T.C. DOĞUŞ UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

COMPUTER ENGINEERING

A New Approach for Ensemble Based Demand Forecasting Using Machine Learning Methodologies in BigData Environment

Ph.D. Thesis by

A.Okay Akyüz

2013194006

Thesis Supervisor

Prof. Dr. Mitat Uysal

İstanbul, May 2019

	DOKTORA TEZ SAVUNMA SINAV TUTANAĞI	*Doküman No*	FR.1.46
		Yürürlük Tarihi	1.11.2017
		Revizyon Tarihi	1.11.2017
		Revizyon No	1
		Sayfa	1/1

~~SOSYAL BİLİMLER~~ / FEN BİLİMLERİ ENSTİTÜSÜ

Tarih : 31/05/2019

Anabilim Dalı : Bilgisayar Mühendisliği

Öğrencinin Adı Soyadı : A. Okay Akyüz

Öğrenci No : 2013194006

Tez Danışmanının Adı Soyadı : Prof. Dr. Mitat UYSAL

İkinci Tez Danışmanının Adı Soyadı : ———

Tezin Başlığı : A New Approach for Ensemble Based Demand Forecasting Using Machine Learning Methodologies in BigData Environment

Doğuş Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'nin 44.Maddesi uyarınca yapılan değerlendirmeler sonunda;

☒ **tezin kabul edilmesine** ☐ **tezde düzeltme verilmesine** ☐ **tezin reddedilmesine**

[✓ oy birliği] / ~~oy çokluğu~~ ile karar verilmiştir. Gereği için arz olunur.

Danışman Üye
Prof. Dr. Mitat UYSAL

Üye
Prof. Dr. Selim AKYOKUŞ

Üye
Prof. Dr. Aynur UYSAL

Üye
Doç. Dr. Murat Can GANİZ

Üye
Dr. Öğr. Üyesi Zeynep Hilal Kilimci

Anabilim Dalı Başkanı Onayı:
Doç. Dr. Hürevren Kılıç

Hazırlayan	*Kalite Onayı*	*Yürürlük Onayı*
Kalite Uzmanı Ceyda YILDIZ	Kalite Koordinatörü Prof. Dr.Mesut KUMRU	Rektör Prof. Dr. Ebru Ural

DECLARATION

I declare that, this thesis is my own work and has not been submitted in any form for another degree or diploma at any university and institution. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references given.

Ahmet Okay Akyüz

Signature : A. Okay

Date: 31.05.2019

For my endless love, my wife Özlem Akyüz, and for my shining stars, my sources of inspirations, my princesses and my daughters Sine Akyüz and Ela Akyüz.

ACKNOWLEDGEMENTS

Firstly, I want to thank to my advisor professor, Prof. Dr. Mitat Uysal for all his supports and challenging and incredible guidance during my thesis project. I am very proud and lucky to be his student since around 30 years. Learning is a continues process and I am learning continuously from him in diverse set of areas. He provided me with a better understanding of many concepts, as well as academic insight. He is a very valuable source for Computer Science students during the last decades in Turkey.

I would like extend my thanks to also Prof. Dr. Selim Akyokuş and Dr. Zeynep Hilal Kilimci. They have really valuable and smart scientific contributions to my study, especially on journal works. Thanks to them for attending to journal studies with their valuable scientific insights. And thanks to my colleagues Dr. Berna Atak Bülbül and M.Ali Ekmiş, for attending experimental studies and development works. Success always needs a good team playing and I am very happy and proud to work with such a good team.

Special thanks to Obase management board. Obase R&D Center supported my studies. Thanks to Obase General Manager Dr. Bülent Dal and board members Banu Dal and Haluk Alpay. And thanks to ŞOK Market Management, without trial data my theories couldn't be so improved and designed well.

And finally my special thanks goes to my family, my wife Özlem Akyüz and my daughters Sine Akyüz and Ela Akyüz. It was a very long, difficult journey and they were all supporting and motivating me during my studies. Thanks to all to be my supporters so patiently.

May, 2019 Ahmet Okay Akyüz

ABSTRACT

We live in the Artificial Intelligence era and most of applications in our daily life also business processes of companies are being embedded with Machine Learning technologies. Machine Learning algorithms allow us to make abstraction layer of many statistical problems. On the other hand, the variety of data types and data sizes are increasing very rapidly in all industries. These sort of problems are being handled by Big Data technologies. At this thesis, we propose and prove a new heuristic algorithm using ensemble learning methodology with its modified boosting strategy, by implementing to real life retail data for demand forecasting purpose. We demonstrate the results of our algorithm and advantages of the contribution of our demand forecasting methodology coming from common minded decision making philosophy using time series forecasting. Then, we also show that the applicability of advanced machine learning techniques will be beneficial for demand forecasting. These technics includes firstly Recurrent Neural Networks, and one more, which's roots come from statistical learning theory, called Support Vector Machines. Their aim is to increase estimation degree of unnormal demand accuracy for supply chains. Deep Learning technics also added to our approach by using open source machine learning libraries on Spark environment, which is in memory and distributed big data platform, takes the advantage of memory processing speed instead of relaying data to disks. The main beneficial output of our approach is to get extremely accuracy increase of demand forecasting results in each product category, it is demonstrated with the results of real life market data. SOK Market is one of the main discount store chain in Turkey with its 5500 stores and around 1500 active SKUs.

Demand forecasting for the purpose of replenishment is being observed as one of the basis critical problem for supply chains in retail industry, by the way of minimizing costs, optimizing stocks, and for also cost related optimization approaches to reduce retailers stock out problem. For retailers, more accurate demand forecasting results to manage operations with more optimum manner and results to maximization of customer sales with more revenue and of course profit. On the other hand, other crucial output of the stock out issue is already loyalty leak of the customers and their churn behaviours from one retailer to an other. If consumers are not able to meet with the products at stores which they want to buy, in generally they do not prefer to get a similar product item from an other category. They might make their shopping from nearest competitor retailer. In this thesis, we show that our proposed methodology performs much better accuracy at demand forecasting problem.

Keywords : Demand forecasting, replenishment, ensemble for time series, ensemble learning, machine learning, big data, deep learning, heuristic algorithms

ÖZET

Artık Yapay Zeka çağında yaşamaktayız ve gerek günlük hayatımızdaki işlerde, gerekse şirketlerin iş proseslerinin bir çoğunda makina öğrenmesi teknolojileri içerilmekte ve daha fazla kullanılmaktadır. Makina öğrenmesi algoritmaları istatiksel problemleri soyutlayarak çözüme kavuşturmamıza yardımcı olmaktadır. Öte yandan tüm endüstrilerde veri çeşitliliği ve veri hacimlerinin de çok yüksek bir hızla artıyor olduğu gözlenmektedir. Veri çeşitliliği ile birlikte yüksek hacimli verilerin bulunduğu ortamlarda hızlı karar verme ihtiyaclarının da ön plana çıkması sonucu, makine öğrenmesi problemleri artık günümüzde Büyük Veri teknolojileri yardımıyla ele alınabilmekte ve çözüme kavuşturulabilmektedir. Bu tezde arttırılmış strateji ile biçimlendirilmiş yeni bir sezgisel topluluk öğrenmesi algoritması önermekte ve gerçek veriler üzerinde de uygulayarak başarısını göstermekteyiz. Algoritmamızın sonuçlarını ve ortak akıl felsefesiyle karar vermeye dayalı zaman serilerinin kullanıldığı talep tahminlemesi metodolojimizin katkılarını da paylaşmaktayız. Daha sonrasında da ileri makina öğrenmesi tekniklerininin talep tahminlemesine uygulanabilirliğini göstermekteyiz. Bu kullandığımız tekniklerin başında ilk aşamada Yapay Sinir Ağları ve Derin Öğrenme uygulaması olan Tekrarlayan Sinir Ağları ve bir başka makina öğrenmesi tekniği olan, kökleri istatistiksel öğrenme teorisine dayanan Destek Vektör Makinaları gelmektedir. Bu yöntemleri de kullanmamızın amacı, tedarik zincirlerinin sonunda kamçı etkisi yapan bozulmuş talep tahminlemesinin doğruluğunu arttırmaktır. Bu yeni yaklaşımımıza derin öğrenmenin de gücünü ekleyerek, bu amaca yönelik olarak da açık kaynak kodlu Spark ortamını kullanmaktayız. Spark, yapısı gereği dağıtık büyük veri platformlarında disk yerine bellek içi çalışma mantığından dolayı daha hızlı işlem yapabilme avantajını sağlayan yenilikçi bir teknoloji sunmaktadır. Bu yeni yaklaşımımızın sağladığı en büyük faydası olarak da ürün kategorisi bazında talep tahminlemesi doğruluklarında son derece yüksek artış olduğu gözlemlenmekte ve deneysel sonuçlar ŞOK Marketlerinin gerçek verileri ile sunulmaktadır. ŞOK Market 5500 civarındaki mağazası ve 1500 aktif ürünü ile Türkiye'nin ileri gelen indirim mağazası zincirlerinden birisidir. Tedarik süreçleri amacıyla talebin tahminlenmesi problemi, maliyetlerin minimize edilmesi, stokların optimize edilmesi ve perakendecilerin yok satmalarını azaltacak maliyet odaklı optimizasyon yaklaşımları aslına baktığımızda perakende sektöründe temel ve kritik bir tedarik zinciri problemi olarak ortaya çıkmaktadır. Perakenciler daha doğru talep tahminlemesi yardımıyla operasyonlarını daha optimum düzeyde yönetebilirler, müşteri satışlarını arttırabilir ve daha yüksek karlılığa ulaşabilirler. Öte yandan yok satmanın diğer kritik bir sonucu da müşteri sadakatindeki azalma neticesinde müşteri kayıplarının artması olarak ortaya çıkabilmektedir. Eğer tüketiciler aradıkları ürünleri raflarda bulamazlar ise, genelde benzer bir ürünü farklı bir kategoriden almaktansa en yakın rakip mağazadan alış veriş yaparak perakendeciyi terk etme eğiliminde olabilmektedirler. Bu tezde özetle önerdiğimiz metodolojinin daha yüksek doğrulukta talep tahminlemesi sonuçlarına ulaştırıyor olduğunu deneysel olarak da göstermiş olmaktayız ve derin öğrenme tekniklerinin gücünü de algoritmamıza ekleyerek sonuçlarımızı pekiştirmekteyiz.

Anahtar Kelimeler: Talep tahminlemesi, tedarik zinciri, zaman serilerinde topluluk öğrenmesi, topluluk öğrenmesi, makina öğrenmesi, büyük veri, derin öğrenme, sezgisel algoritmalar

CONTENTS

TABLE LIST

FIGURE LIST

ABBREVIATIONS

ADABOOST	**:**	Adaptive Boosting
AE	**:**	Auto Encoder
ANN	**:**	Artificial Neural Network
ARMA	**:**	Autoregressive Moving Average
ARIMA	**:**	Autoregressive Integrated Moving Average
CART	**:**	Classification and Regression Tree
CNN	**:**	Convolutional Neural Networks
DBN	**:**	Deep Belief Networks
DeepSVM	**:**	Deep Support Vector Machine
DL	**:**	Deep Learning
EL	**:**	Ensemble Learning
ELM	**:**	Extreme Learning Machine
EMD	**:**	Empirical Mode Decomposition
GPU	**:**	Graphical Processing Unit
GRA	**:**	Grey Relational Analysis
HDFS	**:**	Hadoop File System
MAD	**:**	Mean Absolute Deviation
MAPE	**:**	Mean Absolute Percentage Error
MBO	**:**	Migrating Birds Optimization
MLE	**:**	Maximum Likelihood Estimator
NN	**:**	Neural Networks
RBMs	**:**	Restricted Boltzmann Machines
RBF	**:**	Radial Basis Function
RDD	**:**	Resilient Distributed DataSets
REML	**:**	Restricted Maximum Likelihood
RMASE	**:**	Revised Mean Absolute Scaled Error
RNN	**:**	Recurrent Neural Networks
RSM	**:**	Random Subspace Method

SES	**:**	Simple Exponential Smoothing
SKU	**:**	Stock keeping unit
SLFN	**:**	Single-hidden layer feedforward neural network
SVM	**:**	Support Vector Machine
SVR	**:**	Support Vector Regression
PC	**:**	Principal Components
PCA	**:**	Principle Component Analysis
TSE	**:**	Two-Stage Ensemble

1. INTRODUCTION

1.1 Demand Forecasting

A supply chain includes integration of information from many data sources related with product and many other dimensions between different stages. There are many interactions between every stage of the supply chain (Chopra & Meindl, 2001). Meanwhile, there is an operational level circumstance for the transfers of materials. These sort of information flow is necessary for managing the supply chains. Managers in a supply chain want to be sure that, all the business functions are to progress in synchronization between each other (Kushwaha, 2012). In business processes of retail, demand forecasting takes very crucial role on organizing the resources of companies in more accurate manner and retailers should meet with customer demands with affordable costs (Syntetos, Babai, Boylan, Kolassa & Nikoopoulos, 2012). As a result, demand-driven inventory management has extremely crucial part in supply chains.

