Project Report on

SUPPLIER AND VENDOR EVALUATION ON E-COMMERCE WEBSITES AND PRODUCT RECOMMENDATION

at

Fusion Informatics Limited

External Guide :
Mr. Dhaval Shah

Internal Guide :
Prof. Ketan Sarvakar

Prepared By:
Mr. Rahul Bhenjalia (15012011039)

B.Tech Semester VIII
(Computer Engineering)
April 2019

Submitted to,
Department of Computer Engineering
U.V. Patel College of Engineering
Ganpat University, Kherva - 384 012

U.V.PATEL COLLEGE OF ENGINEERING

GANPAT UNIVERSITY
|| विद्यया समाजोत्कर्षः ||

U V PATEL COLLEGE OF ENGINEERING

28/04/2019

CERTIFICATE

TO WHOM SO EVER IT MAY CONCERN

This is to certify that **Mr. Rahul Bhenjalia** student of **B.Tech. Semester VIII (Computer Engineering)** has completed his full semester on site project work titled “**Supplier and Vendor Evaluation On E-Commerce Websites and Product Recommendation**” satisfactorily in partial fulfillment of the requirement of Bachelor of Technology degree of Computer Engineering of Ganpat University, Kherva, Mehsana in the year 2018-2019.

College Project Guide

Sign

Dr. Paresh M. Solanki,
Head, Computer Engineering

Prof. Ketan Sarvakar

FUSION
INFORMATICS

501, New York Plaza, Opp. Judges Bunglow,
Premchandnagar Road, Bodakdev,
Ahmedabad-380 054.
Phone : +91 - 79 - 3012 2203
Website : www.fusioninformatics.com
E-mail : inquiry@fusioninformatics.com

Certificate

This is to certify that **<u>Mr. Rahul Bhenjalia</u>** has successfully completed his internship on "***Data Science***" at **"Fusion Informatics Limited"** From 7th Jan, 2019 to 26th April, 2019 towards fulfillment of his internship.

FUSION INFORMATICS PVT. LTD.

Dhaval Shah

CTO, Fusion Informatics Limited

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

*I owe my deep gratitude to **my project guide Professor Ketan Sarvakar** who took keen interest on my project work and guided me all along, till the completion of my project work by providing all the necessary information for developing a good system.*

*I respect and thank **Mr. Dhaval Shah, my company project guide**, for providing me an opportunity to do the project work in **Fusion Informatics Limited** and giving us all support and guidance which made me complete the project duly. I am extremely thankful to him for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.*

*I, feel elated in thanking **Dr. Paresh Solanki, Head of Department of Computer Engineering, U.V Patel College of Engineering,** for his valuable suggestion and skilled supervision to carry out this project.*

*I am thankful to and fortunate enough to get constant encouragement, support and guidance from all **Teaching staffs of Computer Engineering Department** which helped us in successfully completing our project work. Also, I would like to extend our sincere esteems to all staff in laboratory for their timely support.*

Last but not the least, I would like to thank my parents, all my family members and friends, who, in spite their own difficulties, have stood as a constant source of' inspiration in eve stage and shown moral support to carry out this research work successfuIIy.

ABSTRACT

Supplier and vendor evaluation framework can help to set up a benchmark and corrective action plan for the existing supplier. Company can decide to reward supplier based on their excellence performance and penalizing or de-listing them if the performance is not in standard.

Supplier Evaluation and Management had been practiced in manufacturing industry. The most important criteria in construction industry is material quality, delivery dependability, and cost. The most important factor of supplier selection should be the quality level of the procurement items.

Businesses are running customer services since a long time. The traditional approach of customer service is to contact customers through emails, posts and telephones and ask them to provide feedback on company's products and their services.

These days' companies' especially online businesses, have ratings and reviews section on their websites for their products. But, it is not easy to read each and every given review online manually. Not just this, but sometimes it becomes difficult to make sense of those reviews also, e.g. the reviews containing incorrect spellings or shorthand words etc. This is where Data Science comes into picture.

Online shopping businesses like Amazon relies highly on their vendor and supplier network to keep morale of the customer satisfaction and hence it is quintessential that they maintain detail system to keep it in check.

INDEX

LIST OF FIGURES

1. INTRODUCTION

1.1 Data Science on E-Commerce

The data is increasing with every single click on the internet. In order to make sense of this huge data and use it for company's benefits etc., we need help from different Data Science techniques.

Every single day people buy and sell things online, with a single mouse click, but in order to keep the customers engaged with the website or to improve customer's experience, companies use Data Science/Machine Learning, i.e. on amazon website, when you are looking for a product, you see number recommendations. These recommendations generate through Machine Learning algorithms. It learns from users past activities and purchases.

The companies store the data of every click customer make, every reviews customer read, every story customer share on social media etc., and use this data to learn about their customer or create a platform to help new customers.

"A recommendation system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item."

The recommendation system is more than what above definition describes. It is used to filter choices for particular user on the basis of their past searches or other customer's search or purchase data. It gives users a personalized view on the e-commerce website and help them to select relevant products. E.g. - While looking for a new phone on Amazon website, there is a possibility that you might want to buy a phone cover too. Amazon will decide this possibility by analysing previous purchases or searches data of their customers.

There are a number of ways to setup a recommendation system. Each of these techniques filter or provide recommendation in different manner. There are three main and known techniques as below-

- Collaborative filtering (Implemented in the project)
- Content Based Filtering
- Hybrid Recommendation Filtering

Businesses are running customer services since a long time. The traditional approach of customer service is to contact customers through emails, posts and telephones and ask them to provide feedback on company's products and their services.

These days' companies' especially online businesses, have ratings and reviews section on their websites for their products. But, it is not easy to read each and every given review online manually. Not just this, but sometimes it becomes difficult to make sense of those reviews also, e.g. the reviews containing incorrect spellings or shorthand words etc. This is where Data Science comes into picture.

Online shopping businesses like Amazon relies highly on their vendor and supplier network to keep morale of the customer satisfaction and hence it is quintessential that they maintain detail system to keep it in check.

1.2 Data Science in B2B era

People often think that the main aim of Data Science is to churn out predictive analytical solutions to anticipate and support company decision-making. But it's automation that's the real added value that Data Science's bringing to the B2B arena. Machine-learning algorithms learn from experience.

They can mimic how we prioritize jobs and calculate our reasoning. Their new role is to relieve us of the repetitive manual work they can do so much more reliably, quickly and efficiently.

An example is the analysis of retention rates. Let's say modern algorithmic processing could calculate a company's customer retention rates. It's not so much the predictive value of probability that's so useful but the actionable intelligence we get.

It's the knowledge of which customers we need to target at which moment to ensure their retention. A Data Scientist team can so easily identify and then automate this kind of time-consuming task.

When it's a case of mimetic learning (from imitation), the machine does need guidance. This is where the Data Scientist comes in. They have to understand what the company employee's needs. And then merge these practical end-goals into the machine's replication process.

Looking ahead, B2B Data Scientists will be called upon to prove via A/B testing that machine-driven tasks are effective. Once they've devised an algorithm, they'll have to compare actual machine and human performance.

This is the only way to benchmark the performance of robots and prove what they can achieve. Then, we'll be in a strong position to convince the sceptics out there. And show them just how much benefit robots can bring to our working lives.

Data Scientists help end-users get results. To achieve this, a three-step logical approach is required. The first is understanding the algorithm. The second is the deployment of computer science. The third is determining the right method to realize the end goal.

This makes it a role far beyond that of software developer. Data Scientists have to fathom the practical issues that drive functionality. It's here that the real value of a Data Science team lies. The added business value they deliver is the result of this synergy of technology and practical functionality.

1.3 Supplier and Vendor Evaluation

Supplier evaluation is the process to access new or existing supplier base on their delivery, price, production, and quality of management, technical and services. A standard supplier evaluation framework shall be used in all cases for the existing and potential suppliers

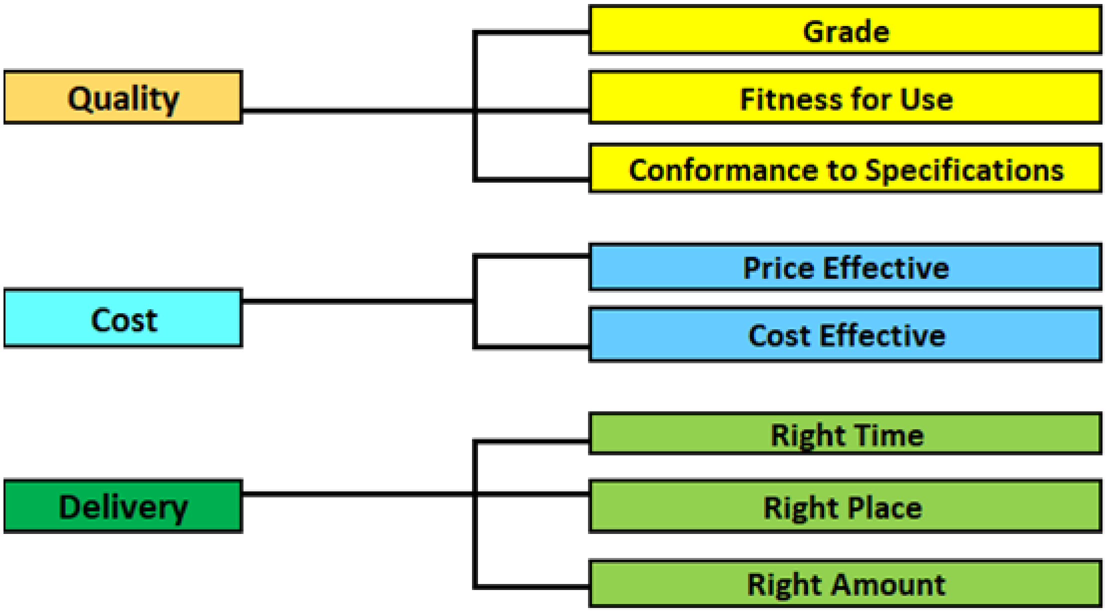


Figure: 1.1 (a) Vendor Evaluation Criteria

The supplier evaluation framework can help to set up a benchmark and corrective action plan for the existing supplier. Company can decide to reward supplier based on their excellence performance and penalizing or de-listing them if the performance is not in standard. Supplier Evaluation and Management had been practiced in manufacturing industry.

The most important criteria in construction industry is material quality, delivery dependability, and cost. The most important factor of supplier selection should be the quality level of the procurement items.

Product quality should consistently meet specified requirements since it can directly affect the quality of the finished goods. Not only product quality reliability, supplier characteristics like delivery lead time shall be consider carefully.

Unit price should not be the only criteria in supplier evaluation. Total cost of ownership is an important factor.

Total cost of ownership includes the unit price of the material, payment terms, cash discount, ordering and carrying cost, logistics and maintenance costs, and other more qualitative costs that may not be easy to assess.

Supplier must value add their product by providing good services when needed. For example, when product information or warranty service is needed, suppliers must respond on a timely basis. Selecting services and products from suppliers with excellent delivery ability can reduce or get rid of waste related with purchasing raw materials such as inventory, storage cost, and cost related with multiple times of material transferring.

Many company adopt to "Just-in-Time" (JIT) Inventory process to reduce the cost of ‘’waste". Supplier need to make the delivery on-time based on company request. Supplier that perform excellent delivery ability can provide additional value to the company by reduce the risks of material running out, saving on unnecessary transportation costs, reduce the need to storage and cost inventory related cost.

2. FEASIBILITY ANALYSIS

2.1 Technical Feasibility

The technical phase divided among three categories – the basic code, file compilation and library implementation.

The basic code requires a "scratchpad" like scenario where bits and pieces of the algorithm can be worked out and even minute changes can be made on the basic level if need be. (JupyterLab)
An intermediate-type platform that enables run Python Networking and ScaPy scripts with one-go routine and able to deal with multiple files to generate library and provide client-server services.
Basically any python platform able to execute "import" command to use the above generated library

2.2 Time Schedule Feasibility

There are two parts for which same approach is to be used to process. The scraped dataset and the company given dataset.

The scraped dataset requires a significant amount of time and depends on Network speed, the web browser and amount of pings you receive in your connection. After the scraper has done its job (1-2 weeks in this case), it depends on the processor configuration to compile that data and produce a comma separated file for viewing process(seemingly 2-3 hours in this case).

Time limit doesn't apply to creating algorithms and running model, it is always an undying progression since every model has room for perfection with more data to be compiled through. Although, to generate a commendable algorithm around 1-2 weeks of time would be considered as sufficient.

Moreover, the whole schedule is to be repeated for the company provided dataset and apart from the scraping part, it took lesser time to preprocess it and modify the current model to its standards. Hence, the given time schedule could be considered to create a sustainable algorithm but there will be always room for improvement.

2.3 Operational Feasibility

Since, the process tends to work in background there is very little but quintessential output for the customers for the organization. The end result is statistics on the vendors and sellers which is of utmost importance to organization rather than its customers. As mentioned before, it is necessary for organizations in B2B industry to maintain customer-seller harmony.

2.4 Implementation Feasibility

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps in building machine learning models more accurately. In machine learning, there is an 80/20 rule. Every data scientist should spend 80% time for data pre-processing and 20% time to actually perform the analysis. Hence, implementation rests upon the noise and aberration data, how those misfits are treated define what kind of model it is going to produce.

Model selection in the context of machine learning can have different meanings, corresponding to different levels of abstraction.

For one thing, we might be interested in selecting the best hyper parameters for a selected machine learning method. Hyper parameters are the parameters of the learning method itself which we have to specify a priori, i.e., before model fitting. In contrast, model parameters are parameters which arise as a result of the fit [1]. In a logistic regression model, for example, the regularization strength (as well as the regularization type, if any) is a hyper parameter which has to be specified prior to the fitting, while the coefficients of the fitted model are model parameters. Finding the right hyper parameters for a model can be crucial for the model performance on given data.

For another thing, we might want to select the best learning method (and their corresponding "optimal" hyper parameters) from a set of eligible machine learning methods. In the following, we will refer to this as algorithm selection. With a classification problem at hand, we might wonder, for instance, whether a logistic regression model or a random forest classifier yields the best classification performance on the given task.

2.5 Economic Feasibility

Data is the source of economic potential, and the Data Lake is a source of latent value. However, data, like oil, needs refining in order to turn it into something of value, and that's the role of Data Science.

Data Science requires a highly iterative, rapid exploration, rapid testing environment that supports a fail fast/learn, continuous learning development approach that seeks to discover the drivers for success.

Economics, when coupled with Data Science and Design Thinking, provides the frame – the connective tissue – against which to focus financial, technology and human investments in order to create new sources of wealth (value).

3. SOFTWARE AND HARDWARE REQUIREMENT

3.1 Hardware Requirements

Anaconda (For Spyder, JupyterLab, GlueViz):

- Physical server or virtual machine.
- CPU: 2 x 64-bit, 2.8 GHz, 8.00 GT/s CPUs or better. Verify machine architecture.
- Memory: minimum RAM size of 32 GB, or 16 GB RAM with 1600 MHz DDR3 installed, for a typical installation with 50 regular users. Verify memory requirements.
- Storage: Recommended minimum of 100 GB, or 300 GB if you are planning to mirror both Anaconda Repository, which is approximately 90 GB, and the PyPI repository, which is approximately 100 GB, or at least 1 TB for an air gapped environment. Additional space is recommended if Repository is used to store packages built by your organization. Verify storage requirements.
- Internet access to download the files from Anaconda Cloud, or a USB drive containing all of the files you need with alternate instructions for air gapped installations.

3.2 Software Requirements. (Python, Anaconda, JupyterLab)

Software Used	*Description*
Operating System	We have chosen Windows operating system for its best support and user-friendliness.
Python	Open source, Lots of libraries, fast, easy to understand.
JupyterLab and JupyTer Notebook	Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference too many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, ReStructuredText, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" command line interface in a shell.

To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

Anaconda-Spyder (IDE) and

User level install of the version of python you want

Able to install/update packages completely independent of system libraries or admin privileges

Conda tool installs binary packages, rather than requiring compile resources like pip - again, handy if you have limited privileges for installing necessary libraries.

More or less eliminates the headaches of trying to figure out which version/release of package X is compatible with which version/release of package Y, both of which are required for the install of package Z

Comes either in full-meal-deal version, with numpy, scipy, PyQt, spyder IDE, etc. or in minimal / alacarte version (miniconda) where you can install what you want, when you need it

No risk of messing up required system libraries

Google Chrome

A browser that supports CGI, HTML & JavaScript.
The WebScraper extension requires Chrome 49+. There are no OS limitations.

4. PROJECT PLAN

Data science projects do not have a nice clean lifecycle with well-defined steps like software development lifecycle (SDLC). Usually, data science projects tramp into delivery delays with repeated hold-ups, as some of the steps in the lifecycle of a data science project are non-linear, highly iterative and cyclical between the data science team and various others teams in an organization. It is very difficult for the data scientists to determine in the beginning which is the best way to proceed further. Although the data science workflow process might not be clean, data scientists ought to follow a certain standard workflow to achieve the output.

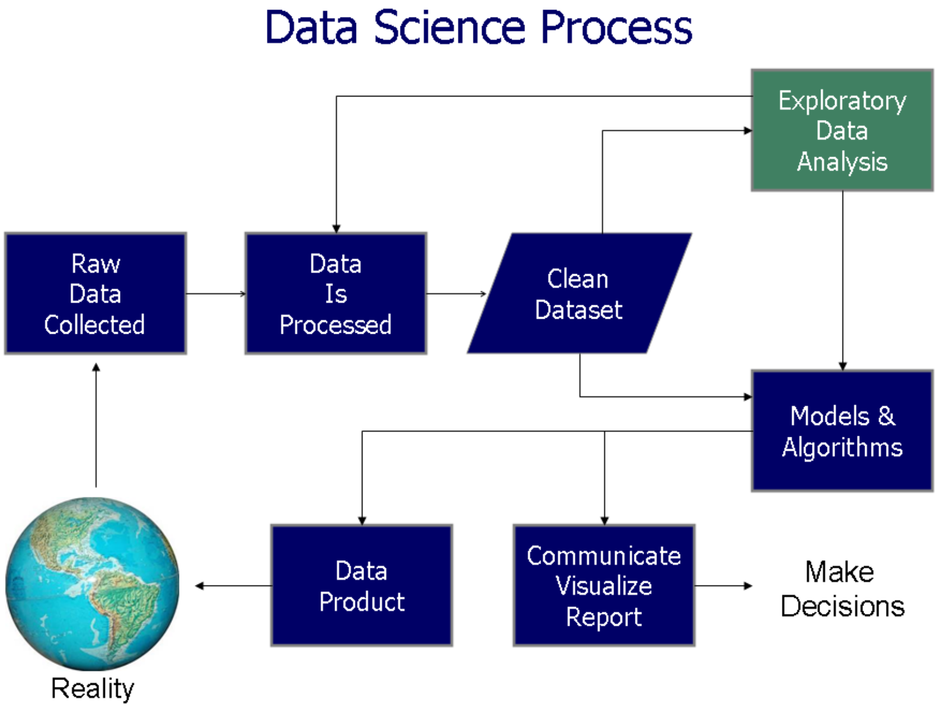


Figure 4 (a) Project Plan for a Data Science project

Data science project lifecycle is similar to the CRISP-DM lifecycle that defines the following standard 6 steps for data mining projects-

- Business Understanding
- Data Understanding
- Data Preparation
- Modelling
- Evaluation
- Deployment

Task	*Time*
Research & Development:	1 week
Brush up on previous ML & DM topics and performing various ML algorithms on small datasets	
Sample Project:	1-2 days
To Study Implementation of Markdown Prices	
Outlining project and preparing hypotheses	1-2 days
Creating sitemaps and crawling for data Updating sitemaps and recoding them to check up on data	1-2 weeks
Data Preprocessing:	1-2 weeks(per category of dataset)
Filling of empty cells, Feature Engineering to check importance of every attribute, Label Encoding,	
Creating graphs on target value to decide which ML models can fit this data.	
Running Exploratory Data Analysis on data. Generating various analytics graphs and drawing conclusions.	2-3 weeks
Determining strategy, approach and needed result.	2-3 days

Build multiple ML models to check the best fit among all. Analyze the accuracy and what attribute makes what difference, which factors may change and which may not the prediction accuracy of particular model. Create report for particular dataset	3-4 weeks
Generating various post-analytics report for the evaluation of the created models.	1 week
Evaluating all algorithms and preparing results from the best fits. Generating evaluations reports for the organization.	1 week

5. DATA COLLECTION

In order to create a model for the given definition, it was necessary that all hypothesis be checked out on temporary and similar characteristics datasets.

