

**Group Members:-**

**Anant Prakash Sharma – 17BIT0387**

**Market Basket Sales Prediction**

for

Data Mining Project

**Done Under the Guidance of**

**Prof. B. VALARMATHI**

**Associate Professor, SITE**

**Topic:** Approach and Solution of Big Mart Sales Prediction

**Objective:** The aim is to build a predictive model and find out the sales of each product at a particular store. Using this model, Big Mart can try to understand the properties of products and stores which play a key role in increasing sales.

**Abstract:** Big Mart is One Stop Shopping center and Free Marketplace. It buy, sell and advertise without fee or at low cost. We will explore the problem in following stages:

**Hypothesis Generation** – understanding the problem better by brainstorming possible factors that can impact the outcome

**Data Exploration** – looking at categorical and continuous feature summaries and making inferences about the data.

**Data Cleaning** – imputing missing values in the data and checking for outliers

**Feature Engineering** – modifying existing variables and creating new ones for analysis

**Model Building** – making predictive models on the data

**Dataset Description:-**

We have train and test data set, train data set has both input and output variable(s). We need to predict the sales for test data set.

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Item\_Identifier** | Unique product ID |
| **Item\_Weight** | Weight of product |
| **Item\_Fat\_Content** | Whether the product is low fat or not |
| **Item\_Visibility** | The % of total display area of all products in a store allocated to the particular product |
| **Item\_Type** | The category to which the product belongs |
| **Item\_MRP** | Maximum Retail Price (list price) of the product |
| **Outlet\_Identifier** | Unique store ID |
| **Outlet\_Establishment\_Year** | The year in which store was established |
| **Outlet\_Size** | The size of the store in terms of ground area covered |
| **Outlet\_Location\_Type** | The type of city in which the store is located |
| **Outlet\_Type** | Whether the outlet is just a grocery store or some sort of supermarket |
| **Item\_Outlet\_Sales** | Sales of the product in the particulat store. This is the outcome variable to be predicted. |