The definition of demand forecasting can be made as the process of making predictions for consumer demands using statistical methods. Historical data is the main oil of forecasting algorithms. In both cases of fashion and food retails, it is being very essential for retailers to save product demands in adequate amounts at their stocks to escape from surpassing of inventory levels, and controlling out of stocks and operational costs such as transportation, with supplying the demands of consumers with higher service levels. Stable demanded products, for example paper towels and milk are generally easiest ones to forecast. Sometimes forecasting and reaching operational management decisions are not easy when the supply of basic items and also demand operations for the finished products. There are many uncertain parameters that affects very closely demand forecasting accuracy. Some examples of them are seasonality, social events, promotional effects, new social trends, terrorism, economical crisis, changes on climate conditions, special days, promotional behaviours of competitor companies at the market. Since demand changes according to time caused by these reasons, profitability, revenue and operational effectivity of retailers effected seriously and may be immediately (Ehrenthal, Honhon & Woensel, 2014). It is not easy to make adjustments on levels of stock for products as much as necessary sizes. That is needed to include them in demand forecasting models with related betting parameters. Fashion products and also electronics category items are difficult to forecast samples. For all cases, an estimation of forecasting error is vital while making planning for the supply chain (Box, Jenkins & Reinsel, 2008).

Also for short shelf-life items in grocery retail, it is significant to minimize of wasting because of their operational effectiveness. As well as better management of stocks for retailers, those behaviours effects financial performances of companies considerably. Since there is very aggressive competition at the market, most of the retail companies are searching for to improve their productivity and operational efficiency. Additionally, they aim to decrease their costs by means of making more effective operations (Villena & Araneda, 2017).

On the other hand, consumers prefer to answer to the offers of retailers which are really the lowest price and they also want to easily reach the product they are looking for. When consumers are not able to reach the product they are searching, they tend to meet with their needs by shopping from another quite possibly a competitor retail store. So there is really a very high competition in retail industry. These industry specific dynamics make inventory control systems and accurate demand forecasting much more important for companies (McGoldrick, 1997). As a result, retailers aim to keep the shelves as possible as full, to minimize disruptions, but not too much to increase stock costs.

With this thesis, we propose a new heuristic approach using ensemble algorithm for retail industry supply chains forecasting consumer demand. Our proposed algorithm run on the top of the different algorithms. The developed demand forecasting system consists of 11 kind of forecasting methods 9 of them are time series algorithms, and additionally Support Vector Regression (SVR), then a Deep Learning (DL) based method. This method works on an enriched data set collected from sales data, similar product's sales data and weather data. It is observed in the experimental results that, this new heuristic approach is helping very powerfully to improve the prediction accuracy around between %14 to %21 for the most of the product groups at production environment for real life data. The developed ensemble system uses a very specific integration strategy to combine the predictions of different forecasting methods. This method then applied to real data of a discount retail chain which has 5500 stores and 22 distribution centers by selling about different 1500 products (SKUs). In this thesis, our ensemble approach, pre-processing methods, experimental results and improvements obtained by the help of applying an ensemble of different forecasting methods including DL methodologies are presented. (Kilimci, Akyuz, Uysal, Akyokus, Uysal, Bulbul and Ekmis, 2019)

In Section 1, we explain the vitality of demand forecasting accuracy in retail supply chain operations and related works done by using Ensemble Learning and Deep Learning, and other innovator approaches. Then in Section 2, it is briefly introduced statistical forecasting concepts and Time Series algorithms. Following in Section 3 detailed information is given respecting to big data technologies and especially Spark improvements. In sections 4 and 5, we introduce Support Vector Machines (SVM) algorithm, which is one of the main player in our ensemble approach, and Deep Learning (DL) which becomes very popular after development of GPU technologies at recent years, respectively. In Section 6, we give detailed explanation of our heuristic ensemble approach modified for retail industry dynamics. We explain also our dataset and DL implementation, in which we use H2O open source library. In Section 7 and 8, we share our experimental results using real life customer sales data and conclusions reached at this thesis respectively.

1.2 Related Work

A brief summary of Ensemble Learning (EL) and Deep Learning (DL) related research for demand forecasting is explained before going into proposal of this thesis. Autoregressive Integrated Moving Average which is shortly called ARIMA, Autoregressive Moving Average which is named as ARMA models, exponential smoothing model, Holt's linear or exponential trend methods, average method and Holt-Winters approach are also some other time series methods (Hydman, 2017). Exponential smoothing methods can have different forms depending on the usage of trend and seasonal components and additive, multiplicative and damped calculations. Pegels presented different possible exponential smoothing methods in graphical form (Pegels, 1969). The types of exponential smoothing methods are further extended by Gardder (Gardner, 1985) to include additive and multiplicative damped trend methods. ARMA or ARIMA are common methods that are applied to obtain the best fit of a model, by means of applying to historical values of a time series (Gujarati, 2003).

Intermittent Demand Forecasting methods try to detect intermittent demand patterns that are characterized with zero or varied demands at different periods. Intermittent demand patterns occur in areas like fashion retail. On intermittent demand forecasting Croston proposed a common method (Croston,1972). This method uses a decomposition approach that uses separate exponentially smoothed estimates of the demand size and the interval between demand incidences. Its superior performance over the Single Exponential Smoothing (SES) method has been demonstrated by (Willemain, Smart, Shockor & DeSautels, 1994). Syntetos–Boylan addressed limitations of this method (Syntetos, 2001; Syntetos & Boylan, 2005) and Teunter, Syntetos and Babai added new ideas onto it (Teunter, 2011).

In recent years by the way, EL is used by researchers and made many different implementations. Song and Dai (Song & Dai, 2017), worked on a new approach, double deep Extreme Learning Machine (ELM) ensemble system focusing on the problem of time series forecasting. At Araque et al, he made the development of a DL based sentiment classifier (Araque, Corcuera-Platas, Sánchez-Rada & Iglesias, 2017). Tong et al showed two-stage ensemble (TSE) phase (Tong, Liu & Wnag, 2018). Qiua presented an ensemble method (Qui, Ren, Suganthan & Amaratunga, 2017) combination of Empirical Mode Decomposition (EMD) with DL algorithms in paper. Qi et al, shows in his study a combination of Ex-Adaboost learning strategy and the DL research based on

SVM, then proposes a new Deep Support Vector Machine (DeepSVM) (Qi, Wang, Tian & Zhang, 2016).

The nature of data is very important in classification problems (Bengio, Courville & Vincent, 2013). Geoff Hinton offered Deep Learning for the first time, in 2006. He announced a very important new approach for the purpose of feature extraction (Hinton, Osindero & Teh, 2006). Deep Belief Networks (DBN) which comes from Restricted Boltzmann Machines (RBM) is another form DL representation (Hinton, Osindero & Teh, 2006). In this representation connections between the layers are kept, on the other hand between units belongs to each layer is not supplied (Hinton, Osindero & Teh, 2006).

One an other representation of DL is Convolutional Neural Networks (CNN) (Qi, Wang, Tian & Zhang, 2016) and neural networks (NN) with multiple layers. Parameters which are decreases computation difficulty (Bengio, 2012; Qi, Wang, Tian & Zhang, 2016). In a NN, it is known that the output should be the input itself. Auto Encoder (AE) is a kind of NN which constructs the input. In the study of (LeChun, Bengio & Hinton, 2015; Kim, Yu, Kil & Lee, 2015) more different DL algorithms can be procured.

2. STATISTICAL FORECASTING

Generally, it is being used with some Time Series algorithms for demand forecasting of the next week for the level of each product (SKU) and store level. Then K-Fold cross validation methodology is used by assuming as $k = 3$ for testing purpose data and finally, the best fit of them is taken for the next weeks forecast. For example, setting $k = 3$ results in 3-fold cross-validation. In 2-fold cross-validation, orderly selected the dataset into two sets d_0 and d_1. Then train on d_0 and test on d_1, followed by training on d_1 and testing on d_0. When $k = n$ (the number of observations), the k-fold cross-validation is exactly the leave-one-out cross-validation. This is a general way for making demand forecasting in retail (Box, Jenkins & Reinsel, 2008).

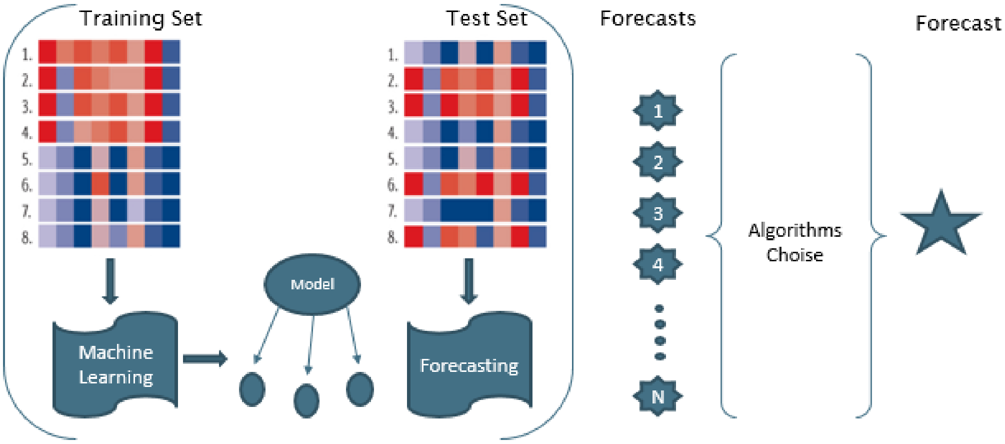


Figure 2.1 Demand Forecasting Flow

In this thesis, current demand forecasting system includes 9 different kind of Time-Series algorithms, for instance ARIMA, Holt-Winters, Exponential Smoothing and other Regression Models (Box, Jenkins & Reinsel, 2008). Means of the last 8 weeks sales quantity is also being used according to our experiments. For evaluation of different forecasting models, use general indexes are chosen to figure out accuracy (Chopra & Meindl, 2001). The formulation of these indexes are shown below:

1. MAPE (Mean Absolute Percentage Error)

Equation 2.1

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{A_t}$$

2. MAD (Mean Absolute Deviation)

Equation 2.2

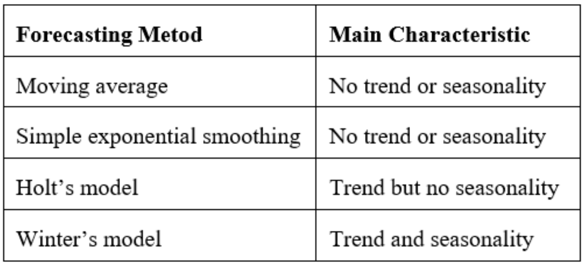
$$MAD = \frac{1}{n}\sum_{t=1}^{n}|F_t - A_t|$$

Where, F_t is the estimation and A_t is the actual values, n is also for the trial count.

When these two indexes decrease, it means forecasting accuracy of the model is better (Yue, Yafeng, Junjun & Chongli, 2007).

In push processes of supply chain, a manager should make a plan for the level of activity and then move that plan to production. They need to consider about transportation activities and any other processes. On the other hand, for the purpose of pull processes, they should make a plan for the level of available inventory and capacity of their organization. They don't need to make any execution for actual amounts.

Table 2.10.1 Forecasting Methods



Forecasting Metod	Main Characteristic
Moving average	No trend or seasonality
Simple exponential smoothing	No trend or seasonality
Holt's model	Trend but no seasonality
Winter's model	Trend and seasonality

3. BIG DATA ENVIRONMENT

Big data technologies become very popular in recent years. Data sizes have been increasing day by day very rapidly, companies may have petabytes of data at rest, on the other hand the internet of anything is doubling the amount of data in the world every 2 years. Companies realised that, data is not only structured relational databases resilient, they can also get benefit from many variety of data such as voice, image data, video, map, seismic data, sensor data.. etc. They know that valuable data in motion is streaming by them, but they have no easy way to capture it or analyze it. Actionable intelligence means that you can capture perishable insights in real-time by analyzing data in motion.

Hadoop files system (HDFS) comes with the solution of keeping data in distributed computing architecture structure. This solution provides companies also storage of their data on clustered commodity servers which are much more cheaper for them. With Apache foundation tools, there are also developed many open source solutions for the community to take the advantage of license free solutions. There are also big data solution provider companies to provide their clients reliable and scalable big data platforms, such as Hortonworks+Cloudera, MapR and Huawei.

3.1 What's Spark?

Spark is a data-processing engine which has a wide range of usage area, enabling us to work in parallel on large data sets with its in-memory engine performance. Interactive queries on large data sets, processes on data flowing from sensors or machine learning can be done with faster results on financial systems have often been done with Spark.

Spark is written in Scala language and has in-memory data processing capabilities (https://spark.apache.org/ access date :21.09.2018). In general, we can use Spark as an alternative to MapReduce. Using Hadoop, we can store the data in HDFS in distributed environment but with Apache Spark, so it is possible to process this data more easily and faster again with distributed approach, but this time at memory. Hadoop's MapReduce operating infrastructure and others reads and writes data to disk in each of the data processing steps. Spark is faster than data processing because it is optimized to work in memory (https://spark.apache.org/ access date :21.09.2018).

With Spark, it is possible to accelerate and generalize the MapReduce model with interactive querying, live data processing, different calculation components. Thus, for big data, it is possible to go beyond just batch data processing, to develop interactive queries and applications and to process live data (https://spark.apache.org/ access date :21.09.2018). Especially, it can be said that the distributed implementation of Artificial Learning algorithms is more efficient than Hadoop. While developing Java and Hadoop MapReduce applications is considered very low-level, developing an application with Spark is much more user friendly (http://www.firatdogan.net/post/128632607361/spark-1-apache-spark-nedir,, access date: 20.09.2018).