The algorithm required the dataset to of following general schema:

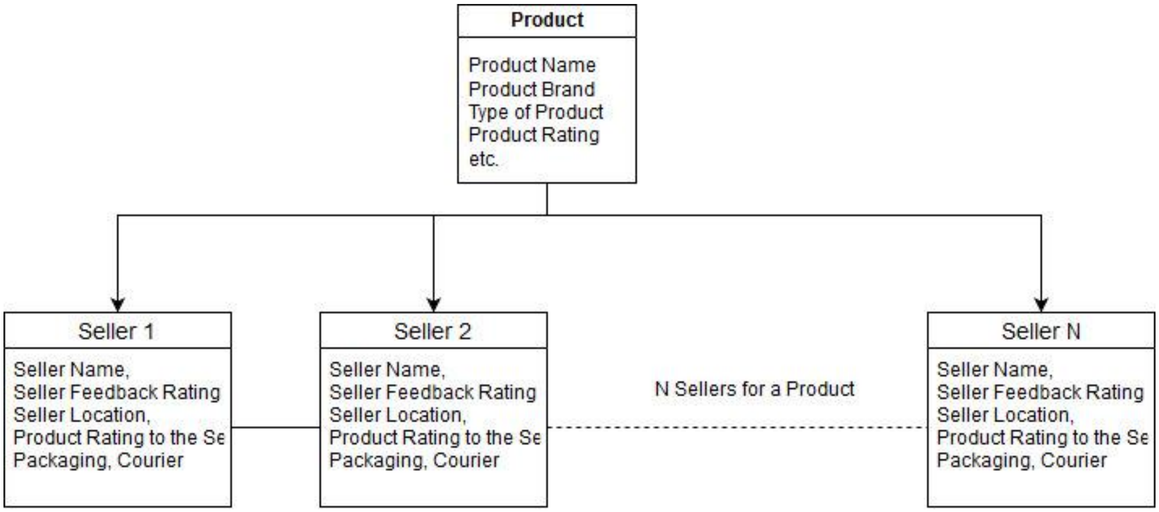


Figure 5.1(a) General Defined Schema

The closest resemblance of schema requested by the guide can be imposed on any e-commerce website.

Since, Amazon.in has one of the largest collection of product and seller database, it was considered that any arbitrary model and hypothesis could be concluded from it.

5.1 Web Scraping for Amazon dataset

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. Web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol, or through a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. It is a form of copying, in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.

5.1.1 About the Web scraper extension

Web Scraper is an extension for chrome browser made exclusively for web data scraping. You can setup a plan (sitemap) on how to navigate a website and specify the data to be extracted. The scraper will traverse the website according to the setup and extract the relevant data. It lets you export the extracted data to CSV. Multiple pages can be scraped using the tool making it all the more powerful. It can even extract data from dynamic pages that use JavaScript and Ajax.

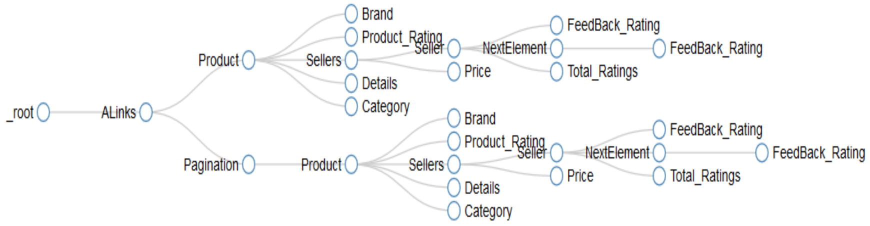
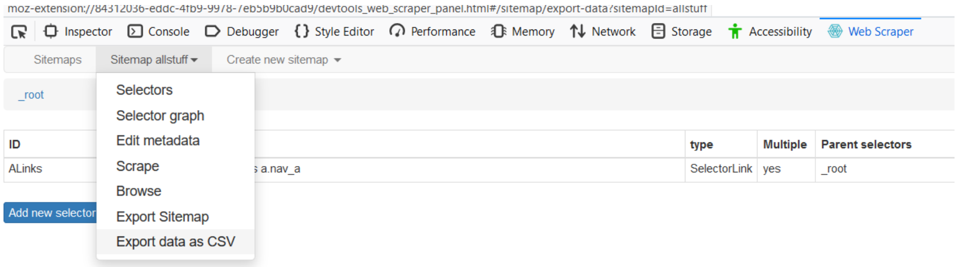


Figure 5.1.1 (b) Selector Graph for Web Scraping

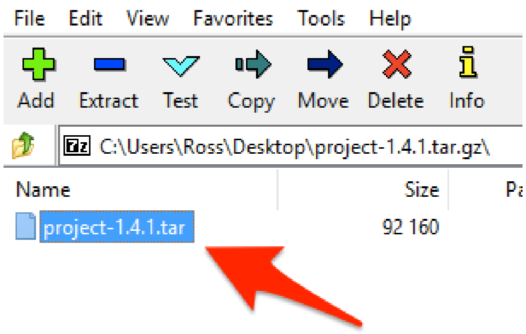

moz-extension://84312036-eddc-4fb9-9978-7eb5b9b0cad9/devtools_web_scraper_panel.html#/sitemap/export-data?sitemapId=allstuff

ID		type	Multiple	Parent selectors
ALinks	s a.nav_a	SelectorLink	yes	_root

Figure 5.1.2 (b) WebScraper Extension

5.2 Data Collection from Organization

The organization data was provided with tar.gz extensions. “.tar” is common archive format used on Unix-like systems. Generally used in conjunction with compressors such as gzip, bzip2, compress or xz to create .tar.gz, .tar.bz2, .tar.Z or tar.xz files.



C:\Users\Ross\Desktop\project-1.4.1.tar.gz\

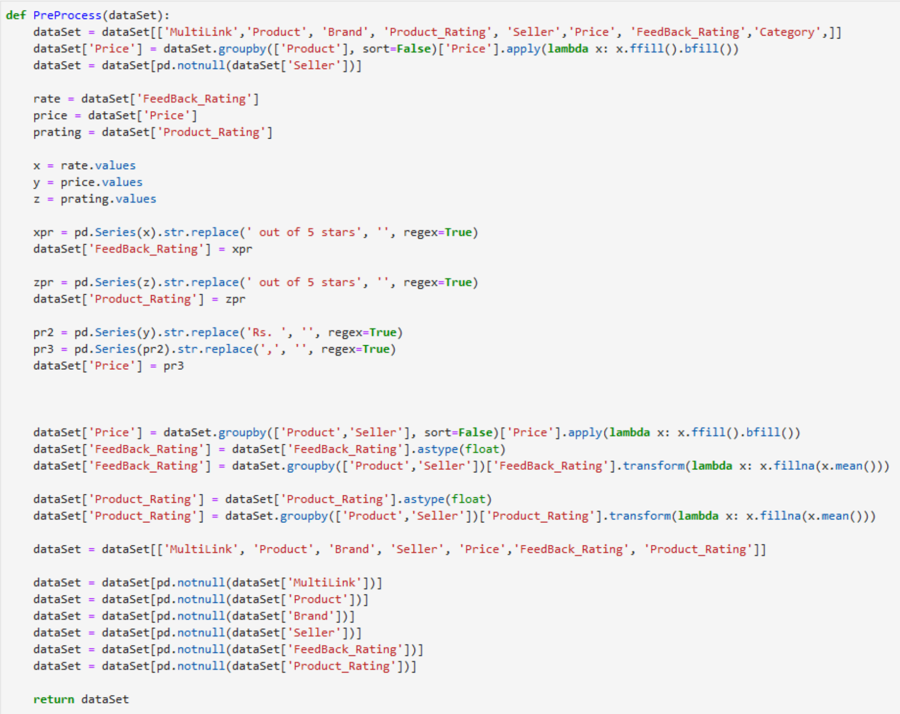
project-1.4.1.tar 92 160

Figure 5.2 (a) tar.gz Extension example

GNU Zip, the primary compression format used by Unix-like systems. The compression algorithm is DEFLATE. The compressed file, through convertors, would provide with a comma or tab separated files as datasets which would be used in further process

6. DATA PRE-PROCESSING

Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model.



```
def PreProcess(dataSet):
    dataSet = dataSet[['MultiLink','Product', 'Brand', 'Product_Rating', 'Seller','Price', 'FeedBack_Rating','Category',]]
    dataSet['Price'] = dataSet.groupby(['Product'], sort=False)['Price'].apply(lambda x: x.ffill().bfill())
    dataSet = dataSet[pd.notnull(dataSet['Seller'])]

    rate = dataSet['FeedBack_Rating']
    price = dataSet['Price']
    prating = dataSet['Product_Rating']

    x = rate.values
    y = price.values
    z = prating.values

    xpr = pd.Series(x).str.replace(' out of 5 stars', '', regex=True)
    dataSet['FeedBack_Rating'] = xpr

    zpr = pd.Series(z).str.replace(' out of 5 stars', '', regex=True)
    dataSet['Product_Rating'] = zpr

    pr2 = pd.Series(y).str.replace('Rs. ', '', regex=True)
    pr3 = pd.Series(pr2).str.replace(',', '', regex=True)
    dataSet['Price'] = pr3



    dataSet['Price'] = dataSet.groupby(['Product','Seller'], sort=False)['Price'].apply(lambda x: x.ffill().bfill())
    dataSet['FeedBack_Rating'] = dataSet['FeedBack_Rating'].astype(float)
    dataSet['FeedBack_Rating'] = dataSet.groupby(['Product','Seller'])['FeedBack_Rating'].transform(lambda x: x.fillna(x.mean()))

    dataSet['Product_Rating'] = dataSet['Product_Rating'].astype(float)
    dataSet['Product_Rating'] = dataSet.groupby(['Product','Seller'])['Product_Rating'].transform(lambda x: x.fillna(x.mean()))

    dataSet = dataSet[['MultiLink', 'Product', 'Brand', 'Seller', 'Price','FeedBack_Rating', 'Product_Rating']]

    dataSet = dataSet[pd.notnull(dataSet['MultiLink'])]
    dataSet = dataSet[pd.notnull(dataSet['Product'])]
    dataSet = dataSet[pd.notnull(dataSet['Brand'])]
    dataSet = dataSet[pd.notnull(dataSet['Seller'])]
    dataSet = dataSet[pd.notnull(dataSet['FeedBack_Rating'])]
    dataSet = dataSet[pd.notnull(dataSet['Product_Rating'])]

    return dataSet
```

Figure 6 (a) Pre-Processor Function for Amazon Dataset

The webscraper gives data in following format:

Sitemap of the scraper (JSON format):

{"_id":"allstuff","startUrl":["https://www.amazon.in/gp/site-directory?ref_=nav_shopall_btn"],"selectors":[{"id":"ALinks","type":"SelectorLink","parentSelectors":["_root"],"selector":"ul.nav_cat_links a.nav_a","multiple":true,"delay":0},{"id":"Product","type":"SelectorLink","parentSelectors":["ALinks","Pagination"],"selector":"a.a-link-normal.s-access-detail-page","multiple":true,"delay":0},{"id":"Pagination","type":"SelectorLink","parentSelectors":["ALinks"],"selector":"a.pagnNext","multiple":false,"delay":0},{"id":"Brand","type":"SelectorText","parentSelectors":["Product"],"selector":"div.centerColAlign div.a-section.a-spacing-none a.a-link-normal","multiple":false,"regex":"","delay":0},{"id":"Product_Rating","type":"SelectorText","parentSelectors":["Product"],"selector":"span.reviewCountTextLinkedHistogram a.a-popover-trigger","multiple":true,"regex":"","delay":0},{"id":"Sellers","type":"Selector

```
Link","parentSelectors":["Product"],"selector":"span.olp-padding-right
a","multiple":false,"delay":0},{"id":"Seller","type":"SelectorLink","parentSele
ctors":["Sellers"],"selector":"span.a-size-medium
a","multiple":true,"delay":0},{"id":"Price","type":"SelectorText","parentSelect
ors":["Sellers"],"selector":"span.a-size-large >
span","multiple":true,"regex":"","delay":0},{"id":"FeedBack_Rating","type":"Sel
ectorText","parentSelectors":["Seller","NextElement"],"selector":"tr.feedback-
row:nth-of-type(1) span.a-icon-alt, th.a-nowrap div.a-
section","multiple":true,"regex":"","delay":0},{"id":"NextElement","type":"Sele
ctorElementClick","parentSelectors":["Seller"],"selector":"a#feedback-next-
link.a-link-
normal","multiple":false,"delay":0,"clickElementSelector":"a#feedback-next-
link.a-link-normal","clickType":"clickMore","discardInitialElements":"do-not-
discard","clickElementUniquenessType":"uniqueText"},{"id":"Details","type":"Sel
ectorTable","parentSelectors":["Product"],"selector":"div.column.col1
table","multiple":true,"columns":[{"header":"OS","name":"OS","extract":true},{"
header":"Android","name":"Android","extract":true}],"delay":0,"tableDataRowSele
ctor":"tr:nth-of-type(n+2)","tableHeaderRowSelector":"tr:nth-of-
type(1)"},{"id":"Category","type":"SelectorText","parentSelectors":["Product"],
"selector":"div.a-section.a-padding-
medium","multiple":false,"regex":"","delay":0},{"id":"Total_Ratings","type":"Se
lectorText","parentSelectors":["Seller"],"selector":"tr:nth-of-type(5) td.a-
text-right:nth-of-type(5) span","multiple":false,"regex":"","delay":0}]}
```

Hence, processing that raw data through the given process function in Figure 6 (a) would finally yield in the final dataset with no missing values or aberrations in Figure 6(c).

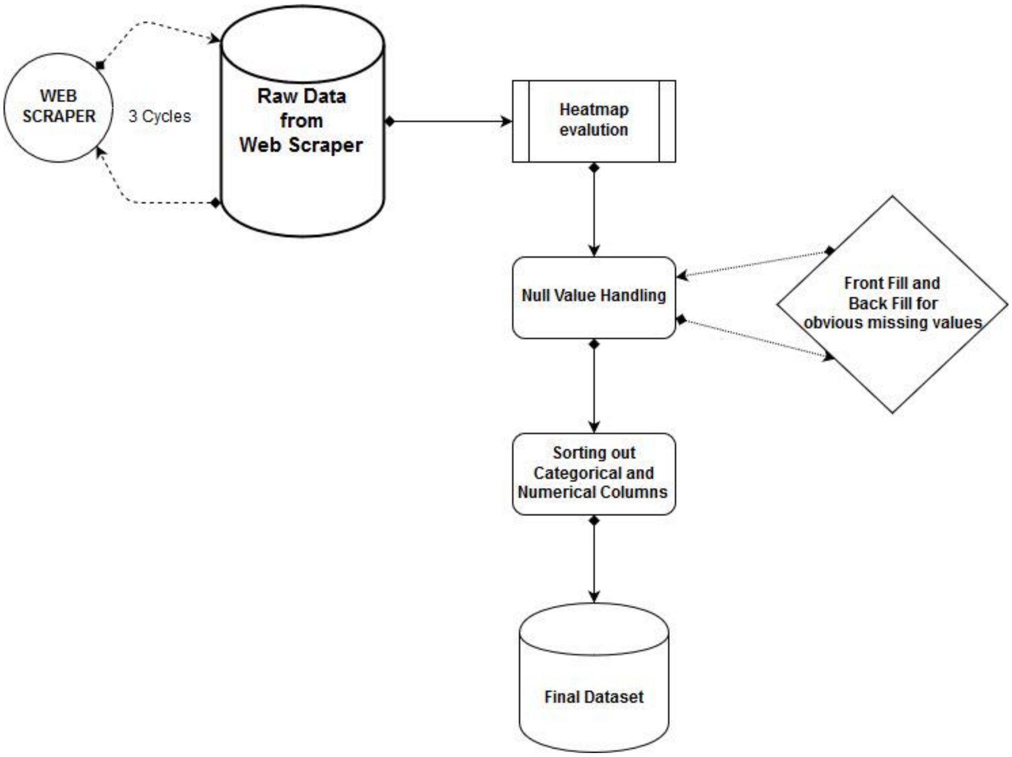
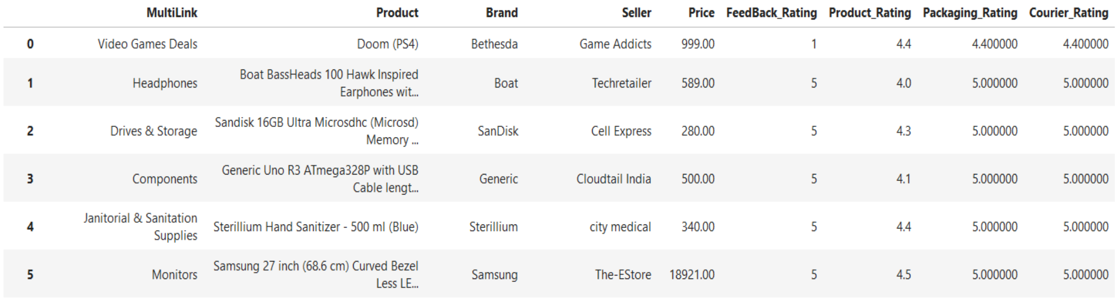

WEB SCRAPER
3 Cycles
Raw Data from Web Scraper
Heatmap evalution
Null Value Handling
Front Fill and Back Fill for obvious missing values
Sorting out Categorical and Numerical Columns
Final Dataset

Figure 6 (b) Pre-Processing Flow



	MultiLink	Product	Brand	Seller	Price	FeedBack_Rating	Product_Rating	Packaging_Rating	Courier_Rating
0	Video Games Deals	Doom (PS4)	Bethesda	Game Addicts	999.00	1	4.4	4.400000	4.400000
1	Headphones	Boat BassHeads 100 Hawk Inspired Earphones wit...	Boat	Techretailer	589.00	5	4.0	5.000000	5.000000
2	Drives & Storage	Sandisk 16GB Ultra Microsdhc (Microsd) Memory ...	SanDisk	Cell Express	280.00	5	4.3	5.000000	5.000000
3	Components	Generic Uno R3 ATmega328P with USB Cable lengt...	Generic	Cloudtail India	500.00	5	4.1	5.000000	5.000000
4	Janitorial & Sanitation Supplies	Sterillium Hand Sanitizer - 500 ml (Blue)	Sterillium	city medical	340.00	5	4.4	5.000000	5.000000
5	Monitors	Samsung 27 inch (68.6 cm) Curved Bezel Less LE...	Samsung	The-EStore	18921.00	5	4.5	5.000000	5.000000

Figure 6 (c) FinalData.csv or Amazon.csv

Most techniques in data mining rely on a data set that is supposedly complete or noise-free. However, real-world data is far from being clean or complete. In data preprocessing it is common to employ techniques to either removing the noisy data or to impute (fill in) the missing data. The following two sections are devoted two missing values imputation and noise filtering.

Missing values have been reported to cause loss of efficiency in the knowledge extraction process, strong biases if the missingness introduction mechanism is mishandled and severe complications in data handling.