**Literaturesurvey:-**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. No**. | **Title of the paper and year** | **Algorithm(s)**  **used** | **Dataset being used** | **Performance measures** | **Gap identified** | **Scope for future work** |
| 1- | Improving organizational decision support: Detection of outliers and sales prediction for a pharmaceutical distribution company  (2017) | Dampened Pegels method and the SMAPE metrics | <https://www.sciencedirect.com/science/article/pii/S1877050917322305> | Using the Box-plot method and the rule created for the detection of outliers (customers and products), outliers may be identified in a fast and precise way, in order to take preventive initiatives. | It would be interesting to verify if, using a broader forecast horizon than the one used in this work, the results obtained could have an acceptable level of accuracy | The inclusion of a longer forecast horizon (e.g. 12 months) in sales forecasting may be of interest for the pharmaceutical distribution company to achieve a greater margin of maneuver in negotiating prices with suppliers. |
| 2- | Enhanced manufacturing storage management using data mining prediction techniques (2017) | Consumption Predictions Algorithms | <https://www.sciencedirect.com/science/article/pii/S235197891730803X> | The persistence method is, in every case, easy and largely outweighed by other tested techniques. If a separate prediction technique is used for each product, there is no universal response about which one of them is the best forecasting method. | If the same prediction technique is used for the forecasting of every product, the hospital method offers outstanding results. | The predictors based on data mining can offer enhanced solutions in many cases. |
| 3- | Combined Mathematical Morphology and Data Mining Based High Impedance Fault Detection (2017) | Data mining decision tree model | <https://www.sciencedirect.com/science/article/pii/S1876610217323871> | The proposed method can reliably detect high impedance fault for secured and stable operations in electrical power system | It is extensively tested on IEEE 13 and 34 test systems with the different system operating conditions | Better method can also be find for HIF detection with more better performance. |
| 4- | Determination of order specific transition times for improving the adherence to delivery dates by using data mining algorithms (2018) | KDD  (Knowledge discovery in database) | <https://www.sciencedirect.com/science/article/pii/S2212827118304050> | In this paper an approach for a databased determination of order specific transition times has been presented. | Approach discovered is not too friendly. | Future research has to evaluate the results of the three phases of the approach and how the approach can be implemented in practice, e.g. in ERP-systems |
| 5-  6- | The Influence of Customer Movement between Sales Areas on Sales Amount: A Dynamic Bayesian Model of the In-store Customer Movement and Sales Relationship (2017) A Data Mining Approach to Discover Critical Events for Event-Driven Optimization in Building Air Conditioning Systems (2017) | MCMC Procedure  and  Bayesian model  Measures of optimization reward  And Finding important variables with Random forest and . Finding the thresholds of events based on decision variable distance | <https://www.sciencedirect.com/science/article/pii/S1877050917316290>  <https://www.sciencedirect.com/science/article/pii/S1876610217364445> | In this study it is propose to use the latent space-time structure of the sales area for improvement of management actions.  This paper has effectively explored the data mining technique for EDO in the field of building optimal control. | Long-term future prediction of the latent variable r is difficult because of the amount of information.  The selections of events and event thresholds are critical for the performance of EDO | It will be necessary to model consumer’s purchase behavior from the individual level, and consider the impact on sales outcomes of the store.  The proposed method is easy to use and general as long as enough BAS data is available only. So better method can be implemented. |
| 7- | Re-discover Values of Data Using Data Jackets by Combining Cluster with Text Analysis (2017) | Text mining approach using Tf-idf | <https://www.sciencedirect.com/science/article/pii/S1877050917314679> | Discussed the new method to re-discover Data Jacket value | Not analyze the text data in different languages, to compare whether language has an impact on its value evaluation or not. | For the further research, it would have to increase the number of experiments, to get a higher degree of experiment accuracy. |
| 8- | Stock price prediction using support vector regression on daily and up to the minute prices (2018) | SVR prediction model | <https://www.sciencedirect.com/science/article/pii/S2405918818300060> | This study obtained results using SVR that were better than those of a null mean return random model. | This article does not propose transactional strategies applicable to the stock market. | Future studies may include a larger number of test stocks and markets other than those selected here. |
| 9- | Quality of classification with LERS system in the data size context (2018) | LERS which is a rule induction algorithm and MODLEM algorithm. | <https://www.sciencedirect.com/science/article/pii/S2210832717303393> | The paper analyzed and observed the similarities between [rule induction](https://www.sciencedirect.com/topics/computer-science/rule-induction) algorithms based on [Rough set theory](https://www.sciencedirect.com/topics/computer-science/rough-set-theory) at the same time dependency between quality of classification and percentage of certain rules were examined. | In case of increasing examples in large size, the algorithms shows wide range of difference in number of rules. | This work will contribute to the further research while approaching the process of mining in the way to get better prediction based on rough set theory and the next step in such analysis should be investigate advance this with rule based classifiers for the development. |
| 10- | A framework to guide the selection and configuration of machine-learning-based data analytics solutions in manufacturing (2018) | ML algorithm | <https://www.sciencedirect.com/science/article/pii/S2212827118303755> | Introduced a profile-based framework to systematically guide the selection and configuration of ML-based analytics solutions in manufacturing (ASMs). |  | It can be further extend and evaluate the generality of our framework by applying it to other kinds of use cases, e. g., for predictive maintenance. |
| 11- | Online Social Network Mining | Naive Bayes Text Classifier | Online Social Networks (OSNs) such as Facebook, Twitter, and LinkedIn | • it becomes more challenging when the textual information is not structured according to the grammatical convention  • mining rules from semi structure and unstructured as in the semantic web becomes a great challenge. | OSN data are largely user-generated content on social media sites which are vast, noisy, distributed, unstructured, and dynamic. These characteristics pose challenges to data mining tasks to invent new efficient techniques, algorithms and frameworks | As the number of social media users continues to grow, we will likelycontinue to see significant changes in the way we communicate and share information with each other. In this regard, the research community needs to continue to look to data mining approaches to provide us with the empowering ability to look deeper into these large data sets generated from OSNs in a more meaningful way |
| 12- | Research Issues and Future Directions in Web Mining: A Survey | Web Mining | Web Server Logs maintains a history of page requests. Information about the request, client IP address, request date/time, page requested, HTTP code, bytes served, user agent, are stored. Proxy Server Logs a caching mechanism which lies between client browsers and Web servers. | mining rules from semi structure and unstructured as in the semantic web becomes a great challenge | More research work need to be done on the web mining domain as it will rule the web in the near future. | The web usage mining algorithms are more efficient and accurate. But there is a challenge that has to be taken into consideration. Web cleaning is the most important process as researchers say 70% of the time is spent on data preprocessing. But data cleaning becomes difficult when it comes to heterogeneous data. Maintaining accuracy in classifying the data needs to be concentrated. Though many classification techniques exist the quality of clustering is still a question to be answered. |
| 13- | Correlation analysis between customer’s behaviour on website and actual purchase  (2018) | Web self-estimation | <https://www.sciencedirect.com/science/article/pii/S1877050918313954> | This result proved the linkage between customer's behaviour on website (micro) and actual sales trend (macro) | Clear linkage logic has to be found | Millennials will make up 60% of the global adult population by 2030, and to understand their behaviour on website will become more important to grasp the actual need of customer |
| 14- | A framework for attribute selection in marketing using rough computing and formal concept analysis  (2017) | As a data mining tool, rough set theory helps in obtaining decision rules about the problem | <https://www.sciencedirect.com/science/article/pii/S0970389617302598> | Further these rules are explored to identify the chief characteristics affecting the decisions total sales by using formal concept analysis. This helps the decision maker with a priori detection of the total sales. | More information can be find regardless of the rule based soft omputng. | The results obtained in pre-process can further be processed with the help of domain intelligence experts to obtain more specific characteristics of attributes affecting decisions. |