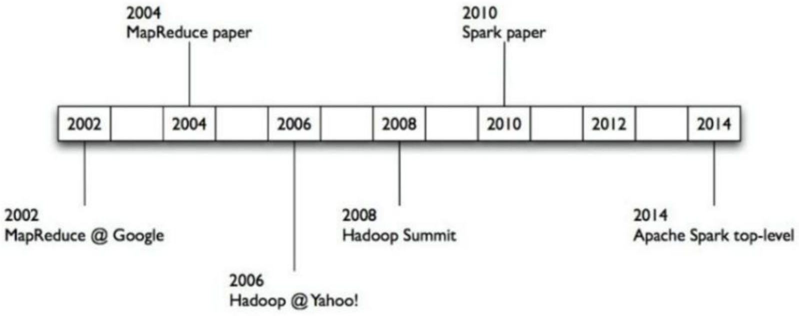


Figure 3.1 : Spark History

Rather than being a substitute for the Spark Hadoop, it can be said to be a member of the Hadoop family and to make up for some of the weaknesses of Hadoop's weaknesses. While Hadoop is designed to store and analyse petabytes of data on thousands of servers in a stable fashion, the Spark is designed to be relatively inexpensive, especially for processing in-memory and getting faster results. Spark does not offer any storage solution like HDFS but can read and write via HDFS.

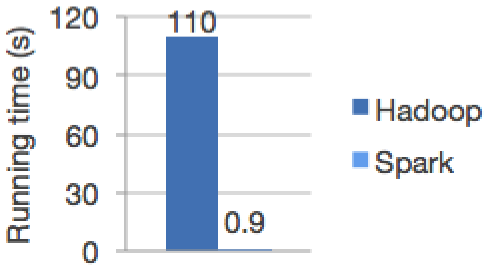


Figure 3.2: Logistic regression in Hadoop and Spark

It is possible to run the Spark cluster together with Amazon EC2, Mesos and YARN, except that it runs standalone. And also it is possible to test Spark applications through its interactive interface (spark-shell).

3.2 Spark advantages

Simplicity: Spark's capabilities are accessible through a rich set of APIs, and offer application developers the ability to integrate Spark into their applications. It is possible to quickly query, analyse and transform in a measurable manner. Spark's flexibility means that, it can be used in almost any situation which allows Spark to be able to handle all the problems and Spark to process petabytes of information on a cluster of physical and virtual servers in a distributed fashion. For data scientists and application developers to quickly implement a simple word counting sample in Hadoop can take a few lines of code on Spark (with Scala), while it can take a lot of code on others.

Speed: Spark's design was based on speed and speed to work on memory and disk. In the Daytona Gray Sort contest held in 2014 (the contest was about the processing of static data sets), Spark won the first prize by ranking in just 23 minutes on the SSD with a total of 100 TB. In the previous competition, Hadoop won the prize by ranking the same data in 72 minutes. It also appears that Spark systems are 100 times faster than Hadoop MapReduce in an environment with in-memory interactive queries.

Support: Spark supports many programming languages. These include Java, Python, R and Scala. Although Hadoop is often linked to the HDFS data storage system, it has built a good integration with leading data storage solutions that can work with or beyond Hadoop (https://spark.apache.org/ access date :21.09.2018).

3.3 Components of Spark

Spark Core: Spark Core includes key components such as distribution of memory management jobs, error recovery, storage and access to file systems. Resilient Distributed DataSets (RDD) is the most basic Spark component distributed in the calculation nodes used in Spark calculations. Spark Core provides the API needed to create and manage RDDs (http://www.firatdogan.net/post/128632607361/spark-1-apache-spark-nedir, 20.09.2018).

With RDD it is possible to handle a collection object as if it were exported. We can process a file by converting a file or collection object to RDD and calling these methods sequentially. RDD calculations help to rearrange and optimize data processing. RDD also provides fault tolerance, since RDD knows how to rebuild itself and how to re-process the data. RDDs are read-only and immutable. We can work with, but we cannot change. RDD supports two types of operations; One of them is transformation**,** which creates a new RDD and returns a new RDD that does not return a single value. Transformation methods do not take any action unless an action method is called, only retrieve one RDD and return a new RDD. The most commonly used functions are coalesce, pipe, aggregateByKey, reduceByKey, groupByKey, flatMap, filter and map. The other one is Action**,** which evaluates the operations and returns a new value. On the RDD object if an action function is called, all data processing queries are executed respectively and they return a result value.

Spark SQL: Spark component used to work with structured data. Standard SQL and Apache Hive language with Json, Parquet, Hive tables and so on, can be questioned. This SQL component and operations supported by RDDs facilitate the development of complex analytical applications.

Spark Streaming: The Spark Stream component is used to process persistent live data such as logs for a web server or instant user comments. This component provides processing for live data of RDDs similar to the APIs in Spark Core. All data such as disk, memory, real time streaming can be processed together.

MLlib: Spark library containing machine learning algorithms such as classification, clustering, regression, filtering. Ancillary components are included in the MLlib library for testing the correctness of the model installed or for transferring data from external sources. If you want to work on the level of optimization methods like gradient decent and contribute to the methods, this is possible with MLlib. All machine learning algorithms in MLlib are planned and optimized to work as well as other Spark components.

GraphX: It is possible to define different properties for each point and interlinks that make up Graph so that social sharing systems can be managed like parallel network using graphical algorithms.

Cluster Managers: Can integrate with cluster management systems such as Spark, Hadoop, Yarn, and Apache Mesos. Spark is distributed with an integrated cluster manager called Standalone Scheduler. If a system is built from scratch, Standalone Scheduler is ideal for quick and easy start. If a cluster manager such as Hadoop, Yarn or Mesos already exists, Spark can be configured to work on the control of these cluster managers. Spark does not need Hadoop to work, but HDFS using Hadoop API is used to extract data from storage systems such as local files, Amazon S3, Cassandra, Hive, Hbase and create RDDs (http://www.firatdogan.net/post/128632607361/spark-1-apache-spark-nedir, access date:20.09.2018).

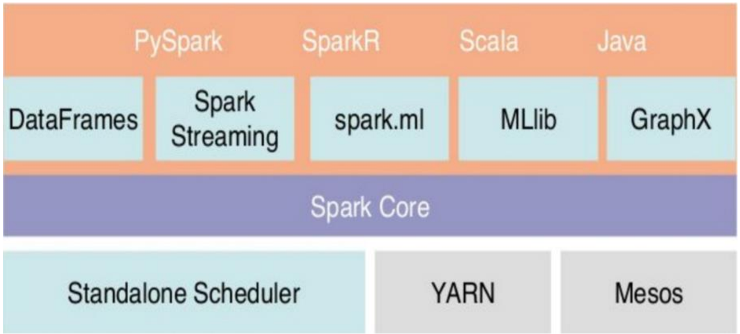


Figure 3.3 Spark Features

3.4 Spark Usage Areas

Stream processing: Application developers are working on stream type information such as log files and sensor information. This type of data is a constant stream of data, often coming from different sources. While it may seem appropriate to store these persistently flowing data on disk and to run past analysis, it can sometimes be important to process and manipulate the data as it arrives.

Machine learning: As data volume grows, machine learning approaches become more feasible and predictions can be made more accurate. The trained model is run on the new incoming and unknown dataset and the model is estimated. Spark's ability is to keep the data in memory and to quickly process repeated queries from the software on the data on the memory. Machine learning algorithms can be trained quickly and easily with this capability. Executing similar query types on a repeatable and scalable infrastructure significantly reduces the runtime of the algorithms used to find the best algorithm choices.

Interactive analytics: By running predefined static queries on BI application, it is possible to monitor sales, monitor productivity of production lines, or monitor stock market data. Often, business analysts and data scientists want to explore by asking questions on data. This cycle continues in the form of a questioning of the questions, observing the results and changing the question initially asked for more detailed exploration. This interactive query process requires a data processing platform, such as Spark, that can respond quickly and adapt to queries.

Data integration: Information generated in different parts of the company (eg human resources, sales and marketing, etc.) rarely matches each other. This adaptation problem in the data comes to the forefront in reporting and analysis processes. Extraction, transformation, and loading (ETL) operations are often used to retrieve, clean, and standardize incoming data from different data sources and eventually throw it into a different data storage area for analysis. The use of Spark and Hadoop, which significantly reduce ETL job times and operating costs, is on the rise (https://spark.apache.org/ access date :21.09.2018).

3.5 Spark MLLib-Machine Learning Algorithms

We can collect machine learning algorithms under two headings; supervised and unsupervised.

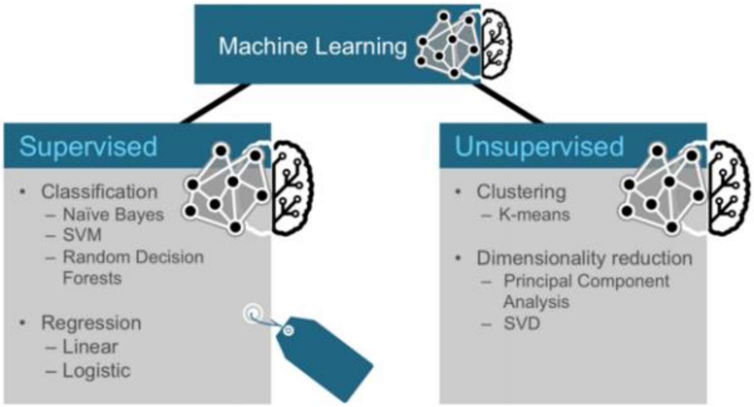


Figure 3.4 Machine Learning Algorithms

Usually, it includes machine learning algorithms classification, clustering and collaborative filtering. In addition, regression algorithms are also used.

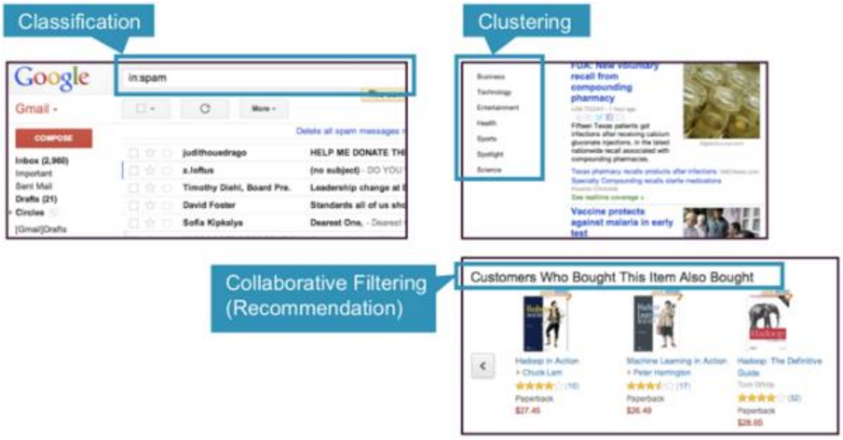


Figure 3.5 Commonly used algorithms

4. SUPPORT VECTOR MACHINES

As a machine learning technic, Support Vector Machines (SVM) is one of the leader classification algorithm. In 1992, the first scientific study on SVM was proposed by Vladimir Vapnik (Vapnik, 1998).

In this thesis also (Support Vector Regression) SVR algorithm is being used in our application, which is regression implementation of SVM with minor differences for continues variable classification problems. SVR algorithm is being used for continues variable prediction problems as a regression method that preserves all the main properties (maximal margin) as well as classification problems. The main idea of SVR is the computation of a linear regression function in a high dimensional feature space. The input data are mapped by means of a non-linear function in high dimensional space. It has been applied to many different areas, including speaker identification, prediction projects in finance, handwritten recognition, face recognition, object definition and benchmark of time-series prediction tests (Han, Kamber & Pei, 2013).

4.1 What is SVM

In machine learning, classification is one of the most common task. While using classification algorithms, one can assign new samples to one of the existing classes according to the basis of extracted previously unknown knowledge from the data (Alpaydin, 2014). As well as classification problems, SVM can be used for numeric predictions. In a numeric prediction problem suppose that, training points two classes set (x_i,y_i) is given for labeled supervised learning algorithm. For i in {1, ..., n}, for each point $x_i \in R^n$ and the set of label $y_i \in \{-1, 1\}$. First quadratic optimization problem is solved and then optimal hyperplane is being reached as shown in Equation (4.1), this is known as primal form of SVM. Variable count is similar to the size of training set.