In supervised problems, noise can affect the input features, the output values or both. When noise is present in the input attributes, it is usually referred as attribute noise.

The worst case is when the noise affects the output attribute, as this means that the bias introduced will be greater. As this kind of noise has been deeply studied in classification, it is usually known as class noise.

In order to treat noise in data mining, two main approaches are commonly used in the data preprocessing literature. The first one is to correct the noise by using data polishing methods, especially if it affects the labeling of an instance.

Even partial noise correction is claimed to be beneficial, but it is a difficult task and usually limited to small amounts of noise. The second is to use noise filters, which identify and remove the noisy instances in the training data and do not require the data mining technique to be modified.

Factoring in the above situations, it can be said conclusively that not every time we can save 100 percent of our collected data but we can do damage control and so far the currently the **Amazon data is on 70% fresh and the company data is on 90% fresh.**

7. EXPLORATORY DATA ANALYSIS

Since, the project tends to be in the Data Science field, EDA is to be encouraged to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis (IDA), which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA.

The objectives of EDA are to:

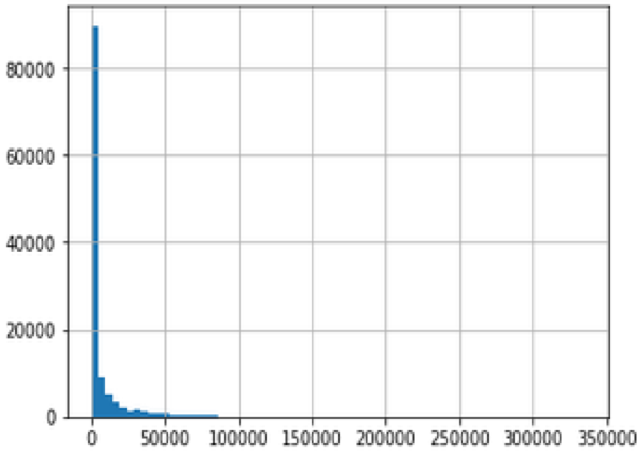
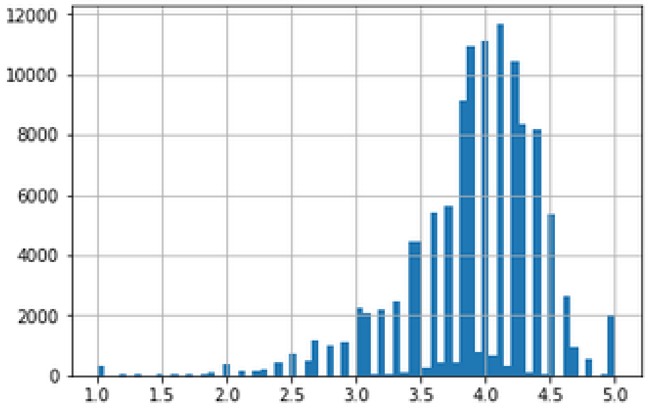
- Suggest hypotheses about the causes of observed phenomena
- Assess assumptions on which statistical inference will be base
- Support the selection of appropriate statistical tools and techniques
- Provide a basis for further data collection through surveys or experiments

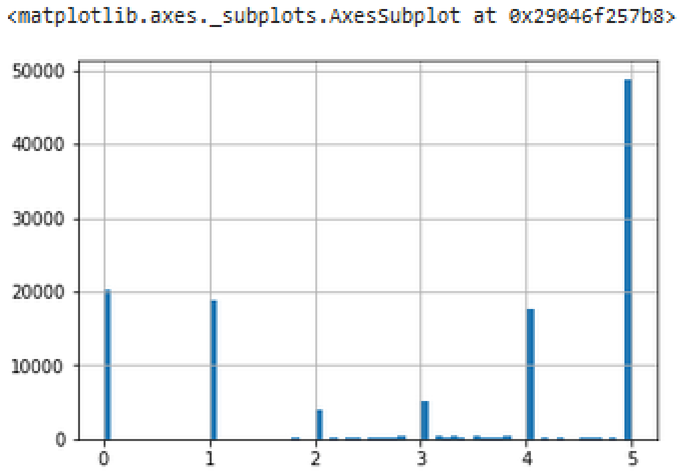
Many EDA techniques have been adopted into data mining, as well as into big data analytics. They are also being taught to young students as a way to introduce them to statistical thinking. Findings from EDA are orthogonal to the primary analysis task.

7.1 General Statistics

Dataset	**Amazon.csv**
Columns	Categorical Columns: Category Product Brand Seller Numerical Columns: Price FeedBack_Rating Product_Rating Packaging_Rating Courier_Rating
Shape	(11,19,986 rows) **and** (9 columns)
Categorical Statistics	Total Sellers: 2903 Total Products: 3224 Total Categories: 88 Total Brands: 967

7.2 Observed Statistics

Graphical Representations	***Conclusions***
	***Column:*** Price
	Plot Type: Histogram
	Analysis: Most of the products in the dataset have prices between 0 and 20,000. A rare but significant amount of products have prices between 50,000 to 1, 00,000. Although there are products for which price could go up to 3,50,000
<matplotlib.axes._subplots.AxesSubplot at 0x290461ddb00> 	***Column:*** Product Rating
	Plot Type: Histogram
	Analysis: Most of the product rating in the dataset have value between 3.5 and 4.5, which is understandable since most people have tendency to rate between 3 to 4 stars and lesser amount of products have received extreme reactions 0 to 2.5 : Steady Increase 2.5 to 3.5: Gradual Increase 3.5 to 4: Steep Increase 4 to 5: Steep Decrease with peak at 5, which suggests there are some products which have nicer response.



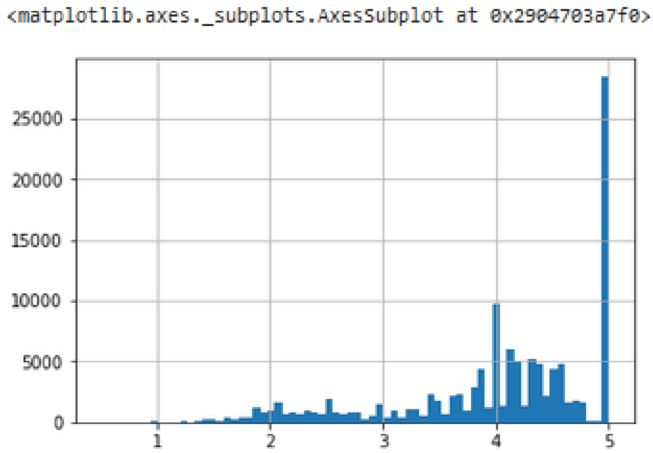
```
<matplotlib.axes._subplots.AxesSubplot at 0x29046f257b8>
```

Column: Feedback Rating

Plot Type: Histogram

Analysis:
Since most of customers on e-commerce have tendency to rate the feedbacks in whole digits, there are steep on the whole numbers.

Some ratings might appear in between those whole numbers, those are results of preprocessing and mean filling of null values.



```
<matplotlib.axes._subplots.AxesSubplot at 0x2904703a7f0>
```

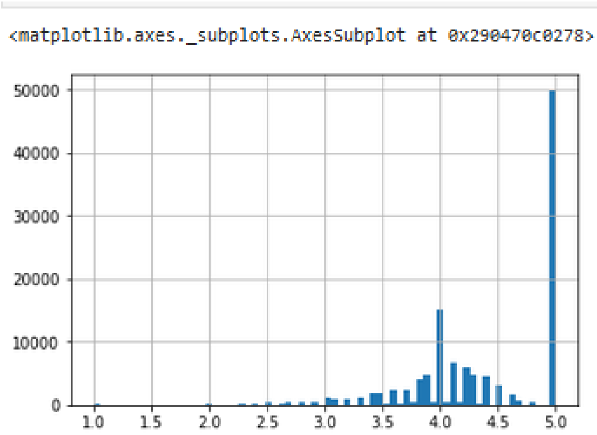
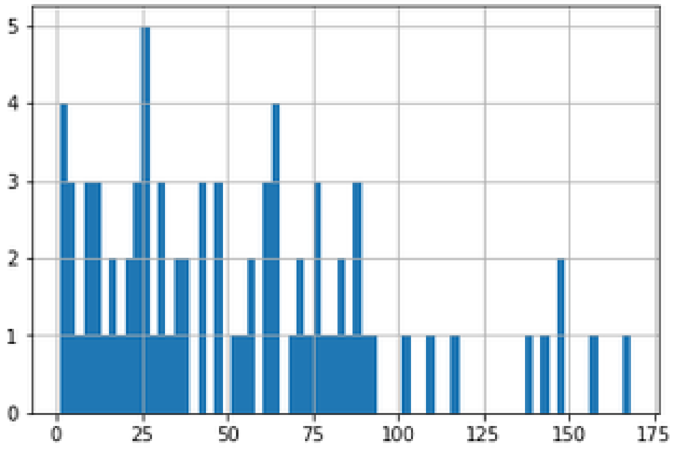
Column: Packaging Rating

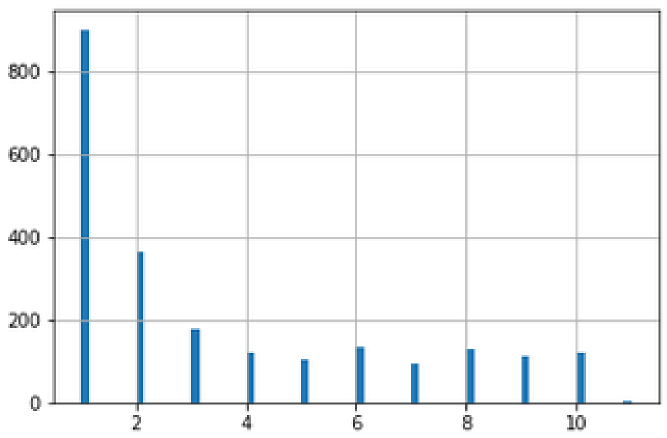
Plot Type: Histogram

Analysis:
Most people doesn’t seem to have concern with packaging since most people have rated 5.

Those who even faced minor difficulties gave 2 to 4 stars.

There are rare instances where people gave less than 2 stars indicating they might have received damaged goods or simply ruled on packaging being not to the code

 `<matplotlib.axes._subplots.AxesSubplot at 0x290470c0278>` 50000 40000 30000 20000 10000 0 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0	***Column:*** Packaging Rating ***Plot Type:*** Histogram ***Analysis:*** Most people doesn't seem to have concern with courier since most people have rated 5. Those who even faced minor difficulties gave 3 to 4.5 stars. There are rare instances where people gave less than 3 stars indicating they might have not preferred the courier service deployed by seller
	***Column:*** Sellers Per Category ***Plot Type:*** Histogram ***Analysis:*** This analysis shows number of sellers per category. Since there are 88 categories there 88 individual bars. Most categories have 0-25 sellers and same equal amount of other categories have 25-100 sellers. Although there are some categories like Televisions, Smart Phones etc. for which seller count goes above 100 and up to 175



Column: Sellers Per Product

Plot Type: Histogram

Analysis:

The bars are only on whole number because all products have exact amount of sellers.

Most products in the set have only one seller

Rest products have sellers between 2 to 10.

A rare amount of products have 11 sellers

8. CREATING ALGORITHM AND MODEL MAKING

8.1 Creating Hypotheses

As mentioned in database there were 5 criteria based on which a particular seller is to be scored:

- Price: Lesser the price, more the score
- Count: Popularity of the seller among the buyers
- Buyer Feedback: Individual buyer rating to the seller's attitude
- Courier Rating: Medium used by the seller to ship the product
- Product Rating: Rating of the product under that seller
- Packaging Rating: Handling the product by seller while shipping

Considerations:

- ✓ Each of the above mentioned criteria has individual value as well as whole value.
- ✓ Since every buyer has a perception of what he/she prefers the most out the criteria determines what kind of seller would he/she will prefer.
- ✓ To ensure the above, some kind of correlation is to be established on the basis of given dataset.
- ✓ But before establishing correlation, it was to be ensured that columns were to be in sync and scored.

Correlation measure how two observed variables are related to each other. It has been used in many different ways in data science.

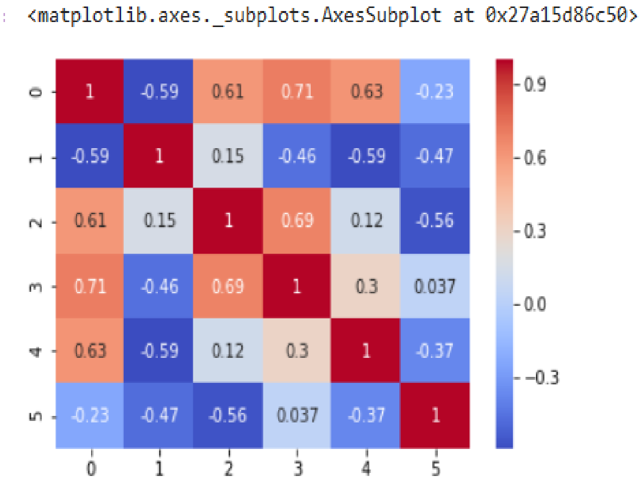
- Correlation is used in univariate analysis to identify which feature is more predictive for classification of regression task.
- To identify multi--collinearity in the feature set. Multi-collinearity reduce the accuracy of model.
- Identify causal relationship between variables.
- There are many other extension like canonical correlation analysis.

When information of a datasets is to be analysed, whose origin can be a raw database which may contain information of raw files, logs, spreadsheets. Then the main task is to analyse the data for drawing conclusions which is to be carried out by correlations, one can do dimensional reduction, machine learning, building models and can even run predictions which gives the path to see the dominant event sequences that leads to the target state and

It is firmly believed that correlation is a basic unit of foundation for many other modelling techniques(in data science), but before applying correlation you must analyse the data whether it is quantitative or qualitative so that models(algorithms) applied to the raw data must not give ambiguous answers.

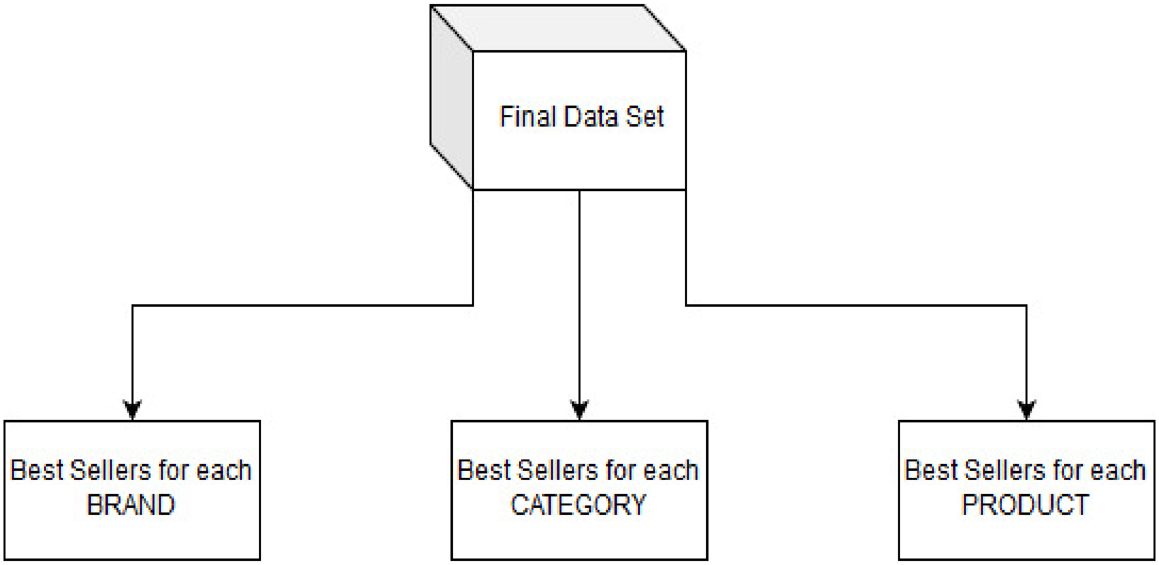
Correlation between features:

- Price Correlation : corr[0]
- Count Correlation : corr[1]
- Buyer_FeedBack Correlation : corr[2]
- Courier_Rating Correlation : corr[3]
- Product_Rating Correlation : corr[4]
- Packaging_Rating Correlation : corr[5]



```
<matplotlib.axes._subplots.AxesSubplot at 0x27a15d86c50>
```

8.2 Generating Results



Final Data Set
Best Sellers for each BRAND
Best Sellers for each CATEGORY
Best Sellers for each PRODUCT

Figure: 8.2(a) Drawing Different Conclusions

The given algorithm has two parameters, an entity and a final dataset.

Entities: BRAND or PRODUCT or CATEGORY

Algorithm:

```
def Best_Seller(entity,data):

    ALL_OVER = data[data['Brand'] ==
product].sort_values(by='ALL_OVER',ascending=False).reset_index(drop=True)

    Best_Seller_ALL_OVER = ALL_OVER['Seller'][0]

    Best_Seller_ALL_OVER_Rating = ALL_OVER['ALL_OVER'][0]


    FeedBack_Rating = data[data['Brand'] ==
product].sort_values(by='FeedBack_Rating',ascending=False).reset_index(drop=Tru
e)

    Best_Seller_FeedBack_Rating = FeedBack_Rating['Seller'][0]

    Best_Seller_FeedBack_Rating_Rating = FeedBack_Rating['FeedBack_Rating'][0]


    Product_Rating = data[data['Brand'] ==
product].sort_values(by='Product_Rating',ascending=False).reset_index(drop=True
)

    Best_Seller_Product_Rating = Product_Rating['Seller'][0]

    Best_Seller_Product_Rating_Rating = Product_Rating['Product_Rating'][0]


    Packaging_Rating = data[data['Brand'] ==
product].sort_values(by='Packaging_Rating',ascending=False).reset_index(drop=Tr
ue)

    Best_Seller_Packaging_Rating = Packaging_Rating['Seller'][0]

    Best_Seller_Packaging_Rating_Rating =
Packaging_Rating['Packaging_Rating'][0]


    Courier_Rating = data[data['Brand'] ==
product].sort_values(by='Courier_Rating',ascending=False).reset_index(drop=True
)

    Best_Seller_Courier_Rating = Courier_Rating['Seller'][0]

    Best_Seller_Courier_Rating_Rating = Courier_Rating['Courier_Rating'][0]


    return Best_Seller_ALL_OVER, Best_Seller_ALL_OVER_Rating,
Best_Seller_FeedBack_Rating, Best_Seller_FeedBack_Rating_Rating,
Best_Seller_Product_Rating, Best_Seller_Product_Rating_Rating,
Best_Seller_Packaging_Rating, Best_Seller_Packaging_Rating_Rating,
Best_Seller_Courier_Rating, Best_Seller_Courier_Rating_Rating
```

9. PRODUCT RECOMMENDATION ALGORITHM

Recommendation Systems usually rely on larger data sets and specifically need to be organized in a particular fashion. Because of this, we won't have a project to go along with this topic, instead we will have a more intensive walkthrough process on creating a recommendation system with Python with the same Movie Lens Data Set.

Note: The actual mathematics behind recommender systems is pretty heavy in Linear Algebra.

Methods Used

Two most common types of recommender systems are **Content-Based** and **Collaborative Filtering (CF)**.

- Collaborative filtering produces recommendations based on the knowledge of users’ attitude to items, that is it uses the "wisdom of the crowd" to recommend items.
- Content-based recommender systems focus on the attributes of the items and give you recommendations based on the similarity between them.