| 15- | A Study on Delivery Evaluation under Asymmetric Information in the Mail-order Industry  (2018) | Decision tree algorithm | <https://www.sciencedirect.com/science/article/pii/S1877050918313577> | This paper is presented investigating for comparing algorithms with the actual transaction data gathered from the mail-order industry in Japan | In this paper, the comparison of weaker learner algorithms is made. | Future work includes; 1) parameter tuning for the accurate classification, 2) examine another machine learning algorithm especially weak learner, 3) generate new transactional patterns with meta-heuristic algorithms. This work will require practical experiments and further survey studies |
| 16- | Personalized Course Recommender System Based on Hybrid Approach  (2018) | Dataset, End User Query | <https://www.sciencedirect.com/science/article/pii/S1877050917328314> | The intention of using neighbourhood formation is to find other similar learners based on their area of interest and requirements of targeted learner. | Overcomes the limitations of present individual recommendation system approach | In future neighbourhood generation to generate recommendations from knowledge base will be added |
| 17 - | Towards Intelligent and Sustainable Production: Combining and Integrating Online Predictive Maintenance and Continuous Quality Control  ( 2017) | ML and decision making tree | <https://www.sciencedirect.com/science/article/pii/S2212827117302457#aep-article-footnote-id4> | The paper makes a contribution to literature by combining and integrating online predictive maintenance and continuous quality control in an action research effort. | If wanted or necessary, it will be possible to combine and integrate additional concepts in the future | Future additions are data stream mining/analytics and artificial intelligence (AI) that can be used to get closer to real-time and improve the depth of the data mining and improve the support for decisions. Further, notification/warnings, improved visualization and further intelligent decision-making support will be of interest when continuing the action research effort. |
| 18- | Personality Assessment using Twitter Tweets  (2017) | Data clustering and test mining | <https://www.sciencedirect.com/science/article/pii/S1877050917314114> | Effects of the growth of social networks sites (SNS) are very heavy on the techniques developed for text mining in social networks. Text mining provides a proficient way to execute and make use of datasets. | Parameters of geolocation (latitude & longitude) can be added to regionalize the assessment. | Parameters of trend and taste can further be added to gain a comprehensive personality profile that can provide a fundamental ground for extending findings of this research to other disciplines |
| 19- | Finding Influentials in Social Networks using Evolutionary Algorithm  (2018) | Evolutionary algorithm | <https://www.sciencedirect.com/science/article/pii/S187775031830070X> | Investigated the influence of the evolutionary algorithm parameters on the result measured as the number of influenced nodes. | One of the disadvantages of the authors’ previous approach was the data representation. | In the future, the main focus should be pointed towards investigating whether some heuristics can be used in order to test a wider parameters space allowing to determine better parameters values sets maximizing the influence spread even further. |
| 20- | Data mining techniques for drug use research  (2018) | Decision Trees (DT) and Artificial Neural Network (ANN) | <https://www.sciencedirect.com/science/article/pii/S2352853218300683> | Never drug users perceive that adolescents use drugs because friends consume, to forget problems and to feel new sensations. | More best method can be approched | Further analysis can be done to distinguish between adolescent substance users and not users. |
| 21- | Towards Extended Data Mining: An Examination of Technical Aspects  (2018) | KDD and web crwling | <https://www.sciencedirect.com/science/article/pii/S1877050918318854> | Emphasized on the importance of incorporating additional data sets for better understanding of the data, and we have discussed technical aspects of implementing extended data mining | Shown a semi-automated process with human intervention. | In addition, beyond the basic methodology, more advanced issues can be explored, including ontology-based web crawling. |
| 22- | Privacy preserving data mining with 3-D rotation transformation  (2018) | Min–Max normalization and three dimensional rotation (3DR) | <https://www.sciencedirect.com/science/article/pii/S1319157816301227> | This paper presents a novel [privacy preserving](https://www.sciencedirect.com/topics/computer-science/privacy-preserving) [data transformation](https://www.sciencedirect.com/topics/computer-science/data-transformation) technique that can be used with different types of [data mining](https://www.sciencedirect.com/topics/computer-science/data-mining) models | Moreover the perturbation of data is not affecting much the data mining capability of the data mining model because of preservation of [Euclidean distances](https://www.sciencedirect.com/topics/computer-science/euclidean-distance) | Further study can be done to show its more high preserving capability. |
| 23- | Urban data and urban design: A data mining approach to architecture education  (2018) | Architecture data visualization and analysis | <https://www.sciencedirect.com/science/article/pii/S0736585317303416> | This research explored online data, either stored in web pages or informally generated by users and posted on social media, as a source of information for [urban planners](https://www.sciencedirect.com/topics/social-sciences/urban-planner) and designers. | Urban spaces can be used to compare and discuss the results. | More stages can be approached to improve the study in this research paper. |
| 24- | Visual Analysis System for Market Sales Data of Agricultural Products  (2018) | Data visualization and the core algorithms | <https://www.sciencedirect.com/science/article/pii/S2405896318312242> | The purpose of this paper is to accurately mine the dynamic information on the business circles and grasp the trends of consumer demand in time | Multiple algorithms such as clustering, path planning, sales trend prediction algorithms still have a lot of room for improvement | The interactive process of the system needs to be improved, and a system interaction solution based on user demands needs to be constructed. |
| 25- | Mining Negatives Association Rules Using Constraints  (2018) | SAT-Based Association Rules Minin | <https://www.sciencedirect.com/science/article/pii/S1877050918301583> | In this paper they proposed a novel approach for discovering strong negative association rules using constraints. | Experiments shows the feasibly of the proposed approach. | As a future work, it is planned to address the question of mining other kind of associations using a constraints approach such as exception and rares ones. |
| 26- | Data Mining Techniques Applied to a Manufacturing SME  (2017) | K-means clustering  Hierarchical clustering and self-organising maps | <https://www.sciencedirect.com/science/article/pii/S2212827117301087> | This paper has presented some datamining techniques namely data exploration, customer segmentation, Kohonen’s Self organising map, association rules, and time series to real data to extract knowledge and patterns for strategic decision making and forecasting | More analytics can be done to find the profitable approach. | Future work would look into data classification of manufacturing time and costs required to manufacture each product in order to assign a profitability label to each account based on purchases. |
| 27- | Research trends on Big Data in Marketing: A text mining and topic modeling based literature analysis  (2018) | Text mining and topic modeling procedure. | <https://www.sciencedirect.com/science/article/pii/S2444883417300268> | This research literature analysis focused on the application of Big Data in Marketing, in an attempt to identify the trends in these applied domains through different dimensions. | Some limitations should be pointed out which could also be addressed in future research. | It is a very dynamic subject, implying the results presented may need updating in a narrow time window. |
| 28- | Enhanced manufacturing storage management using data mining prediction techniques  (2017) | Time series forecasting  consumption prediction  data mining predictors | <https://www.sciencedirect.com/science/article/pii/S235197891730803X> | The persistence method is, in every case, easy and largely outweighed by other tested techniques. If a separate prediction technique is used for each product, there is no universal response about which one of them is the best forecasting method. | If the same prediction technique is used for the forecasting of every product, the hospital method offers outstanding results | Future analysis can be done using consumption prediction algorithm for this topic. |
| 29- | Reducing false positives in fraud detection: Combining the red flag approach with process mining  (2018) | Process mining technique | <https://www.sciencedirect.com/science/article/pii/S146708951630077X> | The goal of our research was to reduce the number of false positives in internal fraud detection. The confusion matrix proved very useful for our research as it shows the false positive rate. | Research subjected to particular limitations. | Adding further fraud detection patterns to the prototype can increase the true positive rate. |
| 30- | Text Mining to Understand the Influence of Social Media Applications on Smartphone Supply Chain  (2018) | Text Mining, Sentiment analysis | <https://www.sciencedirect.com/science/article/pii/S1877050918319690> | This work helped to identify a conceptual model of smart phone supply chain management with built in social media analysis, where information flow is faster and diversified compared to the traditional supply chain. | It should be noted that non-hashtag words also contain important potential information. | The identified framework can be used as a baseline in developing new models and algorithms to facilitate better supply chain decisions by verifying tweets based on a user’s credibility, fetching multiple sets of tweets for validating results, and in applying machine learning techniques to improve the accuracy and reliability of tweet mining models. |