Equation 4.1

$$\min \varphi\ (W,\xi) = \frac{1}{2}\ W^T W\ +\ C\ \sum_{1}^{n} \xi_i$$

Equation 4.2

$$y_i\ (\ W^T.\ x_i + b) \geq 1 - \xi_i) \quad and\ \ \xi_i \geq 0\ \text{ where } i = 1{,}2\ \ldots, n$$

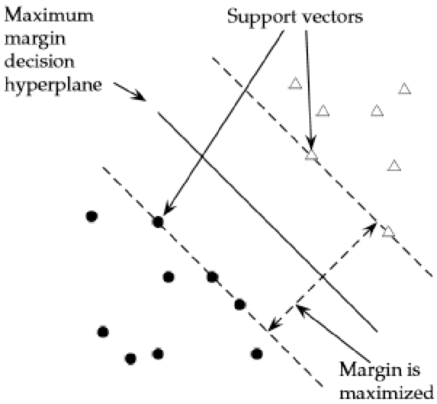


Figure 4.1 The support vectors margin of the classifier

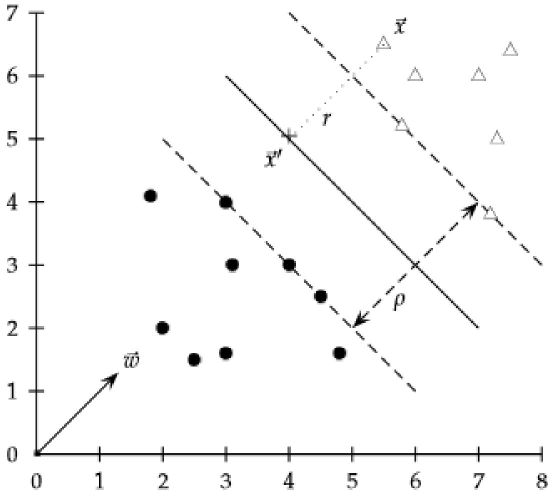


Figure 4.2 The support vectors linear separable case

Where, at both equations above ξ_i's are called as slack variables. Meanwhile, C is the parameter defined by user. The problem at Equation (4.3) can be transferred to a dual form:

Equation 4.3

$$\max\ Q(\alpha) = \sum_{k=1}^{m} \sum_{t=1}^{m} \alpha_k \alpha_t y_k y_t (x_k, x_t) - \sum_{t=1}^{m} \alpha_t$$

$$\sum_{t=1}^{l} \alpha_k y_k = 0 \ , \ 0 \leq \alpha_k \leq C \, , \quad \mathrm{k} = 1, \ldots, \mathrm{m}$$

In which, for kernel trick transformation, there are some quietly preferred substitutions for kernel selection function in such practical applications of SVM. The conventional and generally used kernel functions can be Multilayer Perceptron Kernel(MLP), Linear and Polynomial kernel types and Radial Basis Function(RBF). In SVM solutions, model selection problem is a result of the optimal selection of the regularization, also kernel parameters (Almasi & Rouhani, 2016).

Now the problem is decreased to new level, solving a quadratic function. This problem is exposed to linear equation constraints. In generally, quadratic optimization problems are sort of standardized and on the other hand difficult to solve problems. They are in very familiar category of mathematical optimization problems. There are many algorithms available for the solution of these kind of problems. We can construct our SVM model by the help of standardized quadratic programming (QP) approaches. At literature, there are very recent researches in that field purposing to take the advantage of the structure of the kind of QP problem, which comes from a support vector. Finally, there can be find much more complex, and on the other hand much faster and scalable libraries available to construct SVM, that are being used to perform more effective models. However, it might be very useful to cover the category of the solution for such kind of complex and difficult optimization problems. Finally, the solution contains building a dual problem in which a Lagrange multiplier α_i is regarding to each constraint.

$\alpha = (\alpha_1, \ldots, \alpha_n)$ is the sparse solution vector, for instance if α_i= 0 for most of the values of the training subset. This property is called as the sparseness of the SVM. At this set, support vectors are the points x_i belongs to non-zero α_i. Therefore, data points x_i corresponding to α_i= 0 has no effect and the optimal hyperplane is:

Equation 4.4

$$f(x) = \sum_{i=1}^{\#sv} \alpha_i y_i k\left(x_i, x_j\right) + b \ = 0$$

and the resulting classifier is:

Equation 4.5

$$y(x) = sgn\left[\sum_{i=1}^{\#sv} \alpha_i y_i k\left(x_i, x_j\right) + b\right]$$

According to Equation 4.3, for the SVM training process need large computational resources and takes time to train and also a large amount of kernel matrix is needed. Kernel Trick at non-linear SVM section is explained below on this study. Kernel matrix row count is equivalent to the row count of training set. Large computational cost for an implementation of application relies of that reason. Storage of this kernel matrix needs more space. Complexity of time and disk storage for the SVM training are $o(n^3)\ and\ o(n^2)$ respectively, regarding to (Angiulli & Astorino, 2010). So eventually, we need an approach to get away from execution complexity problems. The SVM in particularly defines the criteria of finding a suitable decision surface. Decision layer is as far as away from any of original data point, which is in maximum degree. In brief, we want to maximize the distance. The margin of the classifier is determined by the distance of the decision surface according to nearest data. This solution construction method tells us information about the decision function of an SVM is completely clarified by means of subset of dataset. The subset of data determines the coordinates of the boundary. Support vectors finally implied by those separator points.

4.2 Multiclass SVM

Support Vector Machines are practically used for two-class classifiers. In practise, the most common technique is to build |C| means, one-versus-rest classifiers. This is shortly called one-versus-all (OVA) at literature. The vector which classifies the test dataset for maximum margin should be chosen. Another classification strategy can be said as to prepare a set of one versus one classifiers. In this strategy our, aim is to make a decision for choosing the class which is picked by the most frequent classifier. During that period invites us constructing |C|*(|C|−1)/2 calculated classifiers, as a result, training phase time period probably decreases. For that reason, the training data of each classifier is transformed to become a smaller set. In fact, for solving multiclass problems these are not very effective approaches. By the construction of multiclass SVM, an other option is supplied, which is better. In this option, we need to construct a classifier on a feature vector $\Phi(\vec{x},\mathrm{y})$ with two-class. The vector is derived by using class and input features of the dataset. During validation phase, target class y is chosen by classifier;

Equation 4.6

$$\mathrm{y} = argmax_{\mathrm{y}'}\vec{w}^T\Phi(\vec{x},\mathrm{y}')$$

The margin for training period is the difference between the result for more correct result and for the closest to other result. Because of that, the QP equation needs a constraint that is defined;

Equation 4.7

$$\forall \mathrm{i}\ \forall \mathrm{y} \neq y_i\vec{w}^T\ \Phi(\vec{x_i}, y_i) - \vec{w}^T\Phi(\vec{x_i},\mathrm{y}) \geq 1-\xi_i$$

Among different sorts of linear classifiers, this method is generalized and can be extended to produce a multiclass formulation. This method is, on the other hand, can be defined as a basic sort of a generalization instance of classification. Result set of classes may not only unrelated or nominal labels, and also they might be randomly structured objects. In the SVM literature, by this way a study labelled as structural SVM, with relationships defined between each other with this classification.

4.3 Nonlinear SVM

Mostly we have introduced recently, datasets which are separable linearly (might by with some noise or with a few exceptions) are really well-handled. But real life data in general is not so easy to be separated linearly. When we interested into a one-dimensional situation first. In Figure 4.1, the superior data set is simply and clearly classified with a linear classifier. For the second data set, on the contrary, you can see that it is not easily possible talk about to make a single linear separation. Instead of this, we need to be able to choose an interval. For the solution of this issue, transformation of the data through a higher dimensional space by using a transformation is a strong alternative. After that, it is possible to use previously obtained linear classifier in a transformed and constituted high dimensional space. At the final side of the figure, for instance, it is clearly observed that, a linear boundary can easily make classification of this data.

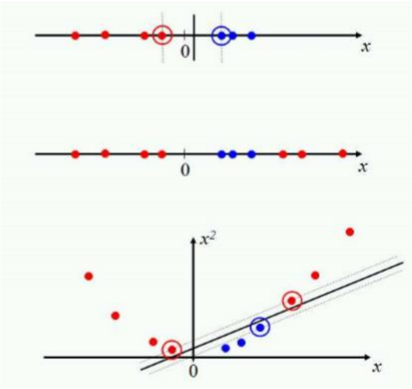


Figure 4.3 Kernel function sample for 1-D Space

When quadratic function is used to transform the data into two dimensions, the main attitude is to transform the original features to an other higher dimensional space. In this dimensional space the training set is separable. While doing this process we need to keep related dimensions of relationship between data points. At the result the classifier should still must be well-generalized. Linear classifiers, in briefly provide us an easy, fast and effective route of making transformation to an obtained higher dimensional space. This process is called in general “kernel trick”. In real it’s not a fake or trick, and it is as explained by the math functions that we explored in functional analysis. Linear classifier vector is calculated as dot product among data points, in SVM.

When K($\overrightarrow{x_i}$, $\overrightarrow{x_j}$) = $\overrightarrow{x_i}^T \overrightarrow{x_j}$. Equation 4.8 shows defined classifier.

Equation 4.8

$$f(\vec{x}) = \text{sign}(\ \sum_i \alpha_i\, y_i\ \ K(\overrightarrow{x_i},\vec{x}) + b)$$

We use this mapping function φ: $\vec{x}$ ⟶ φ($\vec{x}$) for transformation. Then the dot product of them becomes $\varphi(\overrightarrow{x_i})^T \varphi(\overrightarrow{x_j})$. This dot product calculation can be calculated basically and in an effective manner by means of original coordinates of data. After that, we cannot need to really transform for $\vec{x}$ ⟶ φ($\vec{x}$). Instead of this, we can calculate the value as K($\overrightarrow{x_i}$, $\overrightarrow{x_j}$) = $\varphi(\overrightarrow{x_i})^T \varphi(\overrightarrow{x_j})$, then we can use result in Equation (4.1). Kernel function K conforms a dot product in an extended space of features.

In functional analysis, kernel functions needs to satisfy some conditions. These conditions are addressed as valid kernel functions. In same cases, they are referred as Mercer kernels. So these kernel functions need to fulfil the condition of Mercer, which is for any g($\vec{x}$) so $\int$ g($\vec{x}$)^{2}dx becomes finite, so that:

Equation 4.9

$$\int K(\vec{x},\vec{z})\ g(\vec{x})g(\vec{z})dxdz\ ,\ z \geq 0$$

Such a transformation function tells us, there is a transformation for reproducing kernel for Hilbert space. According to definition of a Hilbert space a vector space included and which is closed under dot products. Because of this reason the dot product results same with the function we defined as K. There are two commonly called category of kernels. Radial basis functions are one kind of them and polynomial kernels are the others.

Polynomial kernels are defined at the form of K($\vec{x}, \vec{z}$) = $(1 + \vec{x}^T\vec{z})^d$. For this equation if we take d = 1, our kernel function transforms to a linear kernel. If d = 2, this kind of kernel becomes a quadratic kernel, which is very commonly used. There is an illustration below the quadratic kernel sample for 2-dimensional vectors;

$\vec{u}$ = (u_1, u_2), $\vec{v}$ = (v_1, v_2), lets consider K($\vec{u}, \vec{v}$) = $(1 + \vec{u}^T\vec{v})^2$. We want to demonstrate that is a kernel, saying K($\vec{u}$, $\vec{v}$) = $\varphi(\vec{u})^T\varphi(\vec{v})$ for some φ.

When we think about, $\varphi(\vec{u})$ = $(1\ u_1^2\ \sqrt{2}\,u_1u_2\ u_2^2\ \sqrt{2}\,u_1\ \sqrt{2}\,u_2)$. So, our function becomes:

Equation 4.10

$$K(\vec{u}, \vec{v}) = (1 + \vec{u}^T\vec{v})^2 = 1 + u_1^2v_1^2 + 2\,u_1v_1u_2v_2 + u_2^2v_2^2 + 2\,u_1v_1 + 2\,u_2v_2$$

$$= \varphi(\vec{u})^T\varphi(\vec{v})$$

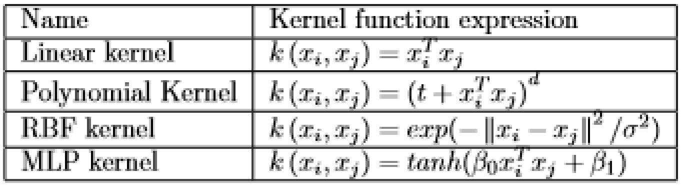
Gaussian distribution is the most known style of radial basis function and can be calculated as in Equation 4.11.

Equation 4.11

$$\mathrm{K}(\vec{x}, \vec{z}) = e^{-(\vec{x}-\vec{z})^2/(2\sigma^2)}$$

RBF is always equivalent to transforming data to Hilbert space which has infinite dimensions. For that reason, the radial basis function cannot be demonstrated by physically, like the way we did a quadratic kernel. Beyond these two main kernel function categories, there are detailed studies on developing other kernels. Most examples of them for text applications. Certainly, there are numerous investigations about string kernels. SVM terminology comes with its own modelling, and it is mostly different from the modelling technics preferred in data science problems. SVM depends on deep roots of statistical learning theory. According to the order of the polynomial, a polynomial kernel function gives us the permission to model variable connectivity's. All of common kernels are classified in table 4.3.1.

Table 4.3.1 Common Kernel Functions



Name	Kernel function expression
Linear kernel	$k\left(x_i, x_j\right) = x_i^T x_j$
Polynomial Kernel	$k\left(x_i, x_j\right) = \left(t + x_i^T x_j\right)^d$
RBF kernel	$k\left(x_i, x_j\right) = exp(-\left\Vert x_i - x_j \right\Vert^2 / \sigma^2)$
MLP kernel	$k\left(x_i, x_j\right) = tanh(\beta_0 x_i^T x_j + \beta_1)$

5. DEEP LEARNING

5.1 Deep Learning Technology

By the help of machine learning technics, the relationships and complex interactions among features can be analyzed and learned. In DL structure of human brain is simulated (Haykin, 2009). DL has become a very popular research topic among researchers and has been shown to provide impressive results in image processing, computer vision, natural language processing, bioinformatics and many other fields (LeChun, Bengio & Hinton, 2015; Bengio, 2012).

Learning is an unsupervised action naturally. We normally discover the structure of the objects outside of the world by observing them in unsupervised way, for example, actually we are not told the names of people at first sight, but we learn it by observation. Briefly, human and animal learning are mostly achieved in an un-supervised manner (LeChun, Bengio & Hinton, 2015). While there are much more studies on DL using supervised learning problems, DL also supports unsupervised learning techniques. In DL, the extraction of higher-level features in successive layers of a network is done automatically in an unsupervised manner.