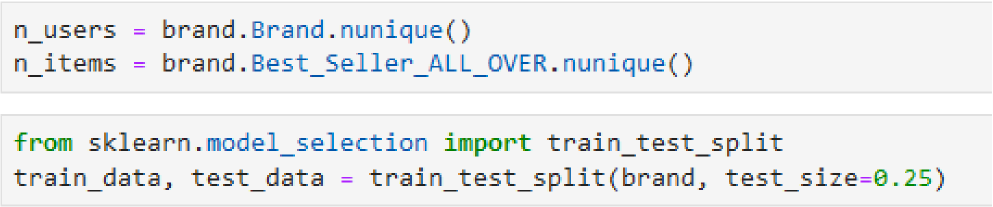
Collaborative Filtering

In general, Collaborative filtering (CF) is more commonly used than content-based systems because it usually gives better results and is relatively easy to understand (from an overall implementation perspective). The algorithm has the ability to do feature learning on its own, which means that it can start to learn for itself what features to use.

CF can be divided into **Memory-Based Collaborative Filtering** and **Model-Based Collaborative filtering**.

Train Test Split

Recommendation Systems by their very nature are very difficult to evaluate, but we will still show you how to evaluate them in this tutorial. In order to do this, we'll split our data into two sets. However, we won't do our classic X_train,X_test,y_train,y_test split. Instead we can actually just segement the data into two sets of data:



```
n_users = brand.Brand.nunique()
n_items = brand.Best_Seller_ALL_OVER.nunique()
```

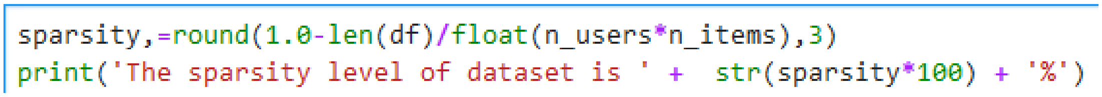
```
from sklearn.model_selection import train_test_split
train_data, test_data = train_test_split(brand, test_size=0.25)
```

Figure: 9 (a) Train Test Split

Model-based Collaborative Filtering

Model-based Collaborative Filtering is based on matrix factorization (MF) which has received greater exposure, mainly as an unsupervised learning method for latent variable decomposition and dimensionality reduction. Matrix factorization is widely used for recommender systems where it can deal better with scalability and sparsity than Memory-based CF. The goal of MF is to learn the latent preferences of users and the latent attributes of items from known ratings (learn features that describe the characteristics of ratings) to then predict the unknown ratings through the dot product of the latent features of users and items. When you have a very sparse matrix, with a lot of dimensions, by doing matrix factorization you can restructure the user-item matrix into low-rank structure, and you can represent the matrix by the multiplication of two low-rank matrices, where the rows contain the latent vector. You fit this matrix to approximate your original matrix, as closely as possible, by multiplying the low-rank matrices together, which fills in the entries missing in the original matrix.

Let's calculate the sparsity level of dataset: (73.435 %)



```
sparsity,=round(1.0-len(df)/float(n_users*n_items),3)
print('The sparsity level of dataset is ' +  str(sparsity*100) + '%')
```

Figure: 9 (b) Sparsity Level Check

Models that use both ratings and content features are called Hybrid Recommender Systems where both Collaborative Filtering and Content-based Models are combined. Hybrid recommender systems usually show higher accuracy than Collaborative Filtering or Content-based Models on their own: they are capable to address the cold-start problem better since if you don't have any ratings for a user or an item you could use the metadata from the user or item to make a prediction.

SVD

A well-known matrix factorization method is **Singular value decomposition (SVD)**. Collaborative Filtering can be formulated by approximating a matrix X by using singular value decomposition.

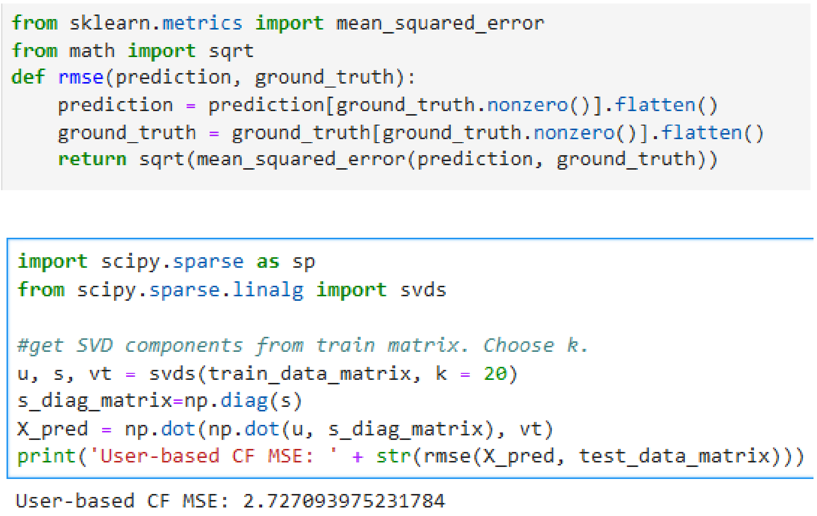
The winning team at the Netflix Prize competition used SVD matrix factorization models to produce product recommendations, for more information I recommend to read articles: Netflix Recommendations: Beyond the 5 stars and Netflix Prize and SVD. The general equation can be expressed as follows: $X = USV^T$

Given m x n matrix X:

- *U* is an *(m x r)* orthogonal matrix
- *S* is an *(r x r)* diagonal matrix with non-negative real numbers on the diagonal
- *V^T* is an *(r x n)* orthogonal matrix

Elements on the diagonal in S are known as *singular values of X.*

Matrix *X* can be factorized to *U*, *S* and *V*. The *U* matrix represents the feature vectors corresponding to the users in the hidden feature space and the *V* matrix represents the feature vectors corresponding to the items in the hidden feature space.



```
from sklearn.metrics import mean_squared_error
from math import sqrt
def rmse(prediction, ground_truth):
    prediction = prediction[ground_truth.nonzero()].flatten()
    ground_truth = ground_truth[ground_truth.nonzero()].flatten()
    return sqrt(mean_squared_error(prediction, ground_truth))
```

```
import scipy.sparse as sp
from scipy.sparse.linalg import svds

#get SVD components from train matrix. Choose k.
u, s, vt = svds(train_data_matrix, k = 20)
s_diag_matrix=np.diag(s)
X_pred = np.dot(np.dot(u, s_diag_matrix), vt)
print('User-based CF MSE: ' + str(rmse(X_pred, test_data_matrix)))
```

```
User-based CF MSE: 2.727093975231784
```

Figure: 9 (c) Training the model and testing for MSE

Carelessly addressing only the relatively few known entries is highly prone to overfitting. SVD can be very slow and computationally expensive. More recent work minimizes the squared error by applying alternating least square or stochastic gradient descent and uses regularization terms to prevent overfitting.

10. EVALUATION AND TESTING

There are different evaluation metrics for different performance metrics. For instance, if the machine learning model aims to predict the daily stock then the RMSE (root mean squared error) will have to be considered for evaluation.

If the model aims to classify spam emails then performance metrics like average accuracy, AUC and log loss have to be considered. A common question that professionals often have when evaluating the performance of a machine learning model is that which dataset they should use to measure the performance of the machine learning model.

Looking at the performance metrics on the trained dataset is helpful but is not always right because the numbers obtained might be overly optimistic as the model is already adapted to the training dataset. Machine learning model performances should be measured and compared using validation and test sets to identify the best model based on model accuracy and over-fitting.

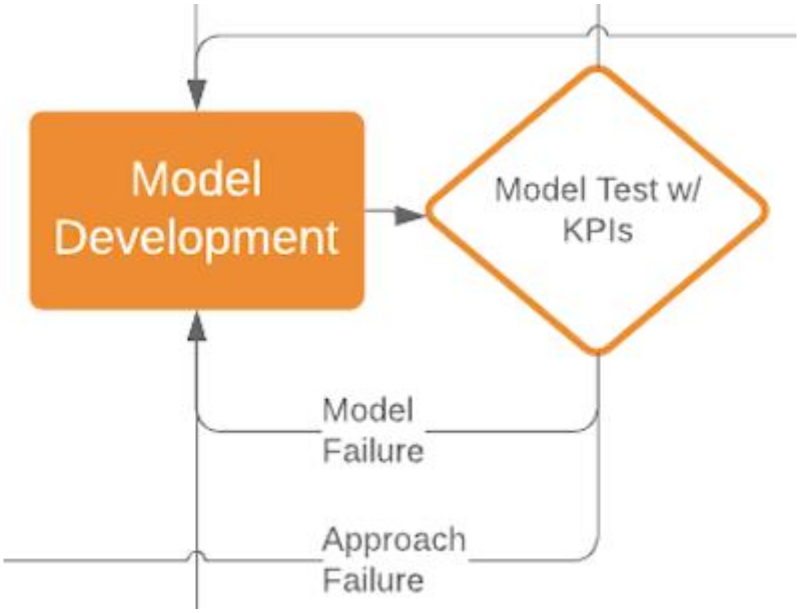


Figure 10(a): Model failure entails iteration

While developing the model, different versions of it (and the data processing pipeline accompanying it) should be continuously tested against the predetermined hard metric(s).

This gives a rough estimate of progress and also allows the data scientist to decide when the model seems to be working well enough to warrant the overall KPI check. Do note that this can be misleading, as getting from 50% to 70% accuracy, for example, is in many cases much easier than getting from 70% to 90% accuracy.

When tests show that a model is off the mark, we usually investigate it and its output to guide improvements. Sometimes, however, the gap in performance is very large, with different variations of the chosen research directions all falling short—an approach failure.

This might warrant a change in the research direction, sending the project back into the research phase. This is the aspect of data science projects that is hardest to accept: the very real possibility of backtracking.

Another possible result of approach failure is a change to the goal. With luck, it can be minor product-wise but restate the goal technically in a simpler way.

In this case, there might be instances, where the approach might decide the way the conclusion goes.

For instance here, if you find a best seller while looking for a product and on the other hand you find a best seller while looking for a brand and then getting to the product

- Example: If the decision is made for Sony Xperia XZ

The choice can be divided among three categories:

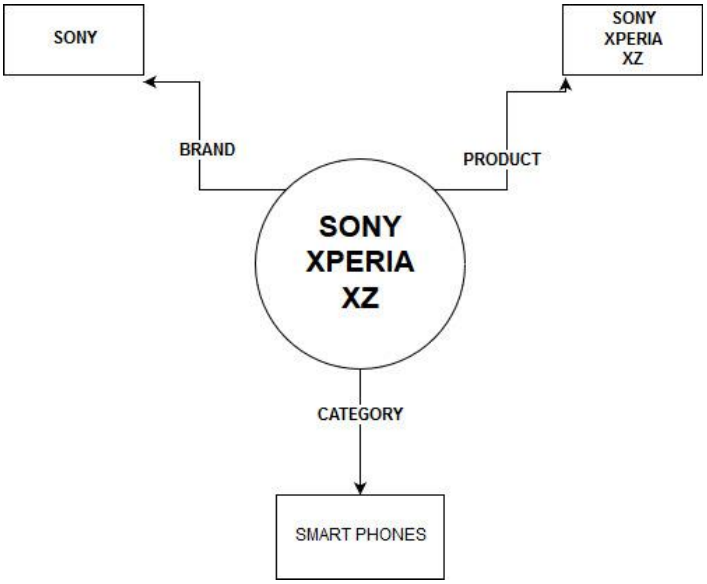


Figure 10(b) Breaking down the search

Now there are three different but alike generated results through which this information can be processed and there is a possibility that in each case, we might arrive on different conclusion.

The generated result had following schema:

- FOR BRAND, CATEGORY AND PRODUCT:

'Best_Seller_ALL_OVER', 'Best_Seller_ALL_OVER_Rating',
'Best_Seller_FeedBack_Rating', 'Best_Seller_FeedBack_Rating_Rating',
'Best_Seller_Product_Rating', 'Best_Seller_Product_Rating_Rating',
'Best_Seller_Packaging_Rating', 'Best_Seller_Packaging_Rating_Rating',
'Best_Seller_Courier_Rating', 'Best_Seller_Courier_Rating_Rating'

These are the tree datasets we have to process the given result through and generate what we call a truth datasets

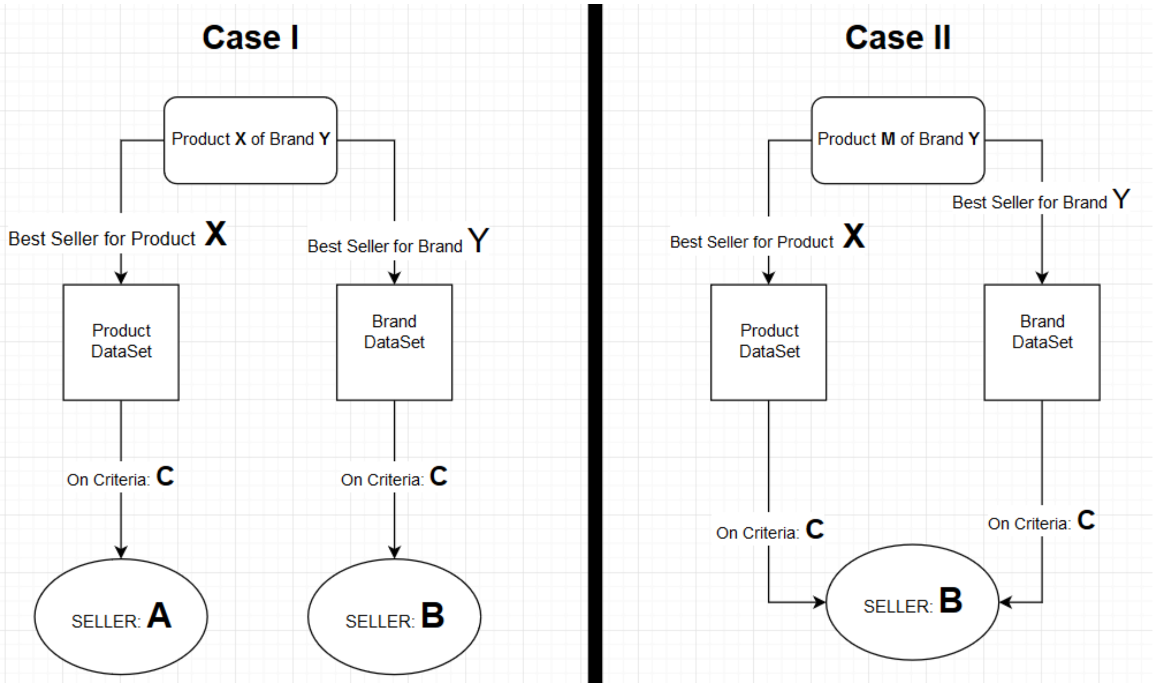


Figure 10(c) Product – Brand Contradiction

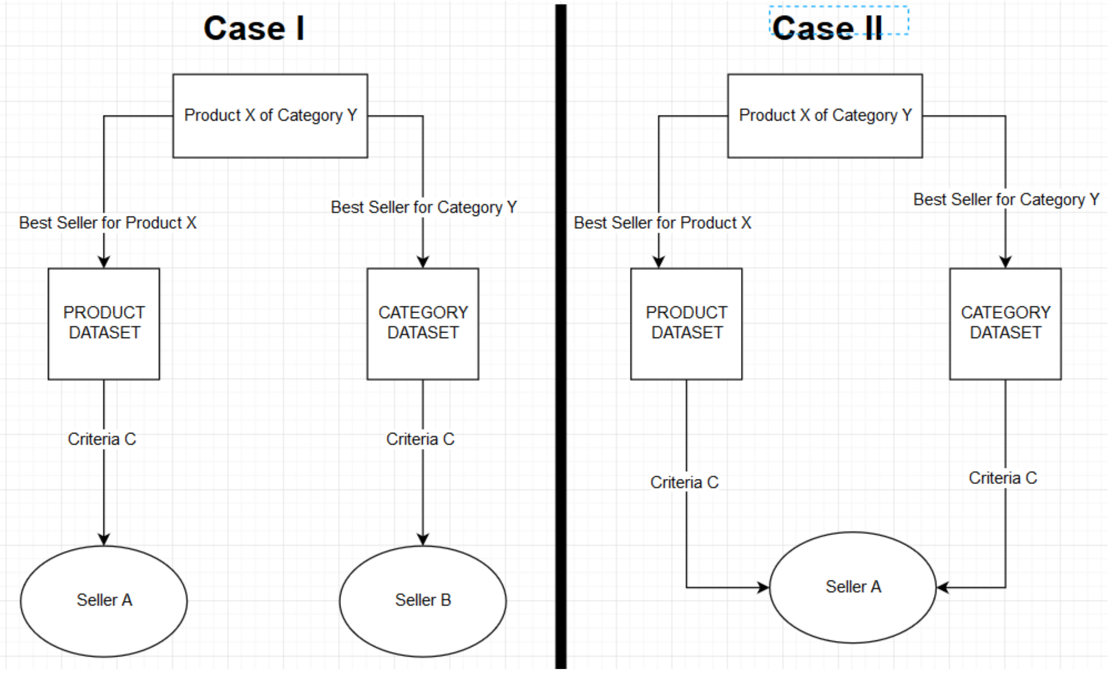


Figure 10(d) Product – Category Contradiction

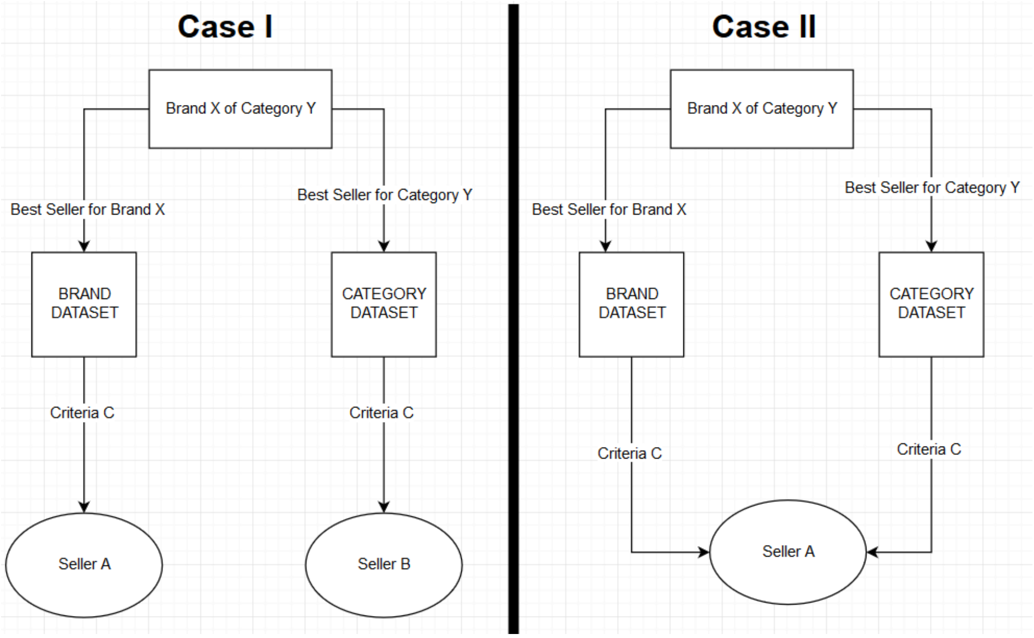


Figure 10(e) Category – Brand Contradiction

Combined Truth Dataset:
TRUE -> Case II
FALSE -> Case I

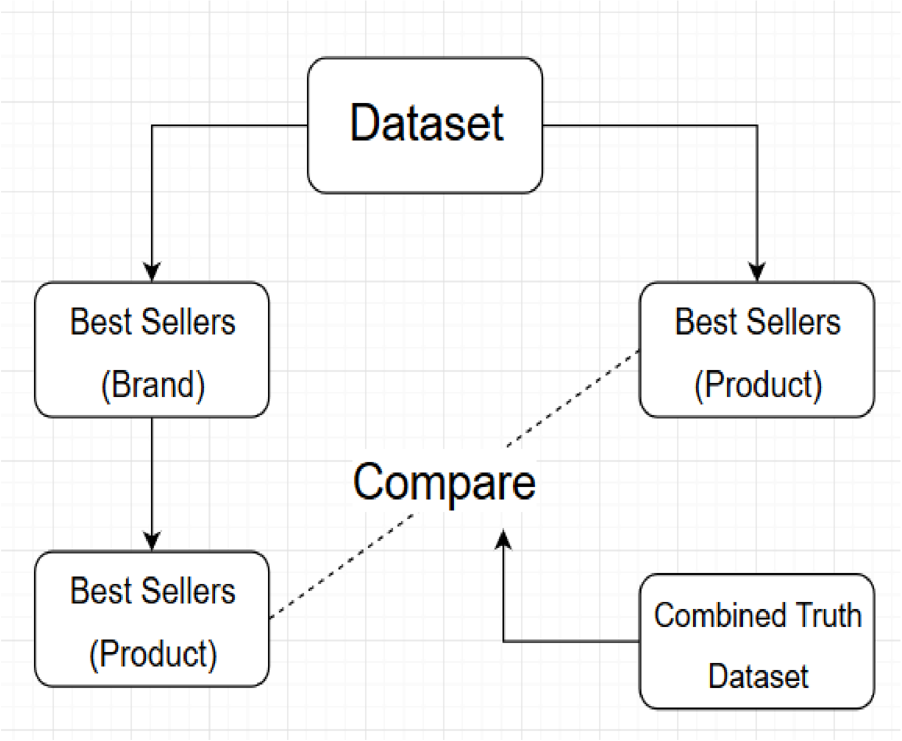
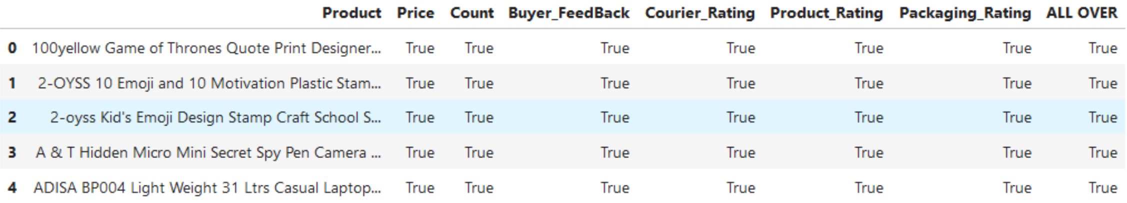


Figure 10(f) Process for generating combined truth dataset

Since the effect of both cases with respect to the features reflects upon the ALL_OVER rating, we consider that as our output.