**Existing Systems:-**

1. System: Windows 10 (64 - bit)
2. Programming Language: Python 3

**Proposed Method:-**

1. Hypothesis
2. Data Exploration
3. Data Cleaning
4. Feature Engineering
5. Model Building

**Sample Data:-**

Train:-

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Item\_Identifier** | **Item\_Weight** | **Item\_Fat\_Content** | **Item\_Visibility** | **Item\_Type** | **Item\_MRP** | **Outlet\_Identifier** | **Outlet\_Establishment\_Year** | **Outlet\_Size** | **Outlet\_Location\_Type** | **Outlet\_Type** | **Item\_Outlet\_Sales** | |
| FDA15 | 9.3 | Low Fat | 0.016047 | Dairy | 249.8092 | OUT049 | 1999 | Medium | Tier 1 | Supermarket Type1 | 3735.138 |  |
| DRC01 | 5.92 | Regular | 0.019278 | Soft Drinks | 48.2692 | OUT018 | 2009 | Medium | Tier 3 | Supermarket Type2 | 443.4228 |  |
| FDN15 | 17.5 | Low Fat | 0.01676 | Meat | 141.618 | OUT049 | 1999 | Medium | Tier 1 | Supermarket Type1 | 2097.27 |  |
| FDX07 | 19.2 | Regular | 0 | Fruits and Vegetables | 182.095 | OUT010 | 1998 |  | Tier 3 | Grocery Store | 732.38 |  |
| NCD19 | 8.93 | Low Fat | 0 | Household | 53.8614 | OUT013 | 1987 | High | Tier 3 | Supermarket Type1 | 994.7052 |  |

Test:-

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Item\_Identifier** | **Item\_Weight** | **Item\_Fat\_Content** | **Item\_Visibility** | **Item\_Type** | **Item\_MRP** | **Outlet\_Identifier** | **Outlet\_Establishment\_Year** | **Outlet\_Size** | **Outlet\_Location\_Type** | **Outlet\_Type** |
| FDW58 | 20.75 | Low Fat | 0.007565 | Snack Foods | 107.8622 | OUT049 | 1999 | Medium | Tier 1 | Supermarket Type1 |
| FDW14 | 8.3 | Reg | 0.038428 | Dairy | 87.3198 | OUT017 | 2007 |  | Tier 2 | Supermarket Type1 |
| NCN55 | 14.6 | Low Fat | 0.099575 | Others | 241.7538 | OUT010 | 1998 |  | Tier 3 | Grocery Store |
| FDQ58 | 7.315 | Low Fat | 0.015388 | Snack Foods | 155.034 | OUT017 | 2007 |  | Tier 2 | Supermarket Type1 |
| FDY38 |  | Regular | 0.118599 | Dairy | 234.23 | OUT027 | 1985 | Medium | Tier 3 | Supermarket Type3 |

Hypothesis

**Store Level Hypotheses:**

1. **City type:** Stores located in urban or Tier 1 cities should have higher sales because of the higher income levels of people there.
2. **Population Density:** Stores located in densely populated areas should have higher sales because of more demand.
3. **Store Capacity:** Stores which are very big in size should have higher sales as they act like one-stop-shops and people would prefer getting everything from one place
4. **Competitors:** Stores having similar establishments nearby should have less sales because of more competition.
5. **Marketing:** Stores which have a good marketing division should have higher sales as it will be able to attract customers through the right offers and advertising.
6. **Location:** Stores located within popular marketplaces should have higher sales because of better access to customers.
7. **Customer Behavior:** Stores keeping the right set of products to meet the local needs of customers will have higher sales.
8. **Ambiance:** Stores which are well-maintained and managed by polite and humble people are expected to have higher footfall and thus higher sales.

**Product Level Hypotheses:**

1. **Brand:** Branded products should have higher sales because of higher trust in the customer.
2. **Packaging:** Products with good packaging can attract customers and sell more.
3. **Utility:** Daily use products should have a higher tendency to sell as compared to the specific use products.
4. **Display Area:** Products which are given bigger shelves in the store are likely to catch attention first and sell more.
5. **Visibility in Store:** The location of product in a store will impact sales. Ones which are right at entrance will catch the eye of customer first rather than the ones in back.
6. **Advertising:** Better advertising of products in the store will should higher sales in most cases.
7. **Promotional Offers:** Products accompanied with attractive offers and discounts will sell more.

## Data Exploration

Code

import pandas as pd

import numpy as np

#Read files:

train = pd.read\_csv("train.csv")

test = pd.read\_csv("test.csv")

train['source']='train'

test['source']='test'

data = pd.concat([train, test],ignore\_index=True,sort=True)

print (train.shape, test.shape, data.shape)

output

(8523, 13) (5681, 12) (14204, 13)

data.apply(lambda x: sum(x.isnull()))

#Filter categorical variables

categorical\_columns = [x for x in data.dtypes.index if data.dtypes[x]=='object']

#Exclude ID cols and source:

categorical\_columns = [x for x in categorical\_columns if x not in ['Item\_Identifier','Outlet\_Identifier','source']]