DL is a methodology that leads some computational models. These models are composed of multiple processing layers for the purpose of learning the representations of data. DL algorithms are absolutely increased for the correctness perspective in some learning areas. By the way of using the backpropagation algorithm, DL has the ability to discover complex structures in huge data sets (LeCun, & Bengio & Hinton, 2015). In backpropagation algorithm, the algorithm should make modifications on parameters, which are harnessed to calculate the notation in each layer that dependent to the notation of the prior ones.

Machine-learning technology is being used in many different aspects of internet world, for instance web searches, social networks, recommendation engines. It is also incrementally represents in consumer products for example, smartphones and cameras.

DL technologies are being used as a machine-learning technic, to transcribe speech into text, identify objects in images, matching news items, user interests to relative items, and select related outputs of search and many other. Using conventional machine-learning techniques it's limited to process natural data in their detailed and normal style. For recent decades, building a DL system for pattern-recognition purpose required painful engineering. Because of computational resource necessity of problem. Additionally, pretty much domain expertise is needed for designing feature extractors. Resulting of transformation of the dataset through a more proper notation or vector of features data in the machine learning system. This learning system is called as a classifier.

DL methods are representation-learning methods you can compose non-linear or simple models. By the help of using some kind of transformations, very difficult models are able to be processed. Representation of learning gives permission to algorithm to get data set and to automatically find out the notations necessary for classification and also detection.

DL is helping to make improvements on solutions of learning subjects that have raised up in the artificial intelligence community during last decades. Therefore, it is acceptable in many fields of science, also business companies and institutes of government as well. In addition to it's successful implementations on image recognition and speech recognition, it can be said that, it is also very effective when comparing to traditional machine-learning models at some areas. There are more areas like on gene expression and disease and predicting mutations effects of DNA. More spectacular, DL has produced extremely advised outputs for some kind of similar tasks in natural language understanding. Mostly sentiment analysis, classification of topics, answering questions and also language translation.

It's known that, DL will be successful in the near future because it is easy to implement. Recent learning architectures and algorithms will only accelerate this progress, which are being developed for, DNNs currently. The learning algorithm calculates a gradient vector to properly adjust the weight vector. If the weights were increased by a small amount, this show that the amount of the error is increasing or decreasing.

Since the 1960s, it’s known that linear classifiers were able to only separate their input space into partial regions. These regions are namely half-spaces separated by means a hyperplane. On the other hand, some problems for example image recognition or speech recognition needs the IO function not to be sensitive to relevant variates of the input. Consider, the difference between two classes of objects. When we consider photo images in different poses and background they seem quite differently. But these images in the same position and similar backgrounds may look quite similarly.

5.2 The Future of Deep Learning

Since there is much more studies on DL using supervised learning problems, it’s disregarded additional effect on unsupervised learning areas. It is supposed that, in longer term unsupervised learning will be much more essential. Animal and human learning is generally classified as unsupervised manner. Normally, we discover outside world with observing them, we are not being defined the name of things when we firstly meet with them (LeCun, Bengio & Hinton, 2015).

Human vision and video games can be thought as some areas, which perform inactive vision systems mostly at classification problems. These methods generate effective results in machine learning via playing in video games differently.

Natural language understanding is also becoming an important field for DL (LeCun, Bengio & Hinton, 2015). It is expected that, systems which use recurrent neural networks (RNN)s understand whole documents or sentences. This can be considered as a better position when they try to learn strategies for selectively. RNN is described as an ANN class. In this network units show directed circuit. Directed circuit causes to go into a specific internal state for the network, that gives the permission it to show dynamic and temporal conduct. Eventually, in AI main efforts will be spent to make combination of representation learning with complex tasks. In spite of DL have been used for handwriting recognition and speech recognition for a long term, it’s necessary to change rule-based transformations with large vectors operations.

5.3 Some Other Forecasting Models

5.3.1 Artificial Neural Network

Artificial NN is a mathematical learning technic which is originated imitation of human brain philosophy. This approach mimics particularly our central nervous system. The main and basic model technic of an ANN is named as Single-Hidden Layer Feedforward Neural Network (SLFN). SLFN includes 3 main layers; first input layer then hidden layer. This layer includes non-linear neurons with activated functions and finally the output. Output is as summary of aggregation of neurons, which belongs to hidden layers.

SLFN output:

Equation 5.1

$$y = g\left(\sum_{j=1}^{h}\left(w_{j0}v_j + b_j\right)\right)$$

Equation 5.2

$$v_j = f\left(\sum_{i=1}^{n}\left(w_{ij}x_i + b_i\right)\right)$$

Where,

- x_i represents the input for neuron
- v_j hidden layer output j
- y SLFN output
- n is index for input features

- h is hidden layer neurons
- w_{ij} is the weight of input variable i
- w_{jo} is the weight of the output o
- b_i and b_j are the biases

5.3.2 Random Forests

Random decision forests are being used in classification and regression problems. Those algorithms are ensemble learning techniques which are combinations of bagging (bootstrap aggregating) and the other one is Random Subspace Method (RSM), one other method is The Classification And Regression Tree named (CART).

5.3.3 Hybrid Methods Based on ELM

The Extreme Learning Machine (ELM) is a very challenging method for constructing in estimation strategies. This feature makes it one of the best applicant to be a part of hybrid modelling for fashion industry, of course it is not perfect and without any problem at all and has some weaknesses (Liu & Ren & Choi & Hui, 2013). If we need to give a sample, Wong and Guo propose a new interesting approach, which comes from NN learning, for firstly generating sales forecasts and after that they use a heuristic approach for fine tuning to reach better results. Their solution makes integration between an ELM and an extended version of harmony search algorithm. Their aim is to increase generalization performance of the network. Their argument is that their novel model is much better than traditional NN and ARIMA models in terms of execution performance in sales forecasting for fashion. In Xia et al. (Xia, Zhang, Weng & Ye, 2013) they made examination for model forecasting based upon ELM using some adaptive metrics. Yu et al in his study proposed ELM and Grey Relational Analysis (GRA) for improvement of estimations in hybrid approach methodology (Yu, Hui & Choi, 2012). According to their experimental results on real life data set, their novel modelling strategy outperforms according to some other models on forecasting colour in fashion retail.

5.3.4 Other Hybrid Methods for Fashion Retail

Choi et al. worked on sales forecasting with a hybrid kind of SARIMA wavelet transform (SW) (Choi, Yu & Au, 2011). They show their method is very successful for highly variable seasonality and also weak seasonality. Thomassey and Fiordaliso at the literature, offer a hybrid method (Thomassey & Fiordaliso, 2006). Ni and Fan developed autoregression and decision tree combination called as ART method (Ni & Fan, 2011).

6. ENSEMBLE LEARNING AND INTEGRATION STRATEGY

6.1 What is Ensemble Learning

Ensemble learning is a learning paradigm that can significantly enhance the generalization ability of the base classifier, where a collection of a particular classifier is trained for the same task. The main vital and sufficient condition for a community to perform better performance than its individual members is the necessity for the correctness and diversity of the basic classifiers (Hansen & Salamon, 1990). Learning a community machine is much less costly than creating a learner with the same sensitivity. (Dietterich, 2000). The community tendency can greatly increase the generalization ability of the machine learner, but choosing some of them for the community may show even better performance (Zhou, Wu & Tang, 2002). The ensemble technique, which prepares the combinations of the outputs coming from different kind of classification models to create a collaborated output, became a very effective technic in a lot of fields (Ho, Hull & Srihari, 1994; Kittler, 1998). There are two main methods for ensemble named bagging and boosting (Han, Kamber &Pei, 2012).

To understand ensemble approach, patient problem is a very common sample (Han, Kamber & Pei, 2013). Suppose that you are a patient and you are looking for a diagnosis. You have some options to search for a couple of doctor’s medical opinion instead of taking only one of them idea. When all doctors get equal voting and you are giving decision according to majority count, this approach is called bagging. Logically, a majority voting should be better, since its common minded decision.

At classification and prediction problems final decision can be calculated according to the combination of the weighted decision of each algorithm in boosting ensemble methods. Adaboost (adaptive boosting) is one of the most popular boosting algorithm (Han, Kamber &Pei, 2012). EL methodology targets to increase predictive performance by making combination of many learning algorithms and converts their weaknesses into a strong learner (Rokach, 2010).

6.2 Ensemble Integration Strategy

Ensemble Learning Technic (Akyuz, Uysal, Uysal & Atak Bulbul, 2017) is being used by the way of combining the strengths of different algorithms into a single collaborated method philosophy. With this thesis we propose a new heuristic approach. Since every algorithm individually behave different reactions in different circumstances (Akyuz, Uysal, Uysal & Atak Bulbul, 2017).

At literature two main EL methodology are called as bagging (Breiman, 1996) and boosting (Han, Kamber & Pei, 2013). For boosting there are two main popular methods, AdaBoost (Freund & Schapire, 1997) and random forest (Breiman, 2001). Breiman proposed Random Forest and specially developed for decision trees (Breiman, 2001). One of the most popular boosting algorithm is AdaBoost (Han, Kamber & Pei, 2013). AdaBoost algorithm uses the weighted majority voting rule and Freund and Schapire showed in their research, in most cases AdaBoost performs better than bagging.

To make the integration of each algorithm, two integration strategies are used in our ensemble proposal. First one is (S_1), used by previous forecasting system (Box, Jenkins, Reinsel & Ljung, 2015; Chopra & Meindl, 2004). This approach finds the best one of algorithm and choices this one to forecast the demand. The second one is (S_2), used by our current proposed forecasting system (Akyuz, Uysal, Uysal & Atak Bulbul, 2017) (before DL integration), which calculates champions of the week and combines weighted results of them.

To integrate decisions of different forecasting methods, in our application, it is applied a modified version of boosting. In our boosting algorithm we have a heuristic approach, final decision is determined by regarding contribution of all algorithms like democratic systems. Our approach considers decisions of forecasting algorithms, those are performing at better on the previous n weeks of current year and k weeks trend of the last year. Since algorithms become better in their accuracy according to time, their weight points are rising. In this heuristic approach (Akyuz, Uysal, Uysal & Atak Bulbul, 2017) we just consider champion algorithms of the week not democratically at all. In other words, we arrive our final decision by collaborating the decisions of the most smart and suitable algorithms only.

In general approach, to make the evaluation and to measure the performance and accuracy of forecasting models MAD and MAPE accuracy measures are being used (Chopra & Meindl, 2004). Model accuracy is calculated for as small as MAPE values (Kushwaha, 2012). Replenishment for demand forecasting in retail industry requires some modifications for EL strategies, regarding to retail specific trends and dynamics. Calculation of weighted average of MAPE for each week of previous n weeks and additionally MAPE of the previous year's the same week and previous year following week's trend in Equation 6.1 are considered. So that forecasts become more reliable according to trend changes and seasonality behaviours. MAPE of each week is being calculated by the formula defined in Equation 2.1 above. Our system automatically changes weekly weights of each member algorithm by the way of their average MAPE for each week at store and product (SKU) level. For each algorithm an average MAPE is figured out according to Equation 6.1 below.

Equation 6.1

$$M_{avg} = \sum_{k=1}^{n} C_k \, M_k$$

In Equation 6.1, the constraint is $\sum_{k=1}^{n} C_k$ = 1. M_k gives us MAPE of the weeks int the formula.

First, MAPE of each algorithm is being calculated additionally, special days effect is also being considered. Special days can be defined to system from calendar manually. The system can be called as an expert system which learns from past data for special days and seasonality trends. Coefficients (C_k) are completely tunable according to dataset, for instance for a fashion or a food retailer the values of these parameters can be different.

We only take first 30% of the algorithms according to their performances. That is very clear, these parameters are very specific to structure of dataset. Ensemble strategy chosen is an important factor for seeing different results of experiences. Our application is parametric and users can tune such parameters after making some observations with dataset. Figure 6.1 shows at *y* axis average accuracy rate change according to empirical results for the subset of SOK Market data. We observe %78 peak at average accuracy for %30 algorithm rate with our sample dataset.

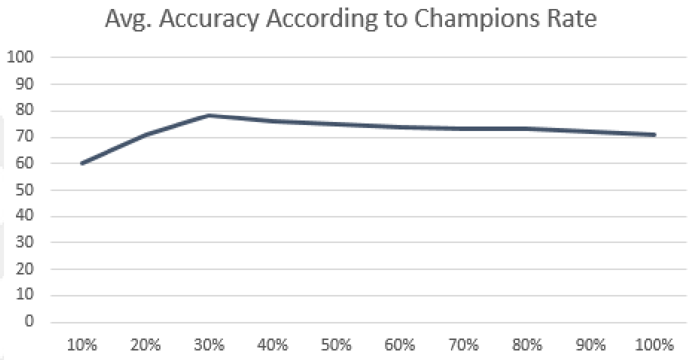


Figure 6.1 Avg. Accuracy change according to chosen algorithm rate

At last step, scaling is done before combined ensemble forecast according to Equation 6.2.

Equation 6.2

$$W_t' = \frac{W_t}{\sum_{j=1}^{k} W_j} \quad , t \; in \; (1, \dots, k)$$

Scaling makes our calculation more comprehensible. After scaling weight of each champion algorithm, the system gives ultimate decision according to new weights by considering the performance of each algorithm's with Equation 6.3.

Equation 6.3

$$F_{avg} = \sum_{j=1}^{k} F_j \mathrm{W}_j'$$

In Equation 6.3, the main constraint is $\sum_{j=1}^{k} \mathrm{W}_j' = 1$, k is the count of best algorithms and F_j is the forecast of these.

