International Journal on Networked Business, Springer, pp. 1-19, 2016.

[05] Charles Nyce, "Predictive Analytics White Paper (PDF)" in American Institute for Chartered Property Casualty, Underwriters/Insurance Institute of America, 2007.

[06] Ling, X. and Li, C., 1998. “Data Mining for Direct Marketing: Problems and Solutions”. In *Proceedings of the 4th KDD conference*, AAAI Press, 73–79.

[07] Ou, C., Liu, C., Huang, J. and Zhong, N. 2003. “On Data Mining for Direct Marketing”. In *Proceedings of the 9th RSFDGrC conference*, 2639, 491–498.

[08] Page, C. and Luding, Y., 2003. “Bank manager’s direct marketing dilemmas – customer’s attitudes and purchase intention”. *International Journal of Bank Marketing* 21, No.3, 147–163.

[09] S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimarães, Portugal, October, 2011. EUROSIS.

[10] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[11] Hand, D. J.; Yu, K. (2001). "Idiot's Bayes — not so stupid after all?". International Statistical Review. 69 (3): 385–399.

[12] Pampel, F. C. (2000). *Quantitative Applications in the Social Sciences: Logistic regression* Thousand Oaks, CA: SAGE Publications Ltd doi: 10.4135/9781412984805

[13] Morreale Azzurra. (2018). Analytics for Business (Principles of modelling: linear regression and logistic regression)

[14] Ledolter, J. (2013). *Logistic Regression*.

[15] Steele, B. (2016). *Algorithms for Data Science*. Cham: Springer International Publishing.

9. Appendixes

Appendix 1. The used source code

```
# Load libraries
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
import pandas as pd #Import pandas library
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

results = []
names = []

# Load dataset
def read_data(): #Read data function
    DF = pd.read_csv('bank-full.csv', delimiter=";", \
      true_values=["success","yes"], false_values=["failure","no"]) #create data frame from iris.data.txt with pandas.read_csv
    DF = feature_to_dummy(DF, "education", True)
    DF = feature_to_dummy(DF, "marital", True)
    DF = feature_to_dummy(DF, "job", True)
    DF = feature_to_dummy(DF, "contact", True)
    return DF
```

```
def feature_to_dummy(DF, column, drop=False):
    tmp = pd.get_dummies(DF[column], prefix=column, prefix_sep='_')
    DF = pd.concat([DF, tmp], axis=1, join_axes=[DF.index])
    if drop:
        del DF[column]
    return DF

def implement_machine_learning(DF): #defined function implement_machine_learning from data frame
    features = ["age", 'job_student', 'job_unemployed',
            "marital_divorced", "marital_married", "marital_single","education_secondary",
                                  "education_secondary", "education_tertiary", "default","balance","housing", "loan","campaign"]
    validation_size = 0.20
    seed = 7
    scoring = 'accuracy'
    Y = DF['y'] #Selected the y from the column Result
    X = DF[features] #Selected the x to be the features
        X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y, test_size=validation_size, random_state=seed)


# Make predictions using KNeighborsClassifier on validation dataset
    knn = KNeighborsClassifier()
    knn.fit(X_train, Y_train)
    predictions = knn.predict(X_validation)
    print(accuracy_score(Y_validation, predictions))
    print(confusion_matrix(Y_validation, predictions))
    print(classification_report(Y_validation, predictions))

    # Test options and evaluation metric
    models = []
    models.append(('LR', LogisticRegression()))
    models.append(('LDA', LinearDiscriminantAnalysis()))
    models.append(('KNN', KNeighborsClassifier()))
    models.append(('CART', DecisionTreeClassifier()))
    models.append(('NB', GaussianNB()))
    # evaluate each model in turn

    for name, model in models:
        kfold = KFold(n_splits=10, random_state=seed)
        cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring=scoring)
        results.append(cv_results)
        names.append(name)
        msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
        print(msg)


def compare_algorithms():
    # Compare Algorithms
    fig = plt.figure()
    fig.suptitle('Algorithm Comparison')
    ax = fig.add_subplot(111)
    plt.boxplot(results)
    ax.set_xticklabels(names)
    plt.show()

def main(): #main program function
```

```
    DF = read_data() #call read data and save data frame to variable
    implement_machine_learning(DF)
    compare_algorithms()

main() #Call the main function
```