	Product	Price	Count	Buyer_FeedBack	Courier_Rating	Product_Rating	Packaging_Rating	ALL OVER
0	100yellow Game of Thrones Quote Print Designer...	True	True	True	True	True	True	True
1	2-OYSS 10 Emoji and 10 Motivation Plastic Stam...	True	True	True	True	True	True	True
2	2-oyss Kid's Emoji Design Stamp Craft School S...	True	True	True	True	True	True	True
3	A & T Hidden Micro Mini Secret Spy Pen Camera ...	True	True	True	True	True	True	True
4	ADISA BP004 Light Weight 31 Ltrs Casual Laptop...	True	True	True	True	True	True	True

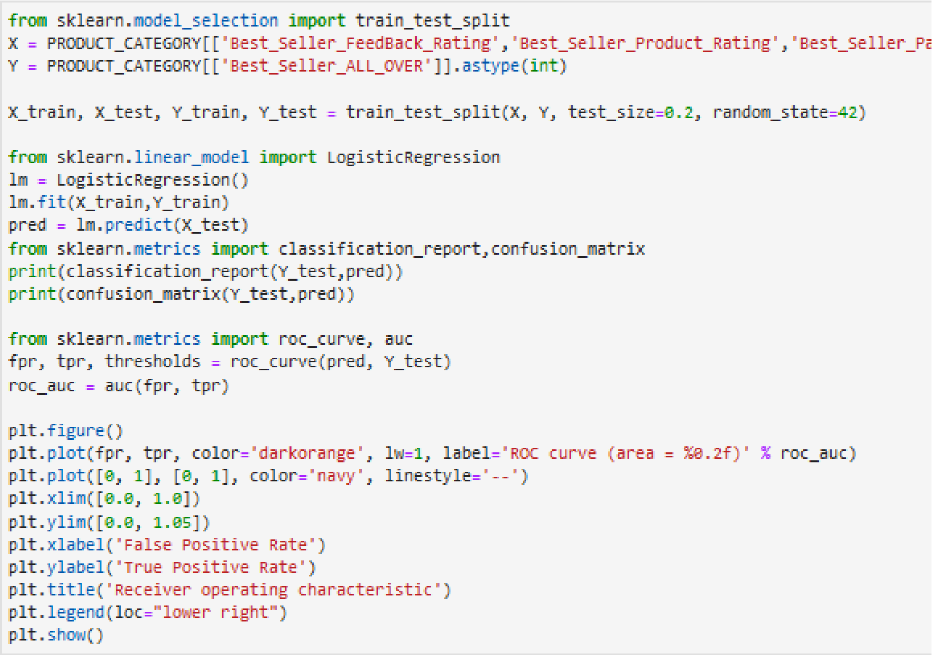
Figure 10(g) The Combined Truth Dataset

Now generated combined truth dataset can be used for testing our hypotheses using various machine learning models.

10.1 Applying Logistic Regression

In logistic regression, we take the output of the linear function and squash the value within the range of [0,1] using the sigmoid function(logistic function). The Sigmoid-Function is an S-shaped curve that can take any real-valued number and map it into a value between the range of 0 and 1, but never exactly at those limits. Typically, if the squashed value is greater than a threshold value we assign it a label 1, else we assign it a label 0. This justifies the name 'logistic regression'.

Since the given truth dataset has only two labels – True or False – but multiple features we can use Logistic regression on the given result.



```
from sklearn.model_selection import train_test_split
X = PRODUCT_CATEGORY[['Best_Seller_FeedBack_Rating','Best_Seller_Product_Rating','Best_Seller_Pa
Y = PRODUCT_CATEGORY[['Best_Seller_ALL_OVER']].astype(int)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

from sklearn.linear_model import LogisticRegression
lm = LogisticRegression()
lm.fit(X_train,Y_train)
pred = lm.predict(X_test)
from sklearn.metrics import classification_report,confusion_matrix
print(classification_report(Y_test,pred))
print(confusion_matrix(Y_test,pred))

from sklearn.metrics import roc_curve, auc
fpr, tpr, thresholds = roc_curve(pred, Y_test)
roc_auc = auc(fpr, tpr)

plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=1, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.show()
```

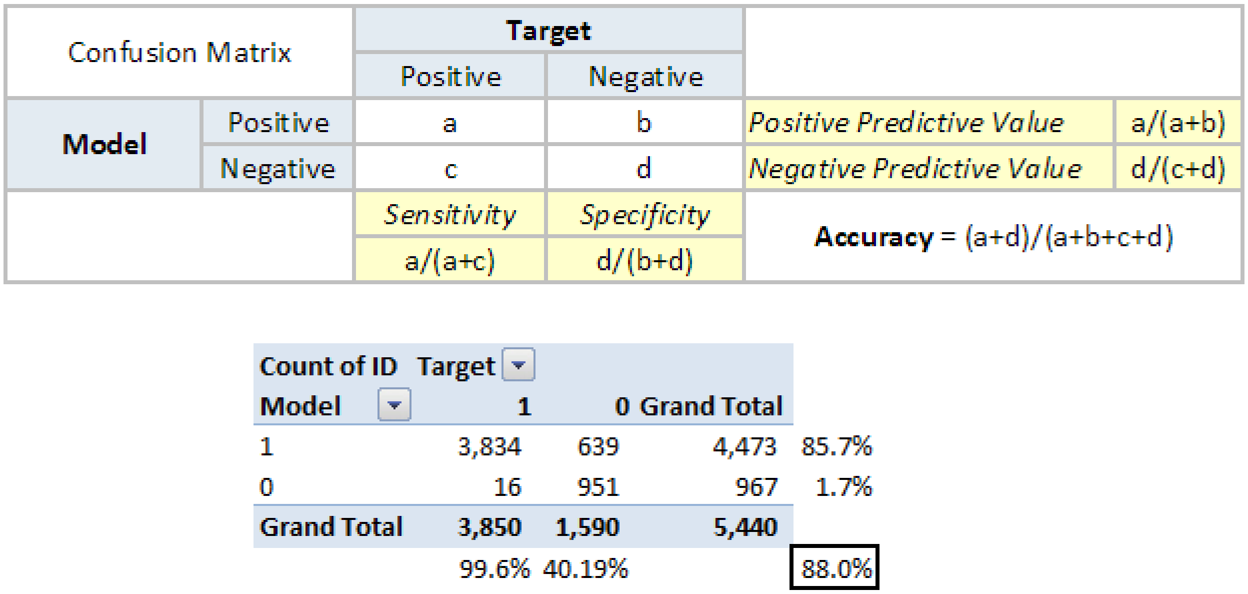
Figure 10.1(a) Logistic Regression Model

10.2 Confusion Matrix and Classification Report

A confusion matrix is an N X N matrix, where N is the number of classes being predicted. For the problem in hand, we have N=2, and hence we get a 2 X 2 matrix. Here are a few definitions, you need to remember for a confusion matrix :

- **Accuracy** : the proportion of the total number of predictions that were correct.
- **Positive Predictive Value or Precision** : the proportion of positive cases that were correctly identified.
- **Negative Predictive Value** : the proportion of negative cases that were correctly identified.

- **Sensitivity or Recall** : the proportion of actual positive cases which are correctly identified.
- **Specificity** : the proportion of actual negative cases which are correctly identified.



Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

Count of ID	Target			
Model	1	0	Grand Total	
1	3,834	639	4,473	85.7%
0	16	951	967	1.7%
Grand Total	3,850	1,590	5,440	
	99.6%	40.19%		88.0%

Figure 10.2(a) Confusion Matrix Introduction

The accuracy for the problem in hand comes out to be 88%. As you can see from the above two tables, the Positive predictive Value is high, but negative predictive value is quite low. Same holds for Senstivity and Specificity. This is primarily driven by the threshold value we have chosen. If we decrease our threshold value, the two pairs of starkly different numbers will come closer.

In general we are concerned with one of the above defined metric. For instance, in a pharmaceutical company, they will be more concerned with minimal wrong positive diagnosis.

Hence, they will be more concerned about high Specificity. On the other hand an attrition model will be more concerned with Senstivity. Confusion matrix are generally used only with class output models.

The classification report visualizer displays the precision, recall, F1, and support scores for the model. In order to support easier interpretation and problem detection, the report integrates numerical scores with a color-coded heatmap. All heatmaps are in the range (0.0, 1.0) to facilitate easy comparison of classification models across different classification reports.

precision

Precision is the ability of a classiifer not to label an instance positive that is actually negative. For each class it is defined as as the ratio of true positives to the sum of true and false positives. Said another way, “for all instances classified positive, what percent was correct?”

recall

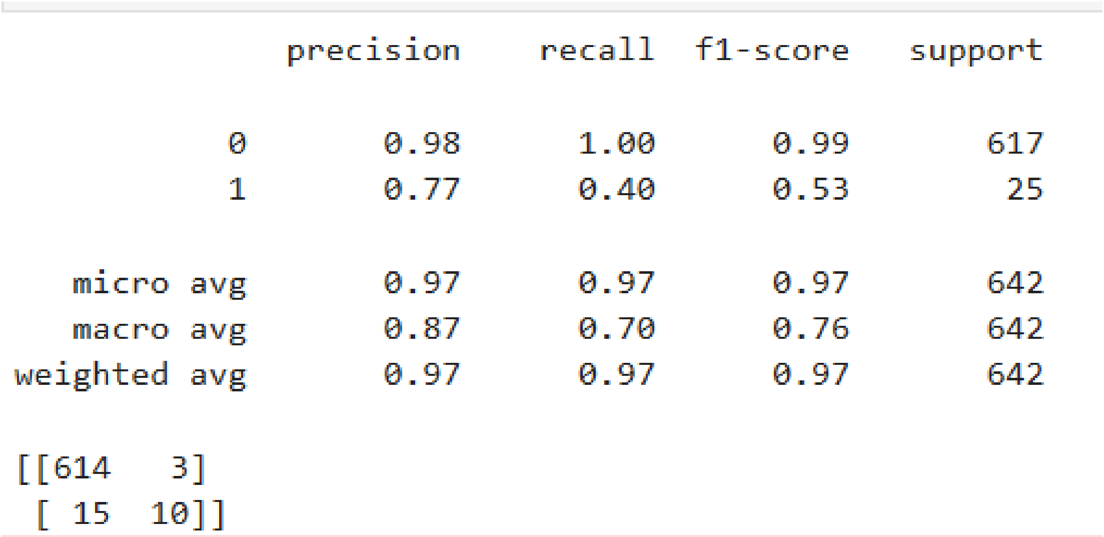
Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives. Said another way, “for all instances that were actually positive, what percent was classified correctly?”

f1 score

The F_1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F_1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F_1 should be used to compare classifier models, not global accuracy.

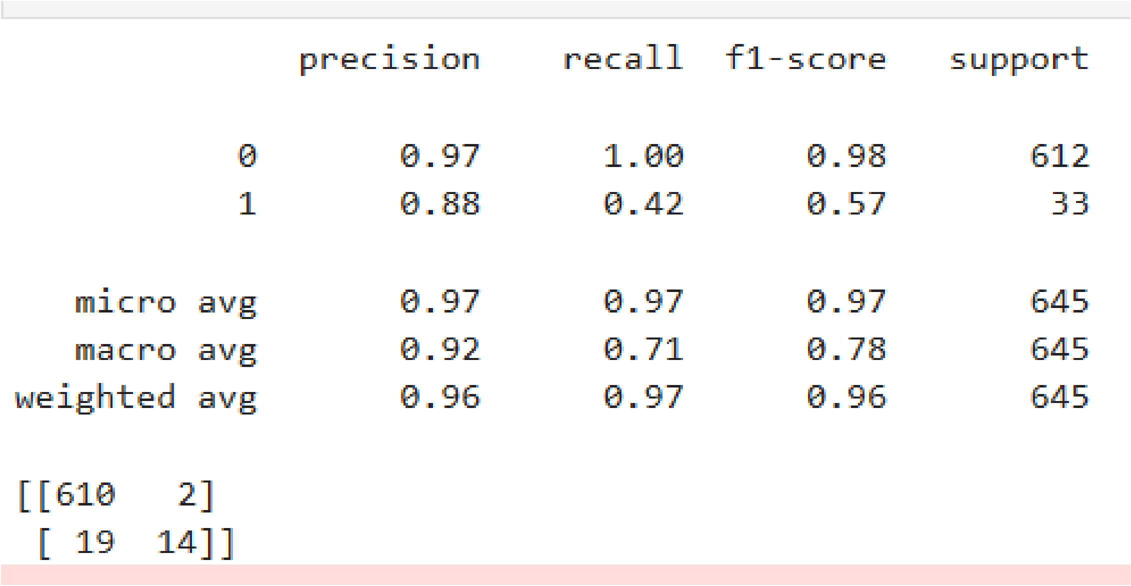
support

Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing. Support doesn’t change between models but instead diagnoses the evaluation process.



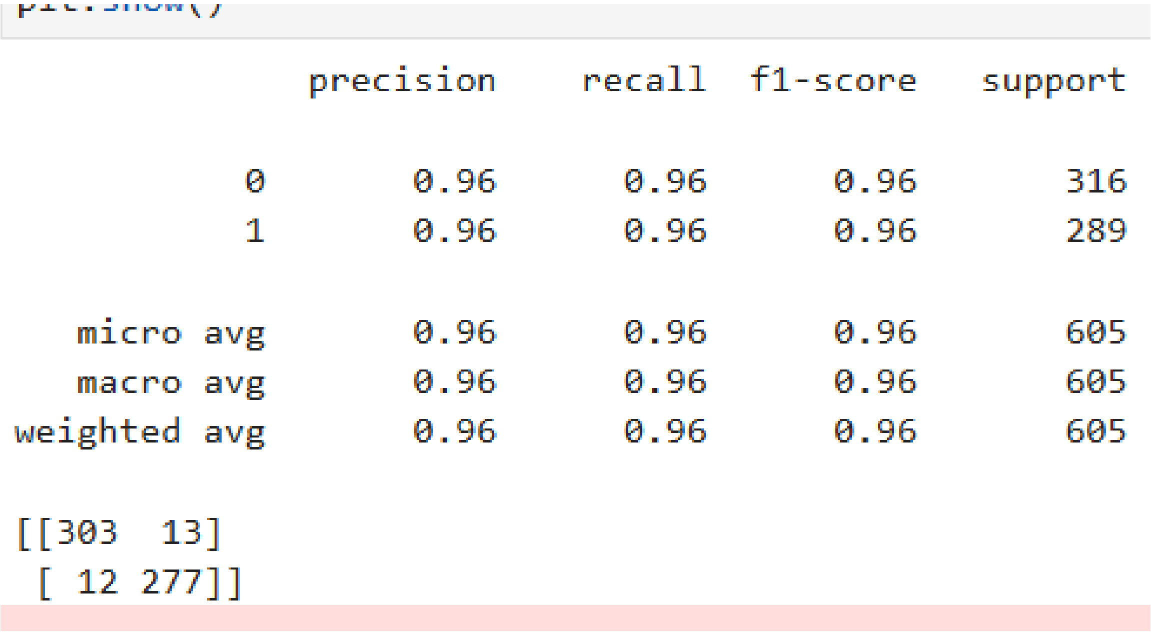
```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       617
           1       0.77      0.40      0.53        25

   micro avg       0.97      0.97      0.97       642
   macro avg       0.87      0.70      0.76       642
weighted avg       0.97      0.97      0.97       642

[[614   3]
 [ 15  10]]
```

Figure 10.2(b) Confusion Matrix and Classification Report for Product – Category Evaluation



```
              precision    recall  f1-score   support

           0       0.97      1.00      0.98       612
           1       0.88      0.42      0.57        33

   micro avg       0.97      0.97      0.97       645
   macro avg       0.92      0.71      0.78       645
weighted avg       0.96      0.97      0.96       645

[[610   2]
 [ 19  14]]
```

Figure 10.2(c) Confusion Matrix and Classification Report for Brand – Category Evaluation



```
plt.show()
```

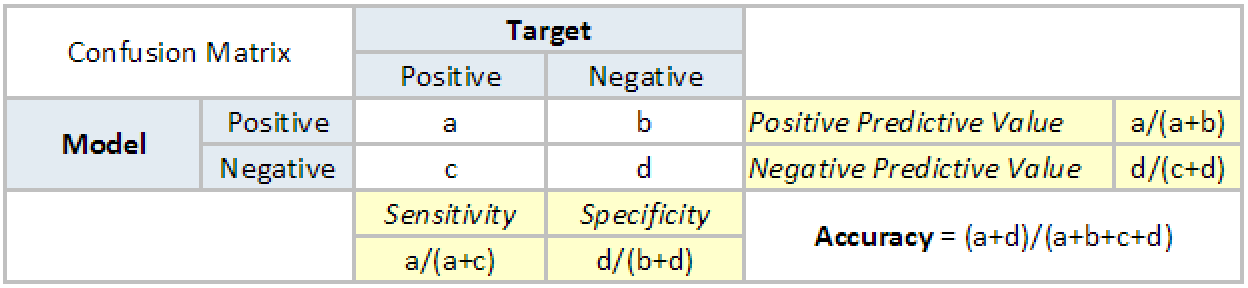
```
              precision    recall  f1-score   support

           0       0.96      0.96      0.96       316
           1       0.96      0.96      0.96       289

   micro avg       0.96      0.96      0.96       605
   macro avg       0.96      0.96      0.96       605
weighted avg       0.96      0.96      0.96       605

[[303  13]
 [ 12 277]]
```

Figure 10.2(d) Confusion Matrix and Classification Report for Product – Brand Evaluation

10.3 Area under AUC-ROC Curve

This is again one of the popular metrics used in the industry. The biggest advantage of using ROC curve is that it is independent of the change in proportion of responders. This statement will get clearer in the following sections.