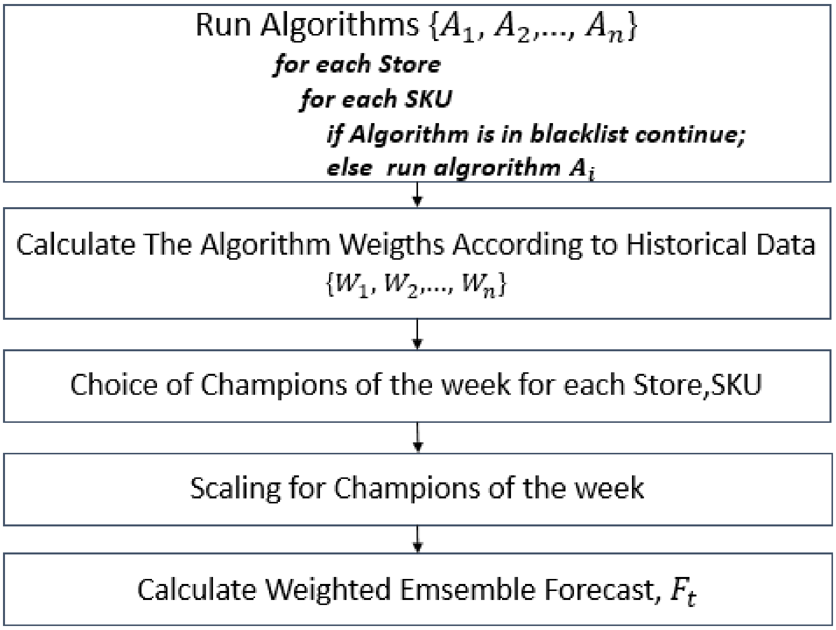
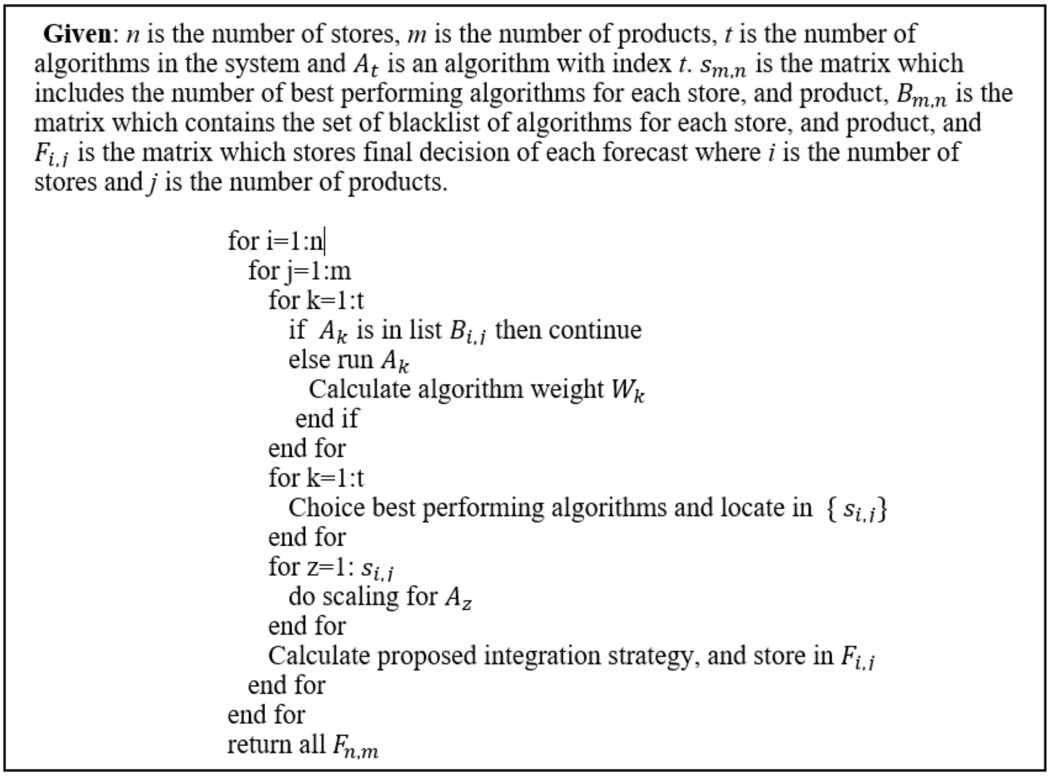

Run Algorithms $\{A_1, A_2, \ldots, A_n\}$
for each Store
for each SKU
if Algorithm is in blacklist continue;
else run algrorithm A_i
Calculate The Algorithm Weigths According to Historical Data
$\{W_1, W_2, \ldots, W_n\}$
Choice of Champions of the week for each Store,SKU
Scaling for Champions of the week
Calculate Weighted Emsemble Forecast, F_t

Figure 6.2 Ensemble Integration Strategy Process Flow



Given: n is the number of stores, m is the number of products, t is the number of algorithms in the system and A_t is an algorithm with index t. $s_{m,n}$ is the matrix which includes the number of best performing algorithms for each store, and product, $B_{m,n}$ is the matrix which contains the set of blacklist of algorithms for each store, and product, and $F_{i,j}$ is the matrix which stores final decision of each forecast where i is the number of stores and j is the number of products.

```
for i=1:n|
  for j=1:m
    for k=1:t
      if  $A_k$ is in list $B_{i,j}$ then continue
      else run $A_k$
        Calculate algorithm weight $W_k$
       end if
    end for
    for k=1:t
      Choice best performing algorithms and locate in  { $s_{i,j}$ }
    end for
    for z=1: $s_{i,j}$
      do scaling for $A_z$
    end for
    Calculate proposed integration strategy, and store in $F_{i,j}$
  end for
end for
return all $F_{n,m}$
```

Figure 6.3 The algorithm of the proposed demand forecasting system

We added a new improvement to our application. If an algorithm cannot go into the list of champions for a while, it is automatically marked as in the black list. As a result of this, computation time of the total system decreases. Figure 6.2 shows the summary of process flow of our ensemble integration strategy (S_2) and in Figure 6.3 there is algorithm flow modified for retail industry.

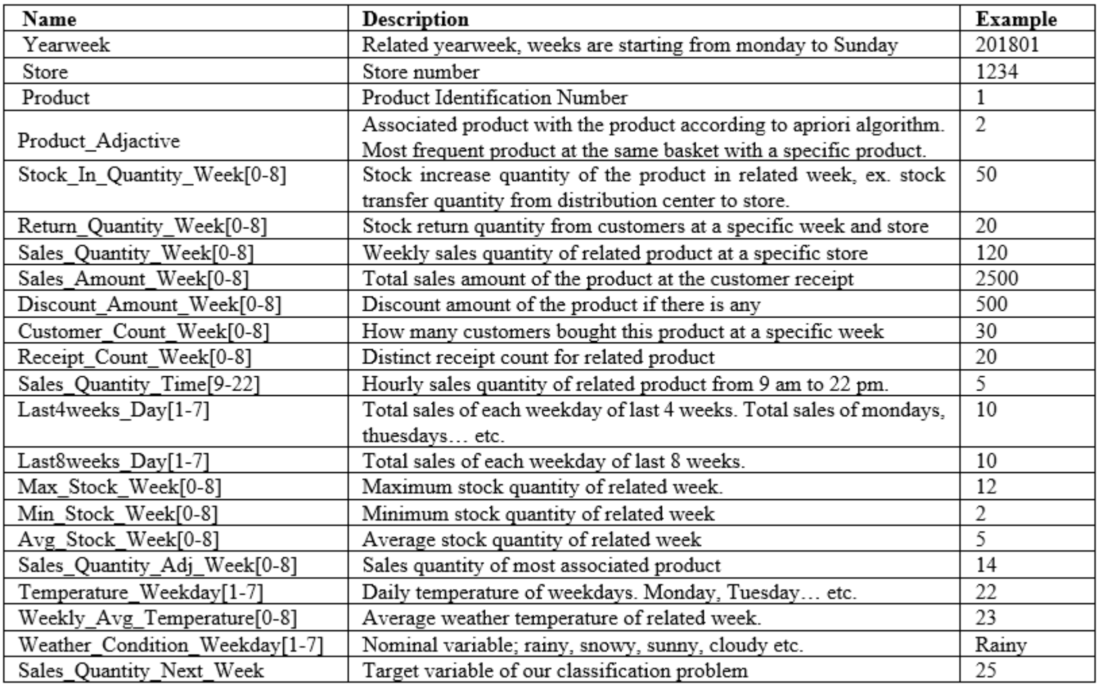
6.3 Deep Learning Forecasting Application

6.3.1 Dataset

For new DL forecasting model, we prepared a new enhanced dataset for the estimation of customer demands. The new dataset contains sales and stock data again but with enriched attributes and facts, those are not considered during our previous studies. Moreover, we also added weather data. Since there is a strong relation between weather and shopping behaviours (Lin & Tsai, 2016), enhancement of dataset with weather related data helps too much to DL algorithm.

To find the similarity among products for market basket analysis detection. We use Apriori Algorithm (Han, Kamber & Pei, 2013) in our solution to reach the correlated products. Apriori algorithm is a smart one to find out most related products. Briefly, enrichments of training dataset mentioned above makes estimates better when taking action with DL algorithms. The dataset contains data for product, store and week level. Last n weeks sales figures and final k weeks in the same period. Yearweek (year and week together, ex. 201801 is the first week of year 2018), Store Number and SKU are the primary keys of our dataset. Around 2 years log of data included in big data environment. Total data size is around 900 million records and consists of 157 features. Definition of dataset is showed at Table 6.3.1.1.

Table 6.3.1.1 Dataset Explanation



Name	Description	Example
Yearweek	Related yearweek, weeks are starting from monday to Sunday	201801
Store	Store number	1234
Product	Product Identification Number	1
Product_Adjactive	Associated product with the product according to apriori algorithm. Most frequent product at the same basket with a specific product.	2
Stock_In_Quantity_Week[0-8]	Stock increase quantity of the product in related week, ex. stock transfer quantity from distribution center to store.	50
Return_Quantity_Week[0-8]	Stock return quantity from customers at a specific week and store	20
Sales_Quantity_Week[0-8]	Weekly sales quantity of related product at a specific store	120
Sales_Amount_Week[0-8]	Total sales amount of the product at the customer receipt	2500
Discount_Amount_Week[0-8]	Discount amount of the product if there is any	500
Customer_Count_Week[0-8]	How many customers bought this product at a specific week	30
Receipt_Count_Week[0-8]	Distinct receipt count for related product	20
Sales_Quantity_Time[9-22]	Hourly sales quantity of related product from 9 am to 22 pm.	5
Last4weeks_Day[1-7]	Total sales of each weekday of last 4 weeks. Total sales of mondays, thuesdays… etc.	10
Last8weeks_Day[1-7]	Total sales of each weekday of last 8 weeks.	10
Max_Stock_Week[0-8]	Maximum stock quantity of related week.	12
Min_Stock_Week[0-8]	Minimum stock quantity of related week	2
Avg_Stock_Week[0-8]	Average stock quantity of related week	5
Sales_Quantity_Adj_Week[0-8]	Sales quantity of most associated product	14
Temperature_Weekday[1-7]	Daily temperature of weekdays. Monday, Tuesday… etc.	22
Weekly_Avg_Temperature[0-8]	Average weather temperature of related week.	23
Weather_Condition_Weekday[1-7]	Nominal variable; rainy, snowy, sunny, cloudy etc.	Rainy
Sales_Quantity_Next_Week	Target variable of our classification problem	25

6.3.2 Deep Learning Implementation with H2O

For DL modelling purpose, we use H2O library (Candel, Parmar, LeDell & Arora, 2018). H2O supplies an open source big data AI environment with a powerful machine learning library. DL algorithms can be implemented onto Hadoop Spark automatically using this library. So it provides in-memory parallel processing platform for tasks. In our project, we use version 3.16.0.8 version 64 bit system with 32 cores and 16GB memory maximum virtual use. Parallel processing while DL training phase, can be observed very clearly proportional to the number of neurons.

H2O's DL algorithm uses a multi-layer feedforward ANN. At the end output nodes are reached from the beginning and one direction. It does not give any feedback from posterior layers to the prior ones. H2O's DL is optimized for speed and accuracy by using in-memory compression. We selected Gaussian distribution option because our response variable is a continuous variable.

When data size of training phase is too much it can be possible to have performance issues with the computation. In our thesis project, we face a similar issue as well. For training phase, when all of the features are put into the model, H2O's DL algorithm consumes to much time to finalize the task. We use feature extraction and clustering methods for reducing modelling time. When we applied clustering algorithms to our raw data, we observe that the samples in each cluster are not distributed evenly and it required very long amounts of time to process.

6.3.3 Dimensionality Reduction And Clustering

To handle time consuming issue and to speed up processes, PCA (Principle Component Analysis) feature extraction algorithm is preferred as a pre-process step. PCA is a generally chosen dimensionality reduction method which finds significant features of all. Its purpose is to make clustering algorithm performs better by decreasing space dimension called Principal Components (P) (Vasan & Surendiran, 2016). In brief, Principal Components can be represented at Equation 6.4.

Equation 6.4

$$P_t = a_1X_1 + a_2X_2 + \cdots + a_zX_z$$

Where,

- P_t is t^{th} Principal Component,
- X_j is j^{th}original feature,
- a_{j} is the numerical coefficient for feature X_j.

After using PCA algorithm of H2O library, we reached 10 different principle components. Table 6.3.3.1 gives the cumulative proportion of explained variance of PC_i's, where we can see the first four principle components accounts for %98 of the total variance of the data. In general approach PCA guarantees that all dimensions of a manifold are orthogonal.