#Print frequency of categories

for col in categorical\_columns:

print ('\nFrequency of Categories for varible %s'%col)

print (data[col].value\_counts())

output

Frequency of Categories for varible Item\_Fat\_Content

Low Fat 8485

Regular 4824

LF 522

reg 195

low fat 178

Name: Item\_Fat\_Content, dtype: int64

Frequency of Categories for varible Item\_Type

Fruits and Vegetables 2013

Snack Foods 1989

Household 1548

Frozen Foods 1426

Dairy 1136

Baking Goods 1086

Canned 1084

Health and Hygiene 858

Meat 736

Soft Drinks 726

Breads 416

Hard Drinks 362

Others 280

Starchy Foods 269

Breakfast 186

Seafood 89

Name: Item\_Type, dtype: int64

Frequency of Categories for varible Outlet\_Location\_Type

Tier 3 5583

Tier 2 4641

Tier 1 3980

Name: Outlet\_Location\_Type, dtype: int64

Frequency of Categories for varible Outlet\_Size

Medium 4655

Small 3980

High 1553

Name: Outlet\_Size, dtype: int64

Frequency of Categories for varible Outlet\_Type

Supermarket Type1 9294

Grocery Store 1805

Supermarket Type3 1559

Supermarket Type2 1546

Name: Outlet\_Type, dtype: int64

The output gives us following observations:

1. **Item\_Fat\_Content:** Some of ‘Low Fat’ values mis-coded as ‘low fat’ and ‘LF’. Also, some of ‘Regular’ are mentioned as ‘regular’.
2. **Item\_Type:** Not all categories have substantial numbers. It looks like combining them can give better results.
3. **Outlet\_Type:** Supermarket Type2 and Type3 can be combined. But we should check if that’s a good idea before doing it.

Data Cleaning

This step typically involves imputing missing values and treating outliers. Though outlier removal is very important in regression techniques, advanced tree based algorithms are impervious to outliers. So I’ll leave it to you to try it out. We’ll focus on the imputation step here, which is a very important step.

### Imputing Missing Values

#Determine the average weight per item:

item\_avg\_weight = data.pivot\_table(values='Item\_Weight', index='Item\_Identifier')

#Get a boolean variable specifying missing Item\_Weight values

miss\_bool = data['Item\_Weight'].isnull()

#Impute data and check #missing values before and after imputation to confirm

print ('Orignal #missing: %d'% sum(miss\_bool))

#data.loc[miss\_bool,'Item\_Weight'] = data.loc[miss\_bool,'Item\_Identifier'].apply(lambda x: item\_avg\_weight[x])

data.loc[miss\_bool,'Item\_Weight'] = data.loc[miss\_bool,'Item\_Identifier'].apply(lambda x: item\_avg\_weight.at[x,'Item\_Weight'])

print ('Final #missing: %d'% sum(data['Item\_Weight'].isnull()))

output

Orignal #missing: 2439

Final #missing: 0

#Import mode function:

from scipy.stats import mode

#Determing the mode for each

outlet\_size\_mode = data.dropna(subset=['Outlet\_Size']).pivot\_table(values='Outlet\_Size', columns='Outlet\_Type',aggfunc=(lambda x:mode(x).mode[0]), dropna=True)

print ('Mode for each Outlet\_Type:')

print (outlet\_size\_mode)

#Get a boolean variable specifying missing Item\_Weight values

miss\_bool = data['Outlet\_Size'].isnull()

#Impute data and check #missing values before and after imputation to confirm

print ('\nOrignal #missing: %d'% sum(miss\_bool))

data.loc[miss\_bool,'Outlet\_Size'] = data.loc[miss\_bool,'Outlet\_Type'].apply(lambda x: outlet\_size\_mode[x])

print (sum(data['Outlet\_Size'].isnull()))

output

Mode for each Outlet\_Type:

Outlet\_Type Grocery Store Supermarket Type1 Supermarket Type2 Supermarket Type3

Outlet\_Size Small Small Medium Medium

Orignal #missing: 4016

:0

## Feature Engineering

We explored some nuances in the data in the data exploration section. Lets move on to resolving them and making our data ready for analysis. We will also create some new variables using the existing ones in this section.

### Step 1: Consider combining Outlet\_Type

data.pivot\_table(values='Item\_Outlet\_Sales',index='Outlet\_Type')

### Step 2: Modify Item\_Visibility

#Determine average visibility of a product

visibility\_avg = data.pivot\_table(values='Item\_Visibility', index='Item\_Identifier')

#Impute 0 values with mean visibility of that product:

miss\_bool = (data['Item\_Visibility'] == 0)

print ('Number of 0 values initially: %d'%sum(miss\_bool))

#data.loc[miss\_bool,'Item\_Visibility'] = data.loc[miss\_bool,'Item\_Identifier'].apply(lambda x: visibility\_avg[x])

data.loc[miss\_bool,'Item\_Visibility'] = data.loc[miss\_bool,'Item\_Identifier'].apply(lambda x: visibility\_avg.at[x,'Item\_Visibility'])

print ('Number of 0 values after modification:%d'%sum(data['Item\_Visibility'] == 0))

output

Number of 0 values initially: 879

Number of 0 values after modification: 0

#Determine another variable with means ratio

data['Item\_Visibility\_MeanRatio'] = data.apply(lambda x: x['Item\_Visibility'], axis=1)

print (data['Item\_Visibility\_MeanRatio'].describe())

output

count 14204.000000

mean 0.069710

std 0.049728

min 0.003575

25% 0.031145

50% 0.057194

75% 0.096930

max 0.328391

Name: Item\_Visibility\_MeanRatio, dtype: float64

Thus the new variable has been successfully created. Again, this is just 1 example of how to create new features. I highly  encourage you to try more of these, as good features can drastically improve model performance and they invariably prove to be the difference between the best and the average model.