Let’s first try to understand what ROC (Receiver operating characteristic) curve is. If we look at the confusion matrix below, we observe that for a probabilistic model, we get different value for each metric.



Confusion Matrix		**Target**			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

Hence, for each sensitivity, we get a different specificity. The two vary as follows:

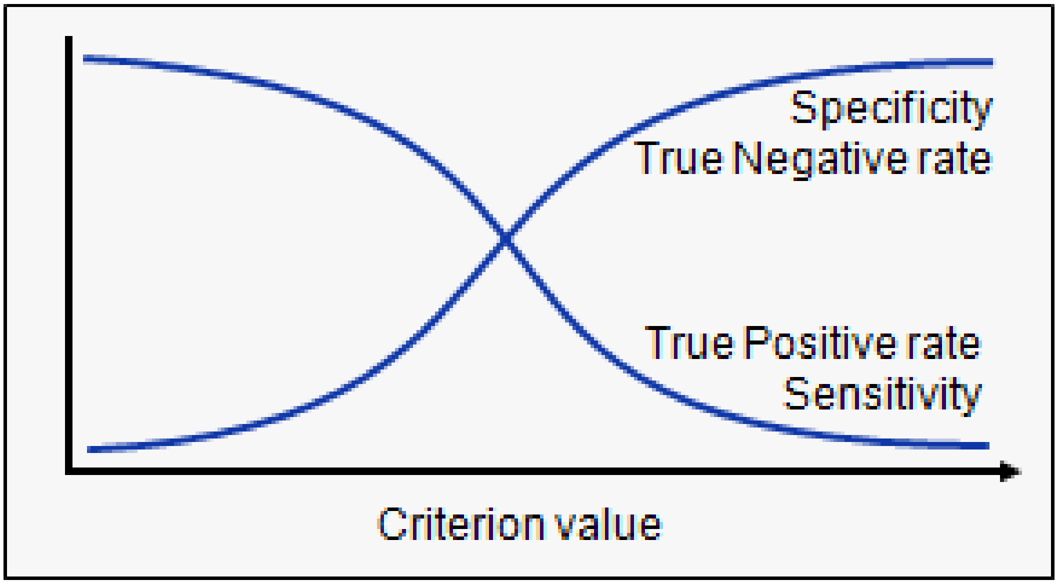
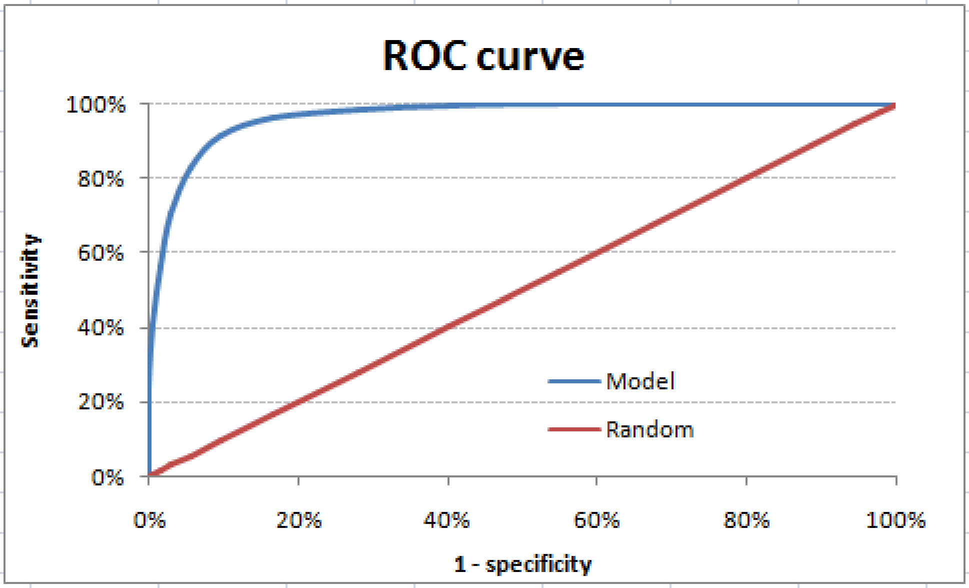


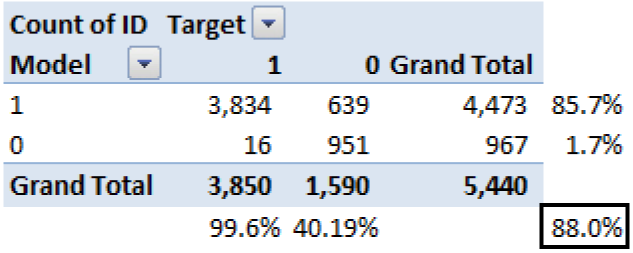
Figure 10.3(a) Criterion Value

The ROC curve is the plot between sensitivity and (1- specificity). (1- specificity) is also known as false positive rate and sensitivity is also known as True Positive rate.

Following is the ROC curve for the case in hand.



Let’s take an example of threshold = 0.5 (refer to confusion matrix). Here is the confusion matrix:



Count of ID	Target ▾			
Model ▾	1	0	Grand Total	
1	3,834	639	4,473	85.7%
0	16	951	967	1.7%
Grand Total	3,850	1,590	5,440	
	99.6%	40.19%		88.0%

As you can see, the sensitivity at this threshold is 99.6% and the (1-specificity) is ~60%. This coordinate becomes on point in our ROC curve. To bring this curve down to a single number, we find the area under this curve (AUC).

Note that the area of Entire Square is 1*1 = 1. Hence AUC itself is the ratio under the curve and the total area. For the case in hand, we get AUC ROC as 96.4%. Following are a few thumb rules:

- .90-1 = excellent (A)
- .80-.90 = good (B)
- .70-.80 = fair (C)
- .60-.70 = poor (D)
- .50-.60 = fail (F)

We see that we fall under the excellent band for the current model. But this might simply be over-fitting. In such cases it becomes very important to in-time and out-of-time validations.

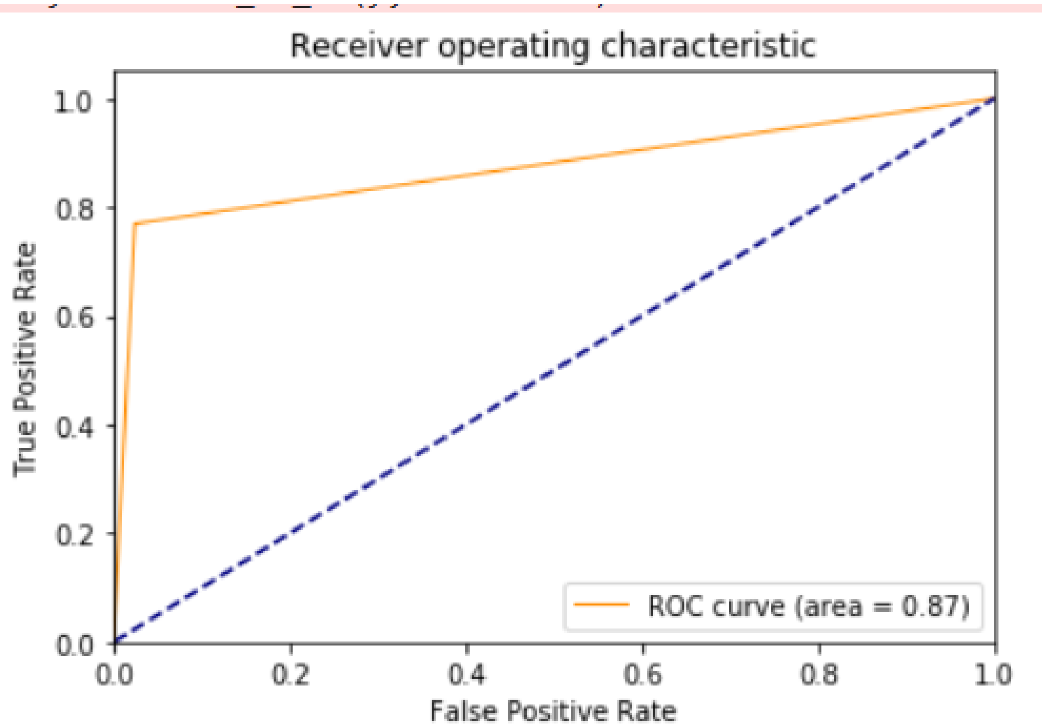


Figure 10.3(b) AUC-ROC curve Product – Category Evaluation

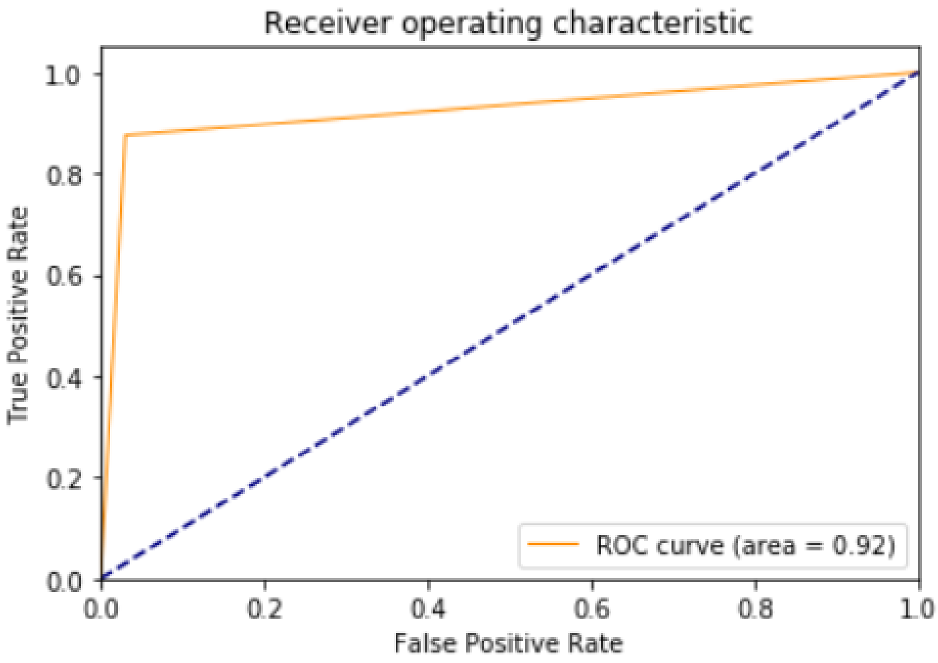


Figure 10.3(c) AUC-ROC curve Product – Category Evaluation

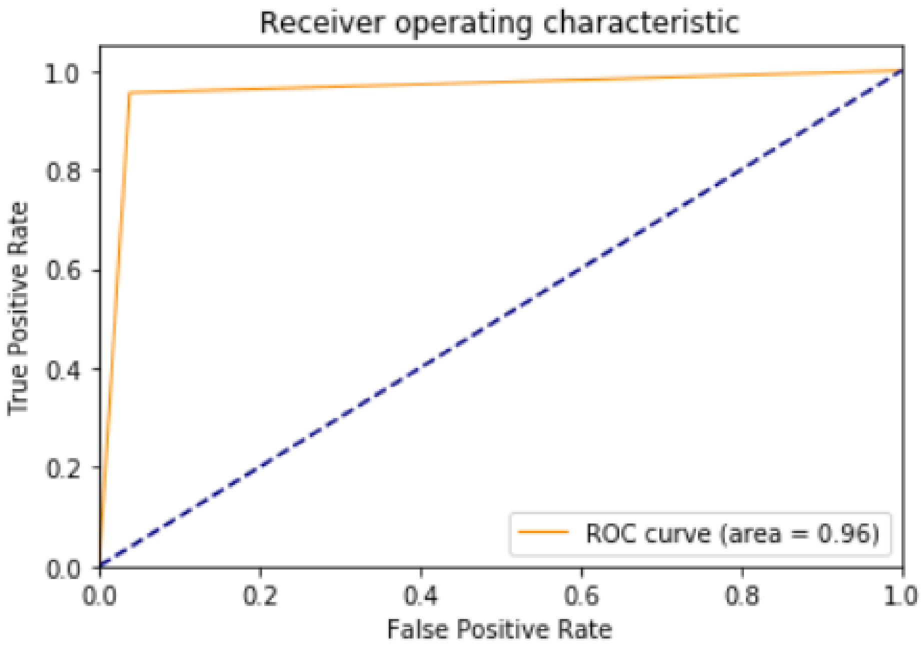


Figure 10.3(d) AUC-ROC curve Product – Category Evaluation

Points to Remember:

1. For a model which gives class as output, will be represented as a single point in ROC plot.

2. Such models cannot be compared with each other as the judgement needs to be taken on a single metric and not using multiple metrics.

3. For instance, model with parameters (0.2,0.8) and model with parameter (0.8,0.2) can be coming out of the same model, hence these metrics should not be directly compared.

4. In case of probabilistic model, we were fortunate enough to get a single number which was AUC-ROC.

5. But still, we need to look at the entire curve to make conclusive decisions. It is also possible that one model performs better in some region and other performs better in other.

Advantages of using ROC

Why should you use ROC and not metrics like lift curve?

Lift is dependent on total response rate of the population. Hence, if the response rate of the population changes, the same model will give a different lift chart. A solution to this concern can be true lift chart (finding the ratio of lift and perfect model lift at each decile). But such ratio rarely makes sense for the business.

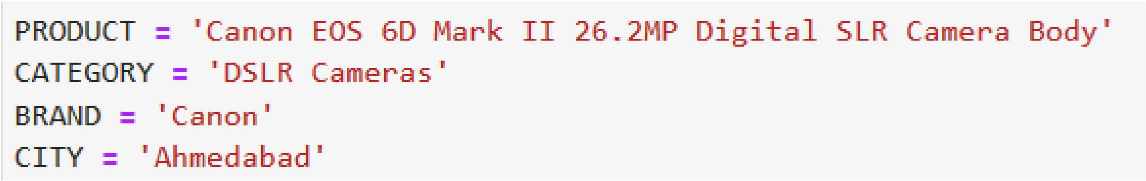
ROC curve on the other hand is almost independent of the response rate. This is because it has the two axis coming out from columnar calculations of confusion matrix. The numerator and denominator of both x and y axis will change on similar scale in case of response rate shift.

11. CREATING PRODUCT AND DEPLOYMENT

11.1 Customer Point-of-View

One of the most important part is how this benefits the customer of the organization. Customers need information about sellers for products they buy, that they can trust and rely for the best services.

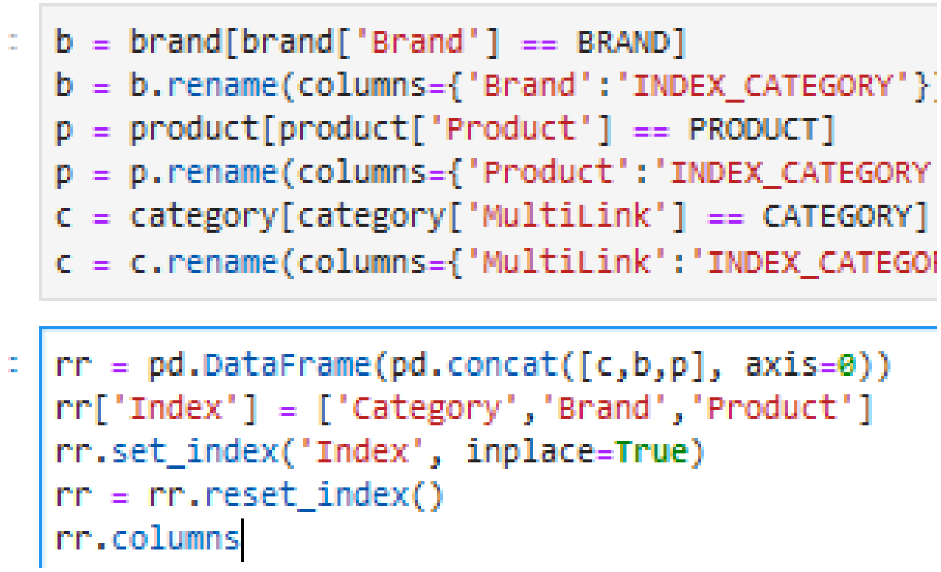
Let's say following are the details a customer has decided upon and the customer resides in 'Ahmedabad'.



```
PRODUCT = 'Canon EOS 6D Mark II 26.2MP Digital SLR Camera Body'
CATEGORY = 'DSLR Cameras'
BRAND = 'Canon'
CITY = 'Ahmedabad'
```

Figure 11.1(a): Example details

The algorithm first scans in the "Best_Seller_for_Brands.csv" for the given brand and determines best seller. Similarly, it will look for the product and category in both "Best_Seller_for_Products.csv" and "Best_Seller_for_Category.csv."