Table 6.3.3.1 PCA Results

Importance of Components	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Standard Deviation	851.2	229.7	119.2	102.7	79.1	53.5	50.5	45.2	42.9	35.1
Proportion of Variance	0.876	0.063	0.028	0.013	0.007	0.004	0.003	0.002	0.002	0.001
Cumulative Proportion	0.876	0.939	0.967	0.980	0.987	0.991	0.994	0.996	0.998	0.999

After dimensionality reduction step, we use K-Means as clustering algorithm which is a really greedy and fast one (Zhao, Deng & Ngob, 2018) that partitions data set into k clusters. Each member of clusters is as much as possible to close to the cluster center. After clustering step, we applied DL algorithm and obtained different DL models for each cluster. Figure 6.4 shows entire steps of processes we perform during the application of DL modelling.

.

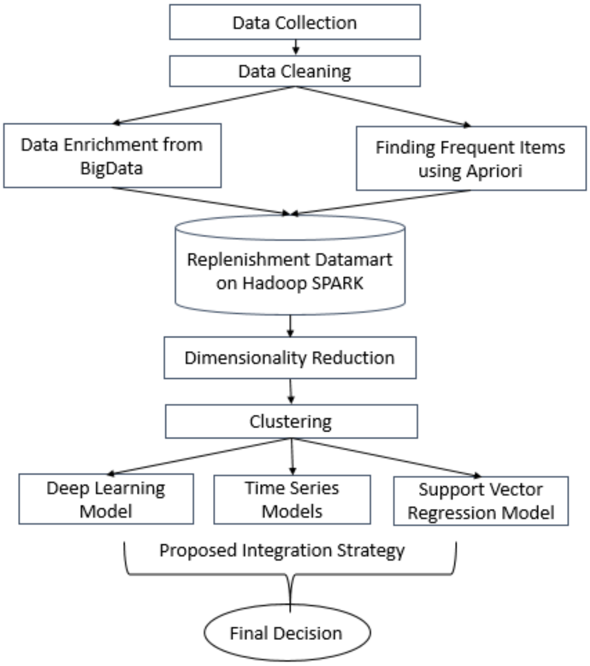

Data Collection
Data Cleaning
Data Enrichment from BigData
Finding Frequent Items using Apriori
Replenishment Datamart on Hadoop SPARK
Dimensionality Reduction
Clustering
Deep Learning Model
Time Series Models
Support Vector Regression Model
Proposed Integration Strategy
Final Decision

Figure 6.4 Flowchart of the proposed system

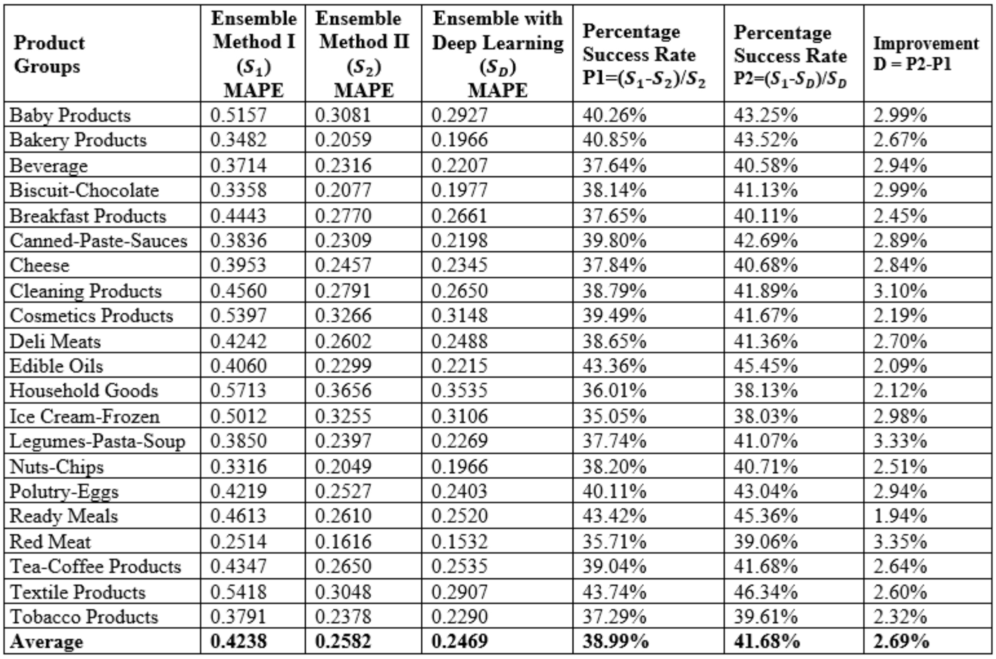
7. EXPERIMENTAL RESULTS

The forecasting is being done for each week to estimate demand of the next week. To make the integration of algorithms regarding to our ensemble approach, we use two strategies for our ensemble method. First one is named (S_1), used by our previous forecasting system (Box, Jenkins, Reinsel & Ljung, 2015; Chopra & Meindl, 2004), chooses the best method of all and uses that one to forecast the demand. The second integration strategy (S_2), used by our current forecasting system (Akyuz, Uysal, Uysal & Atak Bulbul, 2017) (before DL integration), which selects champions of the week and combines their results in a weighted manner.

We show the results of forecasting system in Table 7.1. Table 7.1 demonstrates MAPE of ensemble strategies S_1 and S_2 comparison. As it can be seen from Table 7.1, the percentage success rate of S_1 with respect to S_2 is computed on column 5.

With this thesis, a further improvement is also obtained by adding SVR and DL models to our proposed 9 algorithm based forecasting strategy. The column 4 of Table 7.1 shows MAPEs of 21 different groups of products after the addition of DL model. After the addition of our DL model, we observe significant better measures at our results. It can be seen very clearly in Table 7.1 in each step of integration strategies there is a significant increase in each product group. This results are very important for a retailer, mostly in sales continuously sales products.

Table 7.1 Demand forecasting improvements per product group



Product Groups	Ensemble Method I (S_1) MAPE	Ensemble Method II (S_2) MAPE	Ensemble with Deep Learning (S_D) MAPE	Percentage Success Rate P1=(S_1-S_2)/S_2	Percentage Success Rate P2=(S_1-S_D)/S_D	Improvement D = P2-P1
Baby Products	0.5157	0.3081	0.2927	40.26%	43.25%	2.99%
Bakery Products	0.3482	0.2059	0.1966	40.85%	43.52%	2.67%
Beverage	0.3714	0.2316	0.2207	37.64%	40.58%	2.94%
Biscuit-Chocolate	0.3358	0.2077	0.1977	38.14%	41.13%	2.99%
Breakfast Products	0.4443	0.2770	0.2661	37.65%	40.11%	2.45%
Canned-Paste-Sauces	0.3836	0.2309	0.2198	39.80%	42.69%	2.89%
Cheese	0.3953	0.2457	0.2345	37.84%	40.68%	2.84%
Cleaning Products	0.4560	0.2791	0.2650	38.79%	41.89%	3.10%
Cosmetics Products	0.5397	0.3266	0.3148	39.49%	41.67%	2.19%
Deli Meats	0.4242	0.2602	0.2488	38.65%	41.36%	2.70%
Edible Oils	0.4060	0.2299	0.2215	43.36%	45.45%	2.09%
Household Goods	0.5713	0.3656	0.3535	36.01%	38.13%	2.12%
Ice Cream-Frozen	0.5012	0.3255	0.3106	35.05%	38.03%	2.98%
Legumes-Pasta-Soup	0.3850	0.2397	0.2269	37.74%	41.07%	3.33%
Nuts-Chips	0.3316	0.2049	0.1966	38.20%	40.71%	2.51%
Polutry-Eggs	0.4219	0.2527	0.2403	40.11%	43.04%	2.94%
Ready Meals	0.4613	0.2610	0.2520	43.42%	45.36%	1.94%
Red Meat	0.2514	0.1616	0.1532	35.71%	39.06%	3.35%
Tea-Coffee Products	0.4347	0.2650	0.2535	39.04%	41.68%	2.64%
Textile Products	0.5418	0.3048	0.2907	43.74%	46.34%	2.60%
Tobacco Products	0.3791	0.2378	0.2290	37.29%	39.61%	2.32%
Average	**0.4238**	**0.2582**	**0.2469**	**38.99%**	**41.68%**	**2.69%**

Figure 7.1 below shows the box plots of mean percentage errors (MAPE) for each group of products after application of S_1 integration strategy. The box plots enable us to analyze distributional characteristics of forecasting errors for product groups. As it can be seen from the figure, there are different medians for each product group. So, the forecasting errors are usually dissimilar for different products groups. The inter-quartile range box represents the middle 50% of scores in data. The lengths of inter-quartile range boxes are usually very tall. This means that there are quite numbers of different prediction errors within products of a given group. Figure 7.2 shows the box plots of MAPEs of our forecasting system that apply our new ensemble approach called S_2 ensemble integration strategy (Akyuz, Uysal, Uysal & Atak Bulbul, 2017), which combines the predictions of 10 different forecasting algorithms. Figure 7.3 shows after adding DL to our system results. The lengths of interquartile range boxes are narrower when compared with the ones in Figure 7.1 and 7.2. Therefore, we have less spread of forecasting errors and our ensemble forecasting system with DL in our S_2 integration strategy (S_D) produces results

that are more accurate. Briefly, S_D evolves results around %2 or %3 with additional impact.

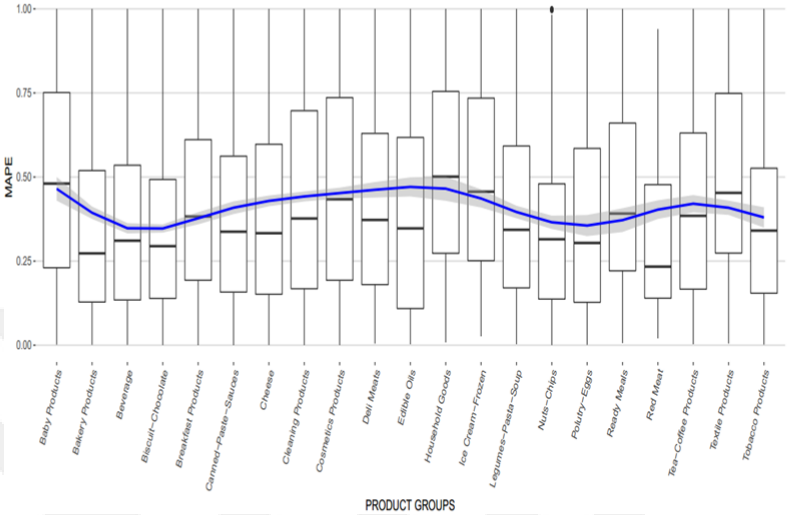


Figure 7.1 MAPE distributions for product groups with $\boldsymbol{S_1}$strategy

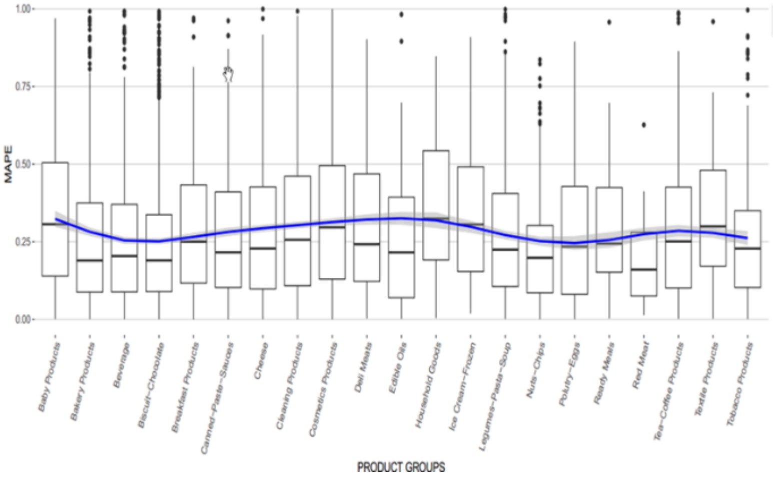


Figure 7.2 MAPE distributions for product groups with $\boldsymbol{S_2}$ strategy

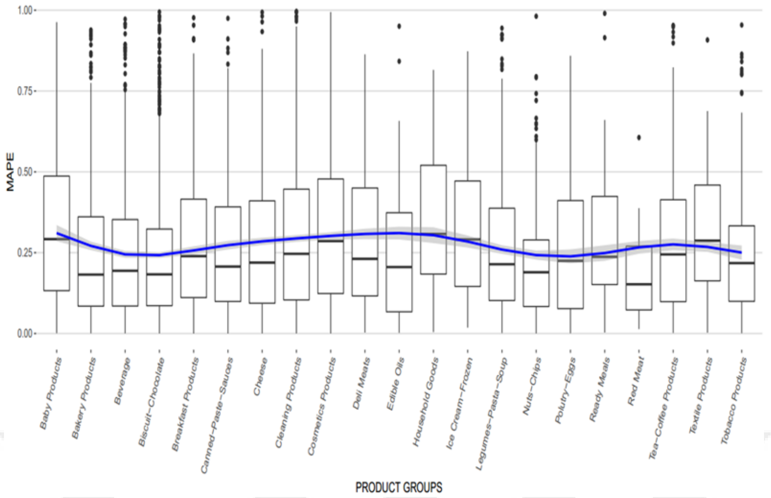


Figure 7.3 MAPE distributions for product groups with $\boldsymbol{S_D}$ strategy

Moreover, it is observed that the top benefitted product groups for the Method I accomplishes over 40% success rate on Baby Products, Bakery Products, Edible Oils, Poultry Eggs, Ready Meals, and Textile Products when the column of percentage success rate is analysed. The common point of these groups is everyone of them are being consumed by a specific customer segment, regularly. For instance, baby products group is being chosen by families who have kids, Edible Oils, Poultry Eggs, and Bakery Products are being preferred by frequently and continuously shopping customer segments and Ready Meals are being preferred by mostly bachelor, single and working consumers, etc. Furthermore, the inclusion of DL into the forecasting system indicates that some other consuming groups (for example; Cleaning Products, Red Meats and Legumes Pasta Soup) exhibits better performance than the others with additional over 3%.