### Step 3: Create a broad category of Type of Item

Earlier we saw that the Item\_Type variable has 16 categories which might prove to be very useful in analysis. So its a good idea to combine them. One way could be to manually assign a new category to each. But there’s a catch here. If you look at the Item\_Identifier, i.e. the unique ID of each item, it starts with either FD, DR or NC. If you see the categories, these look like being Food, Drinks and Non-Consumables. So I’ve used the Item\_Identifier variable to create a new column:

#Get the first two characters of ID:

data['Item\_Type\_Combined'] = data['Item\_Identifier'].apply(lambda x: x[0:2])

#Rename them to more intuitive categories:

data['Item\_Type\_Combined'] = data['Item\_Type\_Combined'].map({'FD':'Food','NC':'Non-Consumable','DR':'Drinks'})

print(data['Item\_Type\_Combined'].value\_counts())

output

Food 10201

Non-Consumable 2686

Drinks 1317

Name: Item\_Type\_Combined, dtype: int64

#Years:

data['Outlet\_Years'] = 2013 - data['Outlet\_Establishment\_Year']

print(data['Outlet\_Years'].describe())

output

count 14204.000000

mean 15.169319

std 8.371664

min 4.000000

25% 9.000000

50% 14.000000

75% 26.000000

max 28.000000

Name: Outlet\_Years, dtype: float64

### Step 5: Modify categories of Item\_Fat\_Content

#Change categories of low fat:

print ('Original Categories:')

print (data['Item\_Fat\_Content'].value\_counts())

print ('\nModified Categories:')

data['Item\_Fat\_Content'] = data['Item\_Fat\_Content'].replace({'LF':'Low Fat','reg':'Regular','low fat':'Low Fat'})

print (data['Item\_Fat\_Content'].value\_counts())

output

Original Categories:

Low Fat 8485

Regular 4824

LF 522

reg 195

low fat 178

Name: Item\_Fat\_Content, dtype: int64

Modified Categories:

Low Fat 9185

Regular 5019

Name: Item\_Fat\_Content, dtype: int64

#Mark non-consumables as separate category in low\_fat:

data.loc[data['Item\_Type\_Combined']=="Non-Consumable",'Item\_Fat\_Content'] = "Non-Edible"

print(data['Item\_Fat\_Content'].value\_counts())

output

Low Fat 6499

Regular 5019

Non-Edible 2686

Name: Item\_Fat\_Content, dtype: int64

### Step 6: Numerical and One-Hot Coding of Categorical variables

Since scikit-learn accepts only numerical variables, I converted all categories of nominal variables into numeric types. Also, I wanted Outlet\_Identifier as a variable as well. So I created a new variable ‘Outlet’ same as Outlet\_Identifier and coded that. Outlet\_Identifier should remain as it is, because it will be required in the submission file.

#Import library:

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

#New variable for outlet

data['Outlet'] = le.fit\_transform(data['Outlet\_Identifier'])

var\_mod = ['Item\_Fat\_Content','Outlet\_Location\_Type','Outlet\_Size','Item\_Type\_Combined','Outlet\_Type','Outlet']

le = LabelEncoder()

for i in var\_mod:

data[i] = le.fit\_transform(data[i])

#One Hot Coding:

data = pd.get\_dummies(data, columns=['Item\_Fat\_Content','Outlet\_Location\_Type','Outlet\_Size','Outlet\_Type','Item\_Type\_Combined','Outlet'])

print(data.dtypes)

print(data[['Item\_Fat\_Content\_0','Item\_Fat\_Content\_1','Item\_Fat\_Content\_2']].head(10))

output

Item\_Identifier object

Item\_MRP float64

Item\_Outlet\_Sales float64

Item\_Type object

Item\_Visibility float64

Item\_Weight float64

Outlet\_Establishment\_Year int64

Outlet\_Identifier object

source object

Item\_Visibility\_MeanRatio float64

Outlet\_Years int64

Item\_Fat\_Content\_0 uint8

Item\_Fat\_Content\_1 uint8

Item\_Fat\_Content\_2 uint8

Outlet\_Location\_Type\_0 uint8

Outlet\_Location\_Type\_1 uint8

Outlet\_Location\_Type\_2 uint8

Outlet\_Size\_0 uint8

Outlet\_Size\_1 uint8

Outlet\_Size\_2 uint8

Outlet\_Type\_0 uint8

Outlet\_Type\_1 uint8

Outlet\_Type\_2 uint8

Outlet\_Type\_3 uint8

Item\_Type\_Combined\_0 uint8

Item\_Type\_Combined\_1 uint8

Item\_Type\_Combined\_2 uint8

Outlet\_0 uint8

Outlet\_1 uint8

Outlet\_2 uint8

Outlet\_3 uint8

Outlet\_4 uint8

Outlet\_5 uint8

Outlet\_6 uint8

Outlet\_7 uint8

Outlet\_8 uint8

Outlet\_9 uint8

dtype: object

Item\_Fat\_Content\_0 Item\_Fat\_Content\_1 Item\_Fat\_Content\_2

0 1 0 0

1 0 0 1

2 1 0 0

3 0 0 1

4 0 1 0

5 0 0 1

6 0 0 1

7 1 0 0

8 0 0 1

9 0 0 1

### Step 7: Exporting Data

Final step is to convert data back into train and test data sets. Its generally a good idea to export both of these as modified data sets so that they can be re-used for multiple sessions. This can be achieved using following code:

#Drop the columns which have been converted to different types:

data.drop(['Item\_Type','Outlet\_Establishment\_Year'],axis=1,inplace=True)

#Divide into test and train:

train = data.loc[data['source']=="train"]

test = data.loc[data['source']=="test"]

#Drop unnecessary columns:

test.drop(['Item\_Outlet\_Sales','source'],axis=1,inplace=True)

train.drop(['source'],axis=1,inplace=True)

#Export files as modified versions:

train.to\_csv("train\_modified.csv",index=False)

test.to\_csv("test\_modified.csv",index=False)

## 4. Model Building

Now that we have the data ready, its time to start making predictive models. I will take you through 6 models including linear regression, decision tree and random forest which can get you into Top 20 ranks in this competition (I mean ranks as of today because after reading this article, I’m sure many new leaders will emerge).