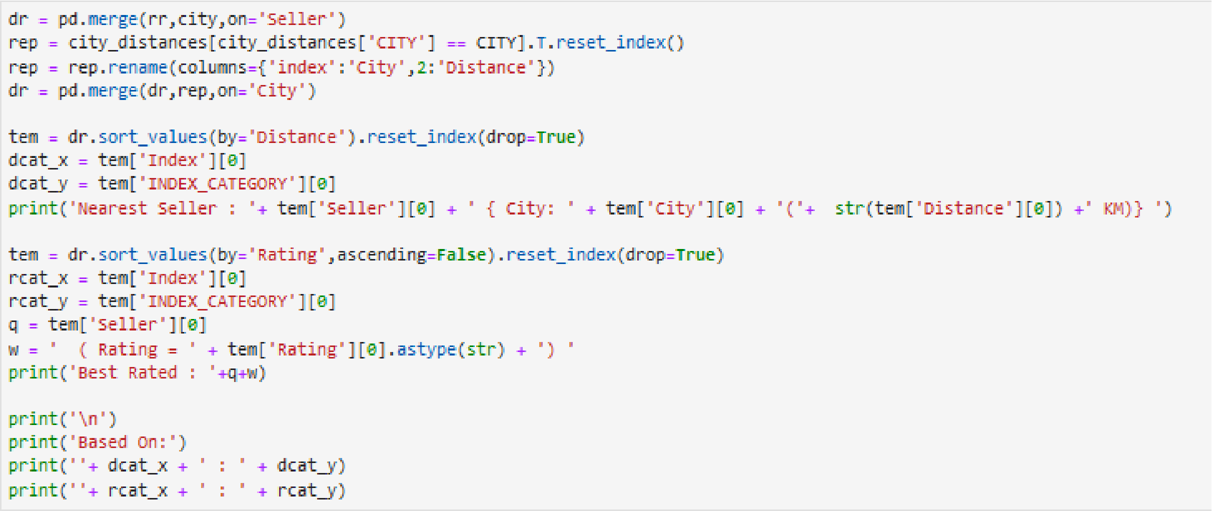


```
b = brand[brand['Brand'] == BRAND]
b = b.rename(columns={'Brand':'INDEX_CATEGORY'}
p = product[product['Product'] == PRODUCT]
p = p.rename(columns={'Product':'INDEX_CATEGORY
c = category[category['MultiLink'] == CATEGORY]
c = c.rename(columns={'MultiLink':'INDEX_CATEGO
```

```
rr = pd.DataFrame(pd.concat([c,b,p], axis=0))
rr['Index'] = ['Category','Brand','Product']
rr.set_index('Index', inplace=True)
rr = rr.reset_index()
rr.columns
```

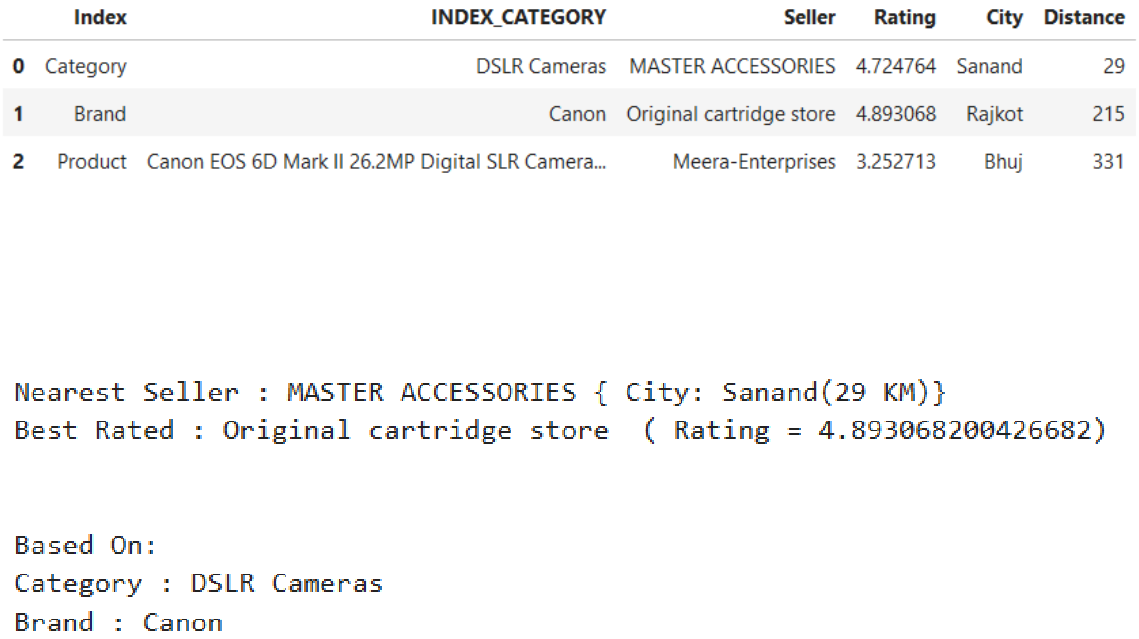
Figure 11.1(b)Generating results from different datasets

Also, out of the best sellers, algorithm also determine the nearest best seller.



```
dr = pd.merge(rr,city,on='Seller')
rep = city_distances[city_distances['CITY'] == CITY].T.reset_index()
rep = rep.rename(columns={'index':'City',2:'Distance'})
dr = pd.merge(dr,rep,on='City')

tem = dr.sort_values(by='Distance').reset_index(drop=True)
dcat_x = tem['Index'][0]
dcat_y = tem['INDEX_CATEGORY'][0]
print('Nearest Seller : '+ tem['Seller'][0] + ' { City: ' + tem['City'][0] + '('+  str(tem['Distance'][0]) +' KM)} ')

tem = dr.sort_values(by='Rating',ascending=False).reset_index(drop=True)
rcat_x = tem['Index'][0]
rcat_y = tem['INDEX_CATEGORY'][0]
q = tem['Seller'][0]
w = '  ( Rating = ' + tem['Rating'][0].astype(str) + ') '
print('Best Rated : '+q+w)

print('\n')
print('Based On:')
print(''+ dcat_x + ' : ' + dcat_y)
print(''+ rcat_x + ' : ' + rcat_y)
```

Figure 11.1(c) Generating results with respect to distance



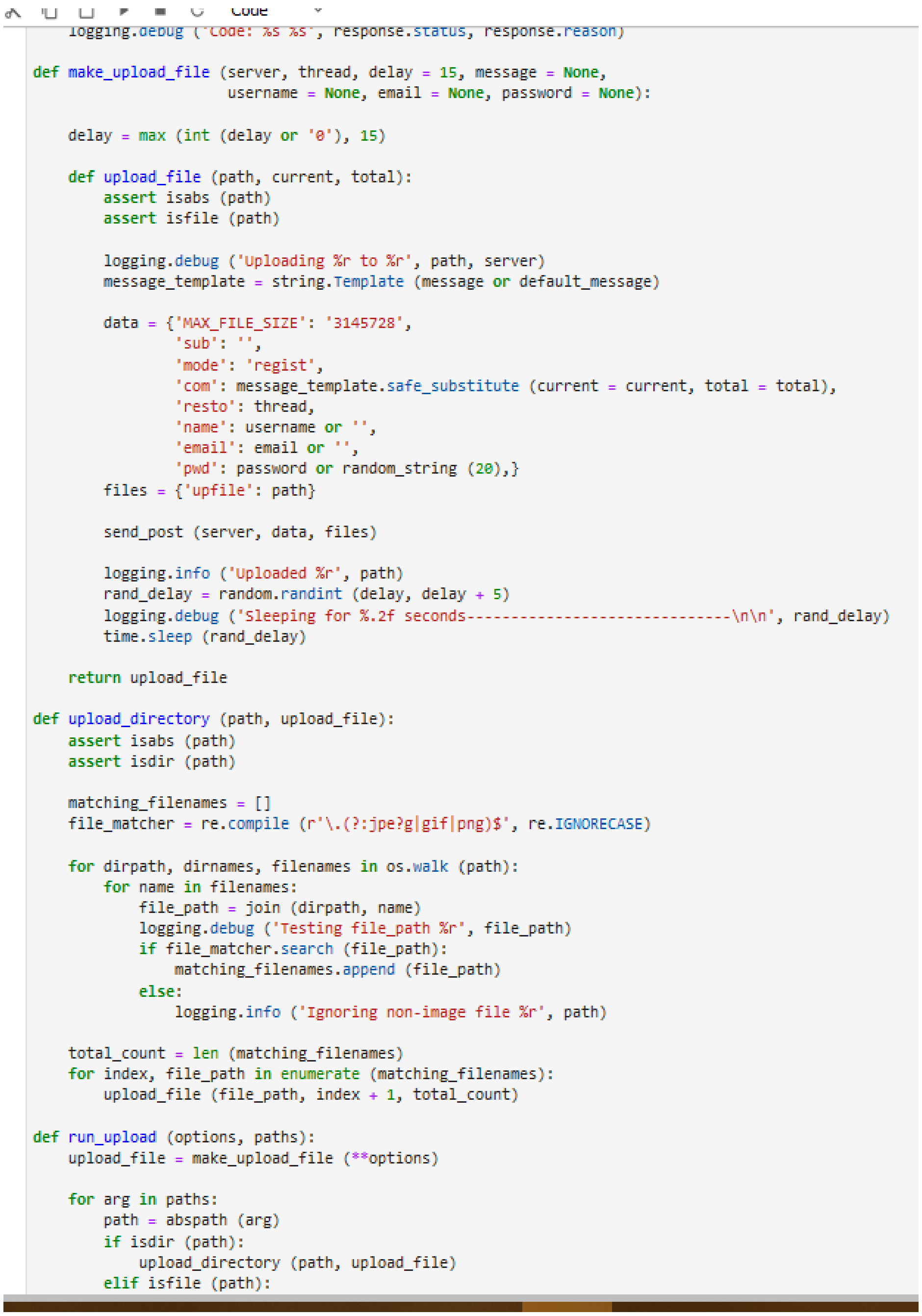
	Index	INDEX_CATEGORY	Seller	Rating	City	Distance
0	Category	DSLR Cameras	MASTER ACCESSORIES	4.724764	Sanand	29
1	Brand	Canon	Original cartridge store	4.893068	Rajkot	215
2	Product	Canon EOS 6D Mark II 26.2MP Digital SLR Camera...	Meera-Enterprises	3.252713	Bhuj	331

```
Nearest Seller : MASTER ACCESSORIES { City: Sanand(29 KM)}
Best Rated : Original cartridge store  ( Rating = 4.893068200426682)


Based On:
Category : DSLR Cameras
Brand : Canon
```

Figure 11.1(d)Generated Results to be displayed to customer

11.2 Creating Python Package

Packages are a way of structuring many packages and modules which helps in a well-organized hierarchy of data set, making the directories and modules easy to access. Just like there are different drives and folders in an OS to help us store files, similarly packages help us in storing other sub-packages and modules, so that it can be used by the user when necessary.



```python
    logging.debug ('Code: %s %s', response.status, response.reason)

def make_upload_file (server, thread, delay = 15, message = None,
                      username = None, email = None, password = None):

    delay = max (int (delay or '0'), 15)

    def upload_file (path, current, total):
        assert isabs (path)
        assert isfile (path)

        logging.debug ('Uploading %r to %r', path, server)
        message_template = string.Template (message or default_message)

        data = {'MAX_FILE_SIZE': '3145728',
                'sub': '',
                'mode': 'regist',
                'com': message_template.safe_substitute (current = current, total = total),
                'resto': thread,
                'name': username or '',
                'email': email or '',
                'pwd': password or random_string (20),}
        files = {'upfile': path}

        send_post (server, data, files)

        logging.info ('Uploaded %r', path)
        rand_delay = random.randint (delay, delay + 5)
        logging.debug ('Sleeping for %.2f seconds------------------------------\n\n', rand_delay)
        time.sleep (rand_delay)

    return upload_file

def upload_directory (path, upload_file):
    assert isabs (path)
    assert isdir (path)

    matching_filenames = []
    file_matcher = re.compile (r'\.(?:jpe?g|gif|png)$', re.IGNORECASE)

    for dirpath, dirnames, filenames in os.walk (path):
        for name in filenames:
            file_path = join (dirpath, name)
            logging.debug ('Testing file_path %r', file_path)
            if file_matcher.search (file_path):
                matching_filenames.append (file_path)
            else:
                logging.info ('Ignoring non-image file %r', path)

    total_count = len (matching_filenames)
    for index, file_path in enumerate (matching_filenames):
        upload_file (file_path, index + 1, total_count)

def run_upload (options, paths):
    upload_file = make_upload_file (**options)

    for arg in paths:
        path = abspath (arg)
        if isdir (path):
            upload_directory (path, upload_file)
        elif isfile (path):
```

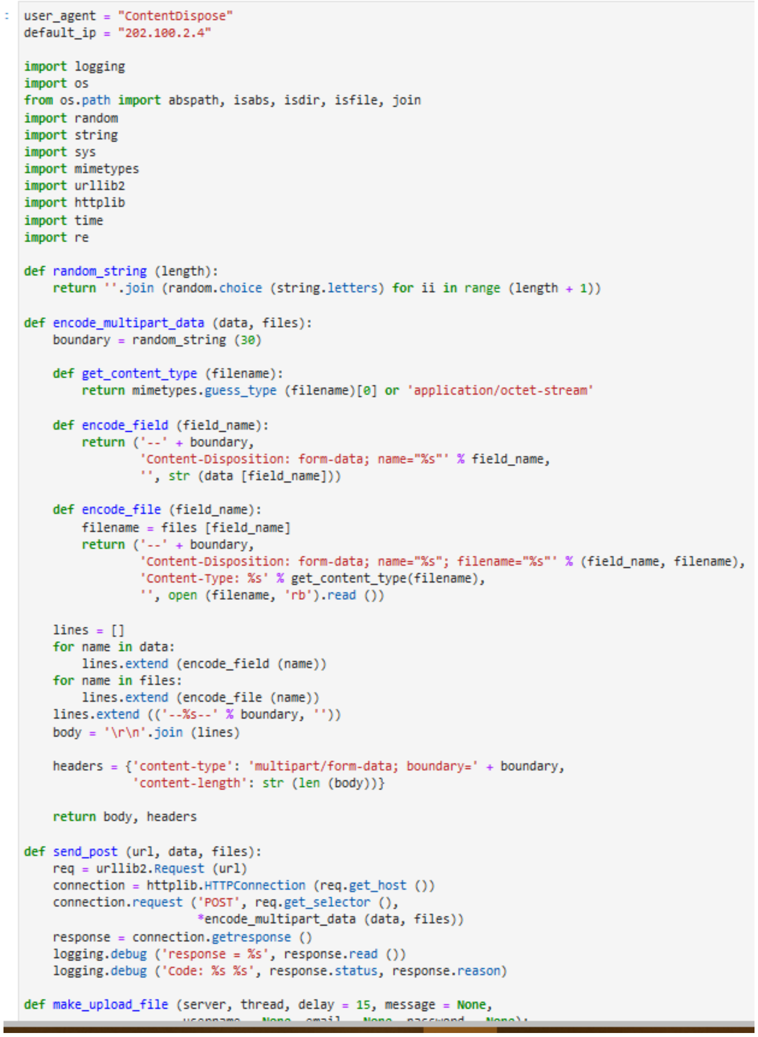
Figure: 11.2 (a) Package Creating Code Snippet

Creating and Exploring Packages

To tell Python that a particular directory is a package, we create a file named __init__.py inside it and then it is considered as a package and we may create other modules and sub-packages within it. This __init__.py file can be left blank or can be coded with the initialization code for the package.

To create a package in Python, we need to follow these three simple steps:

1. First, we create a directory and give it a package name, preferably related to its operation.
2. Then we put the classes and the required functions in it.
3. Finally we create an __init__.py file inside the directory, to let Python know that the directory is a package.



```
user_agent = "ContentDispose"
default_ip = "202.100.2.4"

import logging
import os
from os.path import abspath, isabs, isdir, isfile, join
import random
import string
import sys
import mimetypes
import urllib2
import httplib
import time
import re

def random_string (length):
    return ''.join (random.choice (string.letters) for ii in range (length + 1))

def encode_multipart_data (data, files):
    boundary = random_string (30)

    def get_content_type (filename):
        return mimetypes.guess_type (filename)[0] or 'application/octet-stream'

    def encode_field (field_name):
        return ('--' + boundary,
                'Content-Disposition: form-data; name="%s"' % field_name,
                '', str (data [field_name]))

    def encode_file (field_name):
        filename = files [field_name]
        return ('--' + boundary,
                'Content-Disposition: form-data; name="%s"; filename="%s"' % (field_name, filename),
                'Content-Type: %s' % get_content_type(filename),
                '', open (filename, 'rb').read ())

    lines = []
    for name in data:
        lines.extend (encode_field (name))
    for name in files:
        lines.extend (encode_file (name))
    lines.extend (('--%s--' % boundary, ''))
    body = '\r\n'.join (lines)

    headers = {'content-type': 'multipart/form-data; boundary=' + boundary,
               'content-length': str (len (body))}

    return body, headers

def send_post (url, data, files):
    req = urllib2.Request (url)
    connection = httplib.HTTPConnection (req.get_host ())
    connection.request ('POST', req.get_selector (),
                        *encode_multipart_data (data, files))
    response = connection.getresponse ()
    logging.debug ('response = %s', response.read ())
    logging.debug ('Code: %s %s', response.status, response.reason)

def make_upload_file (server, thread, delay = 15, message = None,
                      username = None, email = None, password = None):
```

Figure: 11.2 (b) Package Creating Code Snippet

12. INSTALLATION GUIDE WITH DETAILS

12.1 Installation (Tools: Anaconda- Spyder, Jupyter)

- Download the Anaconda installer.

- Optional: Verify data integrity with MD5 or SHA-256. More info on hashes

- Double click the installer to launch.

Note
To prevent permission errors, do not launch the installer from the Favorites folder.

Note

If you encounter issues during installation, temporarily disable your anti-virus software during install, then re-enable it after the installation concludes. If you installed for all users, uninstall Anaconda and re-install it for your user only and try again.

- Click Next.

- Read the licensing terms and click “I Agree”.

- Select an install for “Just Me” unless you’re installing for all users (which requires Windows Administrator privileges) and click Next.

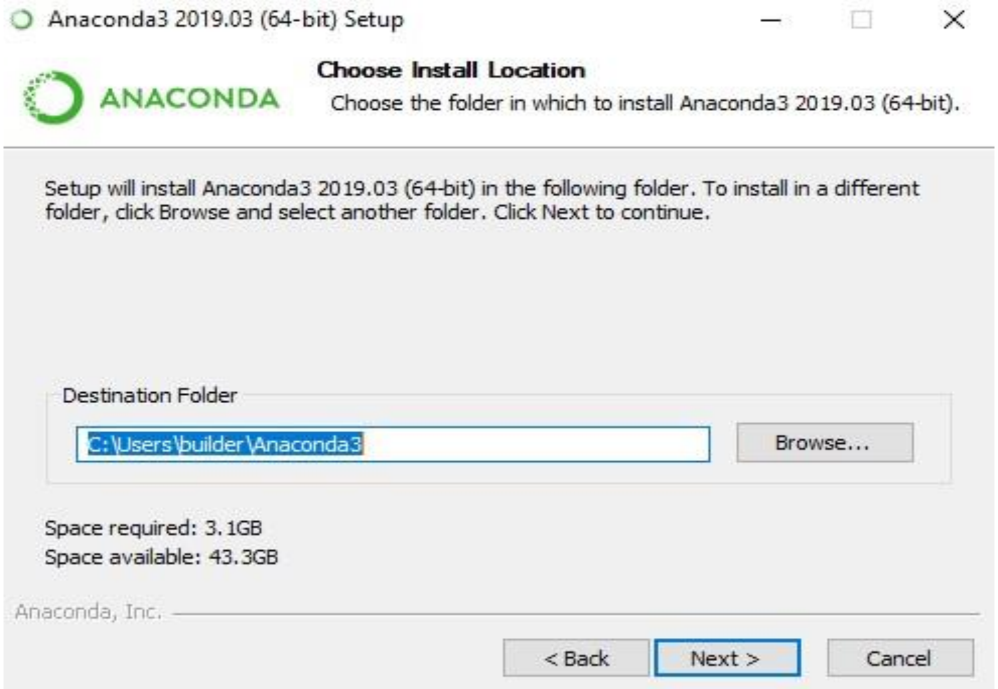
- Select a destination folder to install Anaconda and click the Next button. See FAQ.

Note

Install Anaconda to a directory path that does not contain spaces or Unicode characters.

Note

Do not install as Administrator unless admin privileges are required.



Anaconda3 2019.03 (64-bit) Setup

Choose Install Location

Choose the folder in which to install Anaconda3 2019.03 (64-bit).

Setup will install Anaconda3 2019.03 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

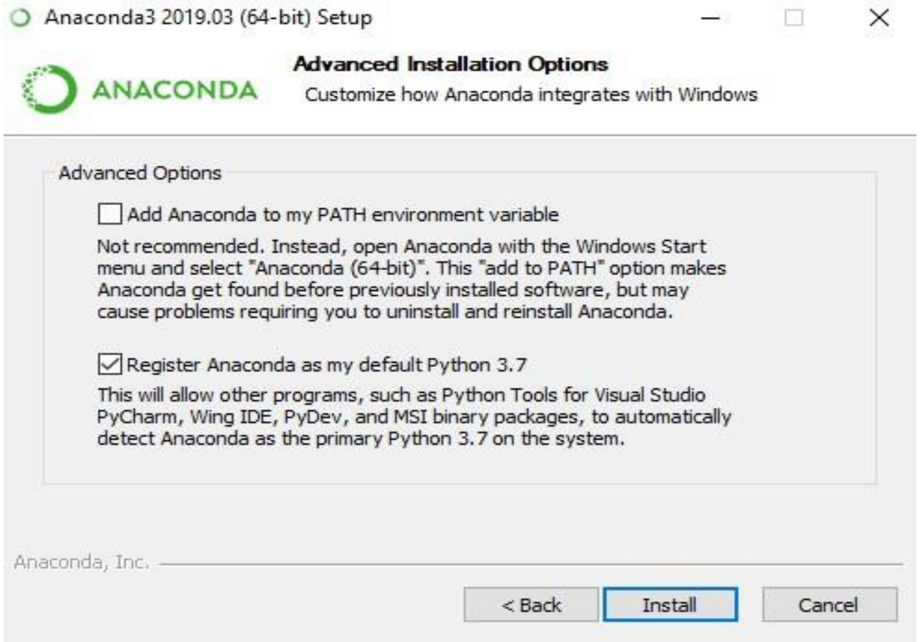
Destination Folder

C:\Users\builder\Anaconda3

Space required: 3.1GB
Space available: 43.3GB

Figure 12.1(a)

Choose whether to add Anaconda to your PATH environment variable. We recommend not adding Anaconda to the PATH environment variable, since this can interfere with other software. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.