In Figure 7.4, it can be seen accuracy comparison among ensemble strategies of one of the best performing sample group during 1 year period. Especially, Christmas and other seasonality effects can be seen very clearly at Figure 7.4 and integration strategy with DL (S_D) is performing with best accuracy.

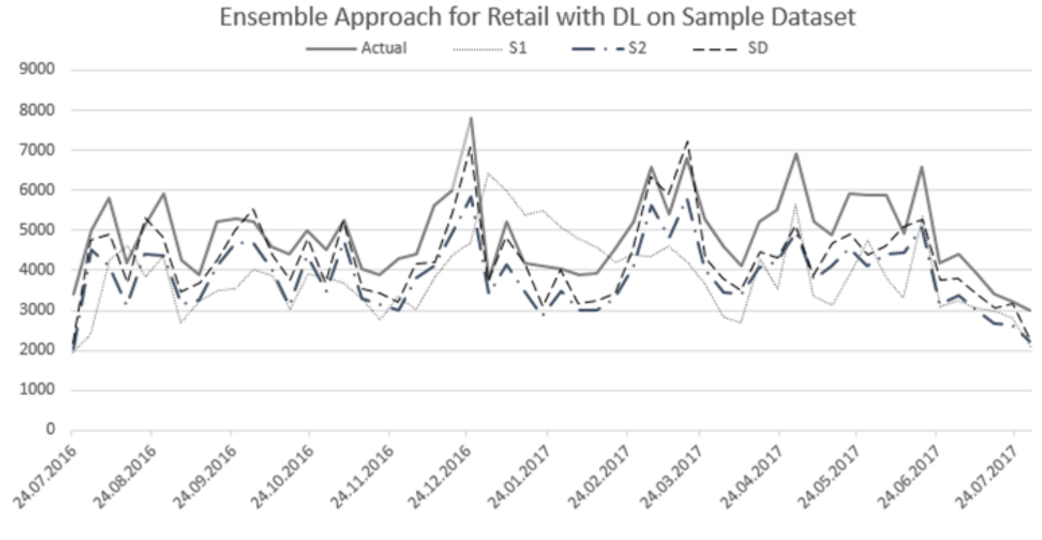


Figure 7.4 Accuracy comparison between ensemble methods

In Figure 7.4 above, it can be seen the comparison of each ensemble strategy and actual differences according to time period. And it is observed very clearly, differences and superiority of our proposed method with deep learning strategy.

8. CONCLUSION AND FUTURE STUDIES

Ensemble approach is a machine learning technique, which combines the predictions of a set of forecasting models to produce improved results. In this thesis, 11 different forecasting methods are used. These are time series algorithms, SVR, regression and DL methods. The times series and regression methods integration strategy is explained in our previous study in (Akyuz, Uysal, Uysal & Atak Bulbul, 2017). In this thesis, we also extend our method adding new DL model into the game. By means of our novel integration strategy, forecasts of different methods are combined together in a common mind manner. Proposed integration strategy makes results better with its collaborated decision making idea. And it is tested and used on real data of a fast growing retail chain. We observe significant improvements after implementation of both our novel integration strategy and DL model. (Kilimci, Akyuz, Uysal, Akyokus, Uysal, Bulbul and Ekmis, 2019)

The main additive of this thesis is, a novel ensemble methodology of different forecasting methods together with the DL model, which uses a rich set of features including associated products and outside weather data. Dimensionality reduction and clustering support to reduce time consuming in DL in this scenario. Common minded forecasts approach is more reliable with respect to the trend changes and seasonality behaviors. Moreover, the proposed approach performs very well integrating with deep learning algorithm on Spark big data environment. In this thesis, we compare results of three models where the model one selects the best performing forecasting method depending on its success on previous period with 42.4% MAPE on average. The second model with the novel integration strategy results in 25.8% MAPE on average. The last model with the novel integration strategy enhanced with deep learning approach provides 24.7% MAPE on average. As a result, the inclusion of deep learning approach to the novel integration strategy reduces average prediction error for demand forecasting process in supply chains.

As a future work, we plan to enrich the set of features by gathering data from other sources like economic studies, shopping trends, social media, social events and location based demographic data of stores. New variety of data sources contributions to DL can be observed. One more study is can be done to determine the hyper parameters for DL algorithm. In addition, we also plan to use other DL technics such as convolutional, recurrent and recursive NNs as base learners. Additionally, for optimization of coefficients there are some heuristic approaches in the literature and these can be also used, for instance MBO (Migrating Birds Optimization) (Duman, Uysal & Alkaya, 2012).

One more implementation area of our heuristic ensemble approach can be demand forecasting problems in airline industry. Passenger load factor forecasting is very essential for planning flights and also optimizing ticket prices for controlling and increasing revenue and profit amounts for airlines. When airline is able to sure that a flight will be less than planes capacity, they have some optional actions to make some decisions. They can change the plane with a smaller one and decrease oil consumption, crew and airport costs or they can decrease the ticket prices to get some more revenue etc.

REFERENCES

Akyuz, A.O., Uysal, M., Bulbul, B.A., Uysal, M.O. (2017). Ensemble approach for time series analysis in demand forecasting: Ensemble learning. DOI: 10.1109/ *INISTA*.2017.8001123

Almasi, O.N., Rouhani, M., (2016). Fast and de-noise support vector machine training method based on fuzzy clustering method for large real world datasets, TÜBİTAK.

Alpaydin, E. (2014). *Introduction to Machine Learning, Third ed.*, MIT Press, Cambridge, MA.

Angiulli F, Astorino A. (2010). Scaling up support vector machines using nearest neighbor condensation. *IEEE T Neural Network* 2010; 21: 351{357.

Bengio, Y., Courville, A., Vincent, P. (2013). Representation learning: A review and new perspectives, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 35, 1798–1828.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade,* Springer. pp. 437–478.

Box, G., Jenkins, G.M., Reinsel, G.C. (2008). *Time Series Analysis: Forecasting and Control*. Wiley, pp.93-194.

Choi, T.M., Yu, Y., Au, K.F. (2011). A hybrid SARIMA wavelet transform method for sales forecasting. *Decision Support Systems,* vol. 51, no. 1, pp. 130–140.

Chopra, S., Meindl, P., (2001). Supply Chain Management: Strategy, Planning, and Operation. Prentice Hall, pp.3-63.

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society, 23* (3), 289–303

Dietterich, T.G. (2000). Ensemble methods in machine learning. *Proceedings of the First International Workshop on Multiple Classifier Systems*, pp. 1–15.

Ehrenthal, J.C.F., Honhon, D., Woensel, T. V. (2014). Demand Seasonality in Retail Inventory Management. *European Journal of Operational Research*, vol. 238, pp. 527–539.

Gardner Jr, E. S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting* 4(1), 1–28.

Gujarati, D. (2003). *Basics Econometrics.* McGraw Hill, pp.835-865.

Han, J., Kamber M., Pei, M. (2012). *Data Mining : Concepts and Techniques*. USA: Morgan Kaufmann Publishers.

Hansen, L.K., Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp.993–1001.

Hinton, G.E., Osindero, S. & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 1527–1554.

Ho,, T., Hull, J., Srihari, S. (1994). Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp.66–75.

Kilimci,Z.H., Akyuz,A.O., Uysal,M., Akyokus,S., Uysal,O., Bulbul,B.A., Ekmis, M.A. (2019). An Improved Demand Forecasting Model Using Deep Learning Approach and Proposed Decision Integration Strategy for Supply Chain. *Complexity*, Volume 2019, Article ID 9067367.

Kittler, J. (1998). Combining classifiers: a theoretical framework, *Pattern Analysis and Applications*, vol. 1, pp.18-27.

Kushwaha, G.S. (2012) Operational Performance through Supply Chain Management Practices. *International Journal of Business and Social Science*, vol. 3, pp. 222-232, January 2012.

LeCun, L., Bengio, Y., Hinton, G. (2015). Deep learning Review, Macmillan Publishers Limited & *International Journal of Business and Social Science*, 3.

Li, C., Lim, A. (2018). A greedy aggregation–decomposition method for intermittent demand forecasting in fashion retailing, *European Journal of Operational Research*, 269, 860–869

Liu, N., Ren, S., Choi, T., Hui, C., Ng, S. (2013). Sales Forecasting for Fashion Retailing Service Industry: A Review, *Hindawi Publishing Corporation Mathematical Problems in Engineering* , Article ID 738675, 9 pages

McGoldrick,P.J. (1997). Consumer misbehaviour-Promiscuity or loyalty in grocery shopping?. *Journal of Retailing and Consumer Services*, vol. 4, pp. 73-81.

Ni, Y., Fan, F. (2011). A two-stage dynamic sales forecasting model for the fashion retail. *Expert Systems with Applications*, vol. 38, no. 3, pp. 1529–1536.

Pegels, C. C. (1969). Exponential forecasting: Some new variations. *Management Science*, 15(5), 311–315.

Qiu, X., Ren, Y., Suganthan, P.N., Amaratunga, A.J. (2017). Empirical Mode Decomposition based ensemble deep learning for load demand time series forecasting, *Applied Soft Computing*, 54. 246–255.

Qi, Z., Wang, Bo, Tian, Y., Zhang, P. (2016). When Ensemble Learning Meets Deep Learning: a New Deep Support Vector Machine for Classification, *Knowledge-Based Systems*, 107, 54–60.

Song, G., Dai, Q. (2017). A novel double deep ELMs ensemble system for time series forecasting. *Knowledge-Based Systems*, 134, 31–49.

Syntetos, A.A., Babai, Z., Boylan, J.E., Kolassa S., Nikoopoulos, K. (2015). Supply chain forecasting: Theory, practice, their gap and the future, *European Journal of Operational Research*, vol. 252, pp. 1-262.

Teunter, R.H., Syntetos, A.A., Babai, M.Z. (2011). Intermittent demand: Linking forecasting to inventory obsolescence. *European Journal of Operational Research*, 214 (3), 606–615.

Thomassey S., Fiordaliso, A. (2006). A hybrid sales forecasting system based on clustering and decision trees. *Decision Support Systems*, vol. 42, no. 1, pp. 408–421.

Tong, H., Liu, B. & Wang, S. (2018). Software defect prediction using stacked denoising autoencoders and two stage ensemble learning. *Information and Software Technology*, 96, 94–111.

Vapnik V. (1998), *Statistical Learning Theory*. New York, NY, USA: Wiley.

Villena, M.J., Araneda, A.A. (2017). Dynamics and stability in retail competition. *Mathematics and Computers in Simulation*, vol. 134, pp. 37-53.

Yue, L., Yafeng, Y., Junjun, G., Chongli, T. (2007). Demand Forecasting by Using Support Vector Machine, IEEE, *Third International Conference on Natural Computation* (ICNC 2007).

Yu, Y., Hui,C.L., Choi, T.M. (2012). An empirical study of intelligent expert systems on forecasting of fashion color trend. *Expert Systems with Applications*, vol. 39, no. 4, pp. 4383–4389.

Xia, M., Zhang, Y.C., Weng, L.G., Ye, X.L. (2012). Fashion retailing forecasting based on extreme learning machine with adaptive metrics of inputs, *Knowledge-Based Systems*, vol. 36, pp. 253–259.

Willemain, T. R., Smart, C.N., Shockor, J.H., DeSautels, P.A. (1994). Forecasting intermittent demand in manufacturing: A comparative evaluation of croston's method. *International Journal of Forecasting*, 10 (4), 529–538.

Zhou, Z.H., Wu, J.X., Tang. W. (2002). Ensembling Neural Networks: Many Could Be Better Than All. *Artificial Intelligence*, vol. 137, pp.239-263.

https://spark.apache.org/ access date :21.09.2018

http://spark.apache.org/docs/latest/mllib-linear-methods.html, access date:20.09.2018)

http://www.firatdogan.net/post/128632607361/spark-1-apache-spark-nedir, access date: 20.09.2018

BIOGRAPHY

Ahmet Okay Akyüz was born in Yenişehir/Bursa, in 1973. After secondary school, he attended Bursa Çelebi Mehmet High School and then for his bachelor degree he is graduated from Istanbul Technical University Mathematical Engineering Department, as second degree, in 1995. He got his masters degree in 1998 at Istanbul Technical University, Science and Technology Institute for Mathematical Engineering again. He worked for different positions during his professional 25 years career as C programmer, system analyst, etl developer, business intelligence consultant, datawarehouse architect, team leader, information management group leader, big data architect, data scientist and R&D center manager roles. His interested areas and studies are mostly in heuristic algoritms, data science, machine learning, datawarehouse and big data technologies. He started Ph.D program in Dogus University Computer Engineering section in 2013, he attended Inista Conference in 2017 with the study of Ensemble Approach in Demand Forecasting. His academical studies published in IEEE and Complexity journals in 2017 and 2019 respectively. He is now working as a Principal Consultant. He is married and has two daughters. He lives in Istanbul/Turkey.