Lets start by making a **baseline model**. Baseline model is the one which requires no predictive model and its like an informed guess. For instance, in this case lets predict the sales as the overall average sales. This can be done as:

#Mean based:

mean\_sales = train['Item\_Outlet\_Sales'].mean()

#Define a dataframe with IDs for submission:

base1 = test[['Item\_Identifier','Outlet\_Identifier']]

base1['Item\_Outlet\_Sales'] = mean\_sales

#Export submission file

base1.to\_csv("alg0.csv",index=False)

Since I’ll be making many models, instead of repeating the codes again and again, I would like to define a **generic function** which takes the algorithm and data as input and **makes the model, performs cross-validation and generates submission**. If you don’t like functions, you can choose the longer way as well. But I have a tendency of using functions a lot (actually I over-use sometimes :D). So here is the function:

#Define target and ID columns:

target = 'Item\_Outlet\_Sales'

IDcol = ['Item\_Identifier','Outlet\_Identifier']

from sklearn import cross\_validation, metrics

def modelfit(alg, dtrain, dtest, predictors, target, IDcol, filename):

#Fit the algorithm on the data

alg.fit(dtrain[predictors], dtrain[target])

#Predict training set:

dtrain\_predictions = alg.predict(dtrain[predictors])

#Perform cross-validation:

cv\_score = cross\_validation.cross\_val\_score(alg, dtrain[predictors], dtrain[target], cv=20, scoring='mean\_squared\_error')

cv\_score = np.sqrt(np.abs(cv\_score))

#Print model report:

print "\nModel Report"

print "RMSE : %.4g" % np.sqrt(metrics.mean\_squared\_error(dtrain[target].values, dtrain\_predictions))

print "CV Score : Mean - %.4g | Std - %.4g | Min - %.4g | Max - %.4g" % (np.mean(cv\_score),np.std(cv\_score),np.min(cv\_score),np.max(cv\_score))

#Predict on testing data:

dtest[target] = alg.predict(dtest[predictors])

#Export submission file:

IDcol.append(target)

submission = pd.DataFrame({ x: dtest[x] for x in IDcol})

submission.to\_csv(filename, index=False)

### Linear Regression Model

from sklearn.linear\_model import LinearRegression, Ridge, Lasso

predictors = [x for x in train.columns if x not in [target]+IDcol]

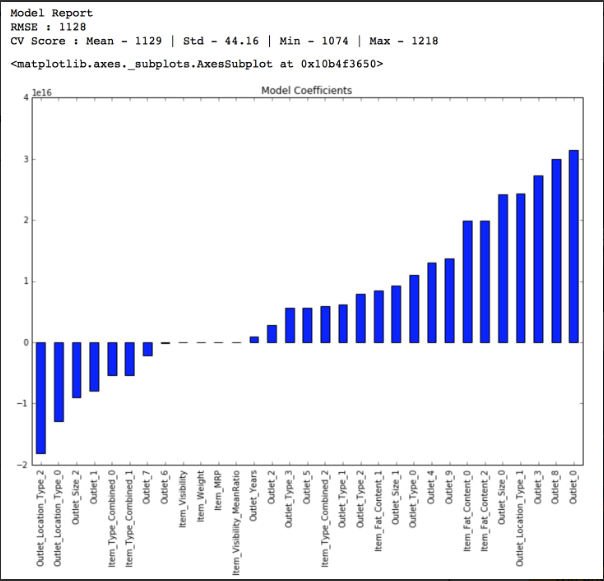
# print predictors

alg1 = LinearRegression(normalize=True)

modelfit(alg1, train, test, predictors, target, IDcol, 'alg1.csv')

coef1 = pd.Series(alg1.coef\_, predictors).sort\_values()

coef1.plot(kind='bar', title='Model Coefficients')



### Ridge Regression Model:

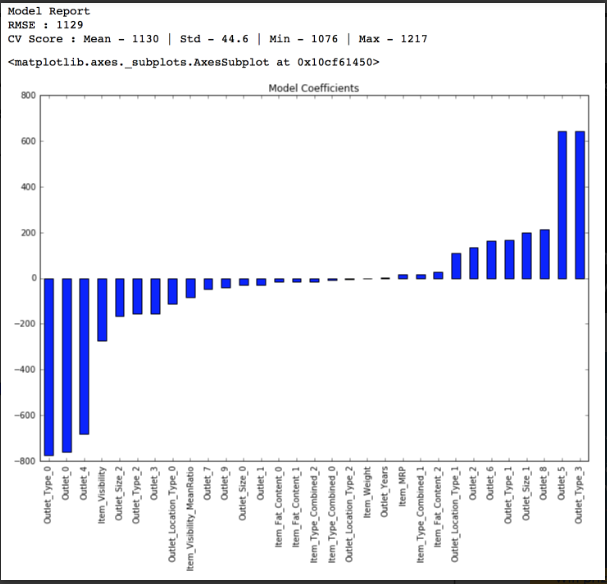
predictors = [x for x in train.columns if x not in [target]+IDcol]

alg2 = Ridge(alpha=0.05,normalize=True)

modelfit(alg2, train, test, predictors, target, IDcol, 'alg2.csv')

coef2 = pd.Series(alg2.coef\_, predictors).sort\_values()

coef2.plot(kind='bar', title='Model Coefficients')



### Decision Tree Model

from sklearn.tree import DecisionTreeRegressor

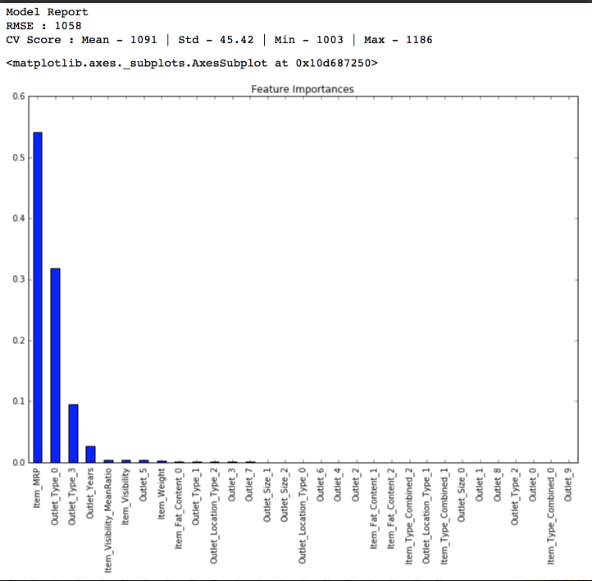
predictors = [x for x in train.columns if x not in [target]+IDcol]

alg3 = DecisionTreeRegressor(max\_depth=15, min\_samples\_leaf=100)

modelfit(alg3, train, test, predictors, target, IDcol, 'alg3.csv')

coef3 = pd.Series(alg3.feature\_importances\_, predictors).sort\_values(ascending=False)

coef3.plot(kind='bar', title='Feature Importances')



Here you can see that the RMSE is 1058 and the mean CV error is 1091. This tells us that the model is slightly overfitting. Lets try making a decision tree with just top 4 variables, a max\_depth of 8 and min\_samples\_leaf as 150.

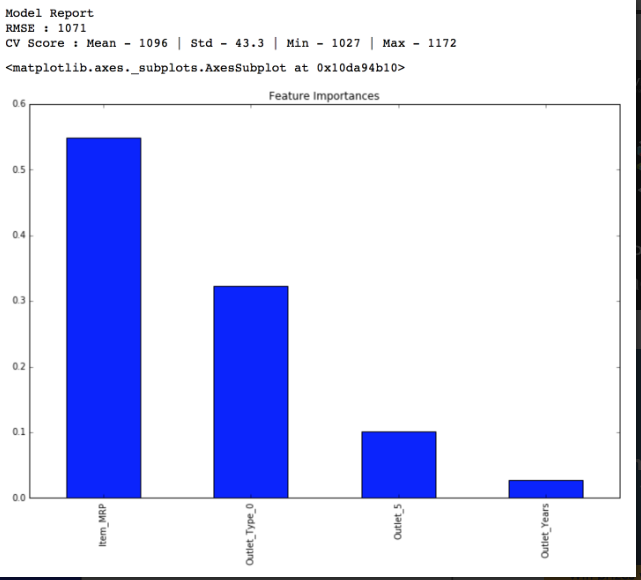
predictors = ['Item\_MRP','Outlet\_Type\_0','Outlet\_5','Outlet\_Years']

alg4 = DecisionTreeRegressor(max\_depth=8, min\_samples\_leaf=150)

modelfit(alg4, train, test, predictors, target, IDcol, 'alg4.csv')

coef4 = pd.Series(alg4.feature\_importances\_, predictors).sort\_values(ascending=False)

coef4.plot(kind='bar', title='Feature Importances')



### Random Forest Model

from sklearn.ensemble import RandomForestRegressor

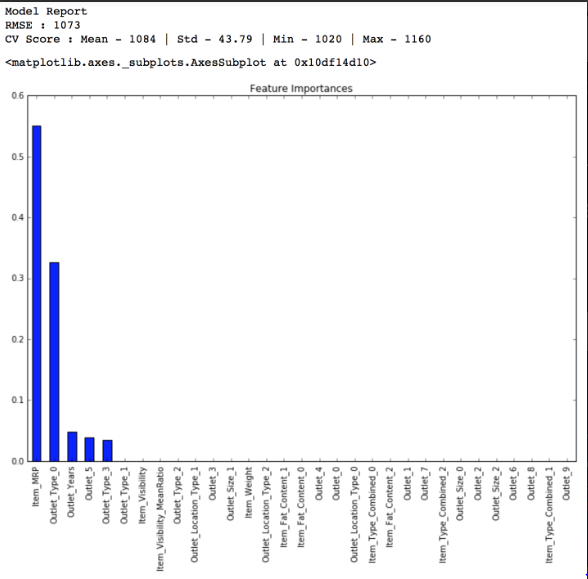
predictors = [x for x in train.columns if x not in [target]+IDcol]

alg5 = RandomForestRegressor(n\_estimators=200,max\_depth=5, min\_samples\_leaf=100,n\_jobs=4)

modelfit(alg5, train, test, predictors, target, IDcol, 'alg5.csv')

coef5 = pd.Series(alg5.feature\_importances\_, predictors).sort\_values(ascending=False)

coef5.plot(kind='bar', title='Feature Importances')



You might feel this is a very small improvement but as our model gets better, achieving even minute improvements becomes exponentially difficult. Lets try another random forest with max\_depth of 6 and 400 trees. Increasing the number of trees makes the model robust but is computationally expensive.

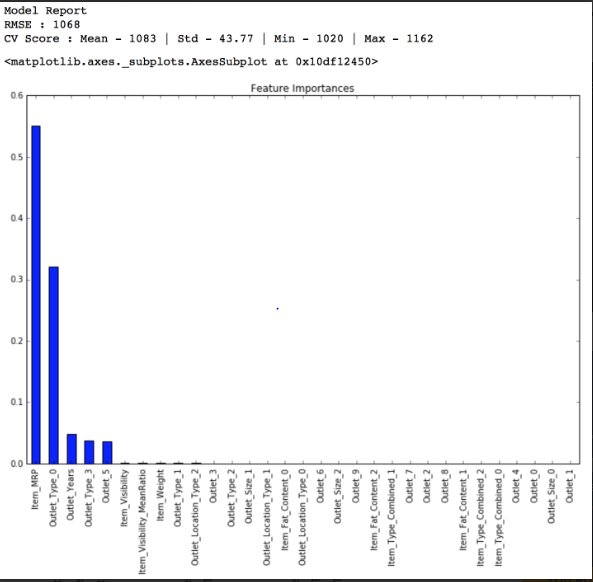
predictors = [x for x in train.columns if x not in [target]+IDcol]

alg6 = RandomForestRegressor(n\_estimators=400,max\_depth=6, min\_samples\_leaf=100,n\_jobs=4)

modelfit(alg6, train, test, predictors, target, IDcol, 'alg6.csv')

coef6 = pd.Series(alg6.feature\_importances\_, predictors).sort\_values(ascending=False)

coef6.plot(kind='bar', title='Feature Importances')



Result of analysis