Anaconda3 2019.03 (64-bit) Setup

Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

□ Add Anaconda to my PATH environment variable

Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda as my default Python 3.7

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.7 on the system.

Figure 12.1(b)

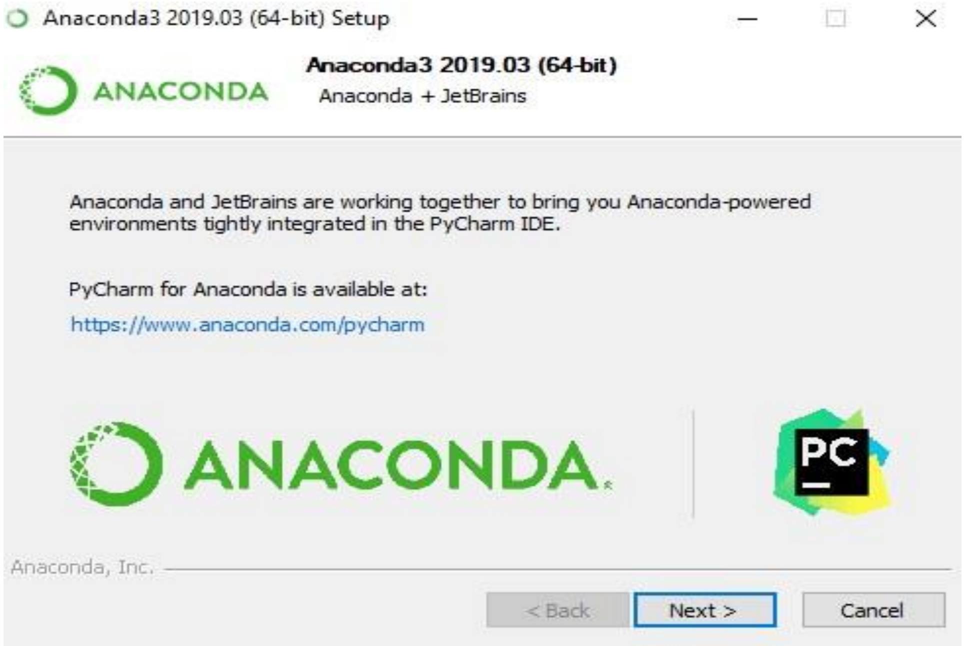
- Choose whether to register Anaconda as your default Python. Unless you plan on installing and running multiple versions of Anaconda, or multiple versions of Python, accept the default and leave this box checked.

- Click the Install button. If you want to watch the packages Anaconda is installing, click Show Details.

- Click the Next button.

Optional:

To install PyCharm for Anaconda, click on the link to https://www.anaconda.com/pycharm.



Anaconda3 2019.03 (64-bit) Setup

Anaconda3 2019.03 (64-bit)

Anaconda + JetBrains

Anaconda and JetBrains are working together to bring you Anaconda-powered environments tightly integrated in the PyCharm IDE.

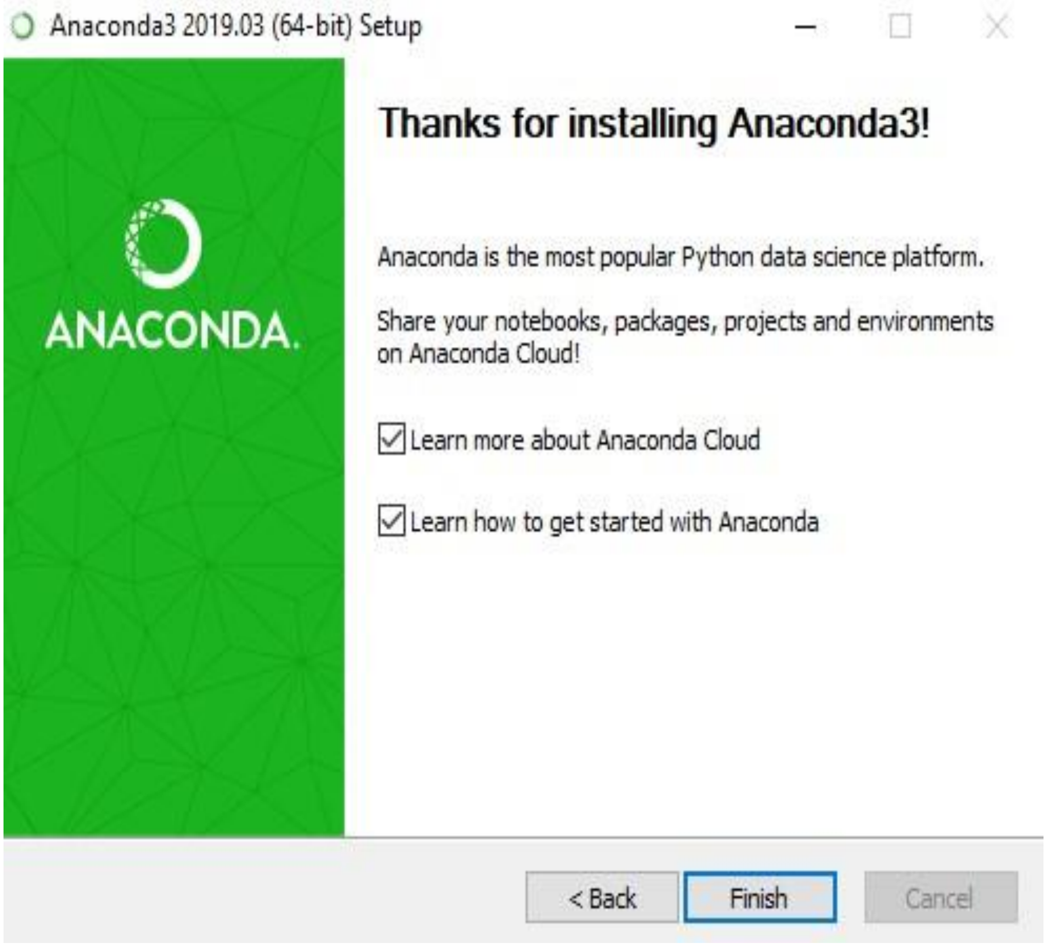
PyCharm for Anaconda is available at:

https://www.anaconda.com/pycharm

Figure 12.1(c)

Or to install Anaconda without PyCharm, click the Next button.

After a successful installation you will see the “Thanks for installing Anaconda” dialog box:



Anaconda3 2019.03 (64-bit) Setup

Thanks for installing Anaconda3!

Anaconda is the most popular Python data science platform.

Share your notebooks, packages, projects and environments on Anaconda Cloud!

☑ Learn more about Anaconda Cloud

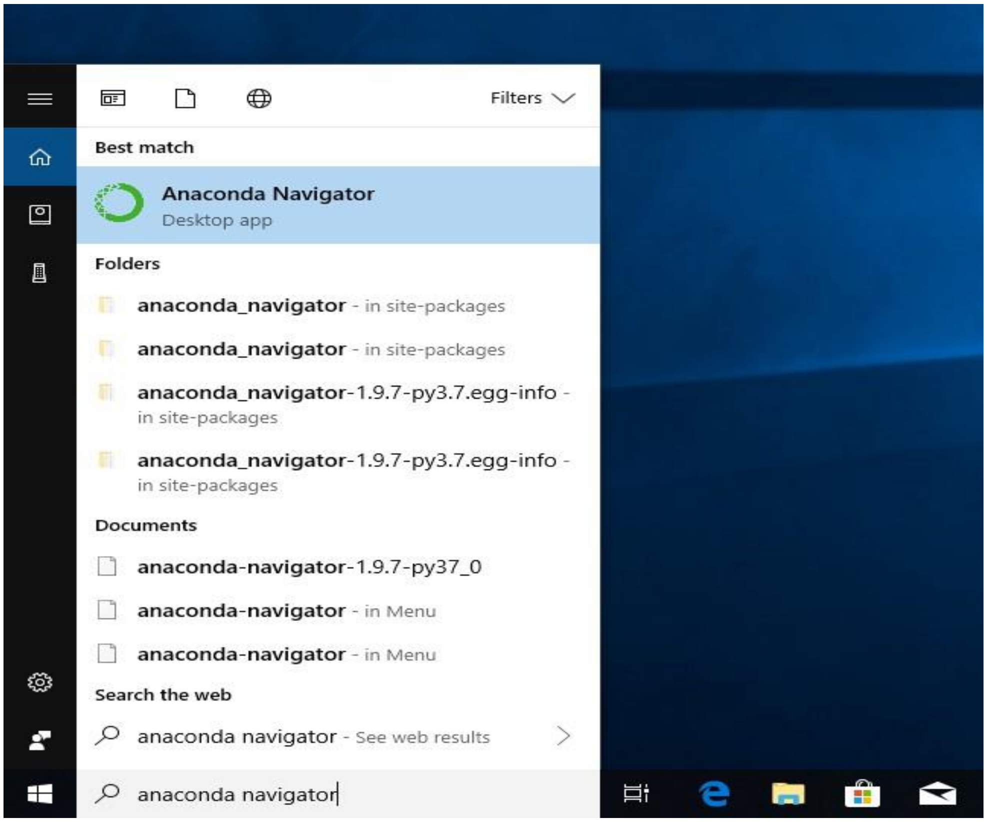
☑ Learn how to get started with Anaconda

Figure 12.1(d)

If you wish to read more about Anaconda Cloud and how to get started with Anaconda, check the boxes “Learn more about Anaconda Cloud” and “Learn how to get started with Anaconda”. Click the Finish button.

After your install is complete, verify it by opening Anaconda Navigator, a program that is included with Anaconda: from your Windows Start menu, select the shortcut Anaconda Navigator from the recently added or by typing “Anaconda Navigator”. If Navigator opens, you have successfully installed Anaconda.

If not, check that you completed each step above, then see our Help page.



anaconda_navigator - in site-packages

anaconda_navigator - in site-packages

anaconda_navigator-1.9.7-py3.7.egg-info - in site-packages

anaconda_navigator-1.9.7-py3.7.egg-info - in site-packages

anaconda-navigator-1.9.7-py37_0

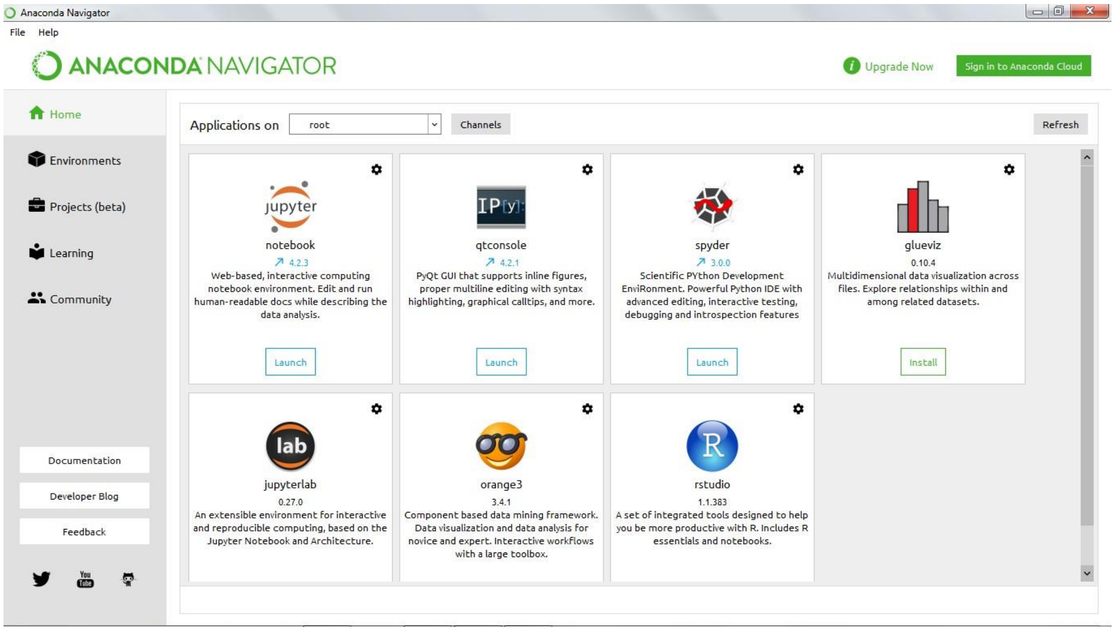
anaconda-navigator - in Menu

anaconda-navigator - in Menu

anaconda navigator - See web results

anaconda navigator

Figure 12.1(e)



notebook
4.2.3
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

qtconsole
4.2.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch

spyder
3.0.0
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch

glueviz
0.10.4
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

jupyterlab
0.27.0
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

orange3
3.4.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

rstudio
1.1.383
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Figure 12.1(f)

12.2 Installation (Python Libraries)

NumPy: conda install -c anaconda numpy

NumPy is the core library for scientific computing in Python. It provides a highperformance multidimensional array object, and tools for working with these arrays. A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multidimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy is licensed under the BSD license, enabling reuse with few restrictions.

SciPy: pip install scipy

SciPy is a library that uses NumPy for more mathematical functions. SciPy uses NumPy arrays as the basic data structure, and comes with modules for various commonly used tasks in scientific programming, including linear algebra, integration (calculus), ordinary differential equation solving, and signal processing.

Pandas: conda install pandas

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in

Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal.

pandas is well suited for many different kinds of data:

- Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet
- Ordered and unordered (not necessarily fixed-frequency) time series data.
- Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
- Any other form of observational / statistical data sets. The data actually need not be labeled at all to be placed into a pandas data structure

The two primary data structures of pandas, Series (1-dimensional) and DataFrame (2dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering. For R users, DataFrame provides everything that R's data.frame provides and much more. Pandas is built on top of NumPy and is intended to integrate well within a scientific computing environment with many other 3rd party libraries.

Here are just a few of the things that pandas does well:

- Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data
- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
- Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let Series, DataFrame, etc. automatically align the data for you in computations
- Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data
- Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into DataFrame objects
- Intelligent label-based slicing, fancy indexing, and subsetting of large data sets
- Intuitive merging and joining data sets
- Flexible reshaping and pivoting of data sets
    - Hierarchical labeling of axes (possible to have multiple labels per tick)
    - Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving / loading data from the ultrafast HDF5 format

- Time series-specific functionality: date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging, etc.

scikit-learn: conda install scikit-learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

- NumPy: Base n-dimensional array package
- SciPy: Fundamental library for scientific computing
- Matplotlib: Comprehensive 2D/3D plotting
- IPython: Enhanced interactive console
- Sympy: Symbolic mathematics
- Pandas: Data structures and analysis

Extensions or modules for SciPy care conventionally named SciKits. As such, the module provides learning algorithms and is named scikit-learn.

The vision for the library is a level of robustness and support required for use in production systems. This means a deep focus on concerns such as easy of use, code quality, collaboration, documentation and performance.

Although the interface is Python, c-libraries are leverage for performance such as numpy for arrays and matrix operations, LAPACK, LibSVM and the careful use of cython.

Some popular groups of models provided by scikit-learn include:

- Clustering: for grouping unlabeled data such as KMeans.
- Cross Validation: for estimating the performance of supervised models on unseen data.
- Datasets: for test datasets and for generating datasets with specific properties for investigating model behavior.

- Dimensionality Reduction: for reducing the number of attributes in data for summarization, visualization and feature selection such as Principal component analysis.
- Ensemble methods: for combining the predictions of multiple supervised models.
- Feature extraction: for defining attributes in image and text data.
- Feature selection: for identifying meaningful attributes from which to create supervised models.
- Parameter Tuning: for getting the most out of supervised models.
- Manifold Learning: For summarizing and depicting complex multi-dimensional data.
- Supervised Models: a vast array not limited to generalized linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines and decision trees.

sklearn.model_selection.train_test_split (*arrays, **options)

Split arrays or matrices into random train and test subsets

Quick utility that wraps input validation and next(ShuffleSplit().split(X, y)) and application to input data into a single call for splitting (and optionally subsampling) data in a oneliner.

sklearn.preprocessing

The sklearn.preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators.

In general, learning algorithms benefit from standardization of the data set. If some outliers are present in the set, robust scalers or transformers are more appropriate. The behaviors of the different scalers, transformers, and normalizers on a dataset containing marginal outliers is highlighted in Compare the effect of different scalers on data with outliers.
sklearn.metrics.f1_score

Compute the F1 score, also known as balanced F-score or F-measure

The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

F1 = 2 * (precision * recall) / (precision + recall)

In the multi-class and multi-label case, this is the average of the F1 score of each class with weighting depending on the average parameter.

Matplotlib: pip install matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter, has an active development community and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012, and further joined by Thomas Caswell.

As of 23 June 2017, matplotlib 2.0.x supports Python versions 2.7 through 3.6. Matplotlib 1.2 is the first version of matplotlib to support Python 3.x. Matplotlib 1.4 is the last version of Matplotlib to support Python 2.6.

Matplotlib has pledged to not support Python 2 past 2020 by signing the Python 3 Statement.

Seaborn: pip install seaborn

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.

Here is some of the functionality that seaborn offers:

- A dataset-oriented API for examining relationships between multiple variables
- Specialized support for using categorical variables to show observations or aggregate statistics

- Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
- Automatic estimation and plotting of linear regression models for different kinds dependent variables
- Convenient views onto the overall structure of complex datasets
- High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations
- Concise control over matplotlib figure styling with several built-in themes
- Tools for choosing color palettes that faithfully reveal patterns in your data

Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on DataFrame and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

seaborn.pairplot

Plot pairwise relationships in a dataset.

By default, this function will create a grid of Axes such that each variable in data will by shared in the y-axis across a single row and in the x-axis across a single column. The diagonal Axes are treated differently, drawing a plot to show the univariate distribution of the data for the variable in that column.

It is also possible to show a subset of variables or plot different variables on the rows and columns.

This is a high-level interface for PairGrid that is intended to make it easy to draw a few common styles. You should use PairGrid directly if you need more flexibility.

Beautiful Soup: pip install beautifulsoup

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

- Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree: a toolkit for dissecting a document and extracting what you need. It doesn't take much code to write an application

- Beautiful Soup automatically converts incoming documents to Unicode and outgoing documents to UTF-8. You don't have to think about encodings, unless the document doesn't specify an encoding and Beautiful Soup can't detect one. Then you just have to specify the original encoding.
- Beautiful Soup sits on top of popular Python parsers like lxml and html5lib, allowing you to try out different parsing strategies or trade speed for flexibility.

Beautiful Soup parses anything you give it, and does the tree traversal stuff for you. You can tell it "Find all the links", or "Find all the links of class externalLink", or "Find all the links whose urls match "foo.com", or "Find the table heading that's got bold text, then give me that text."

Valuable data that was once locked up in poorly-designed websites is now within your reach. Projects that would have taken hours take only minutes with Beautiful Soup.

Mlbox: pip install mlbox

MLBox is a powerful Automated Machine Learning python library. It provides the following features:

- Fast reading and distributed data preprocessing/cleaning/formatting.
- Highly robust feature selection and leak detection.
- Accurate hyper-parameter optimization in high-dimensional space.
- State-of-the art predictive models for classification and regression (Deep Learning, Stacking, LightGBM).
- Prediction with models interpretation.

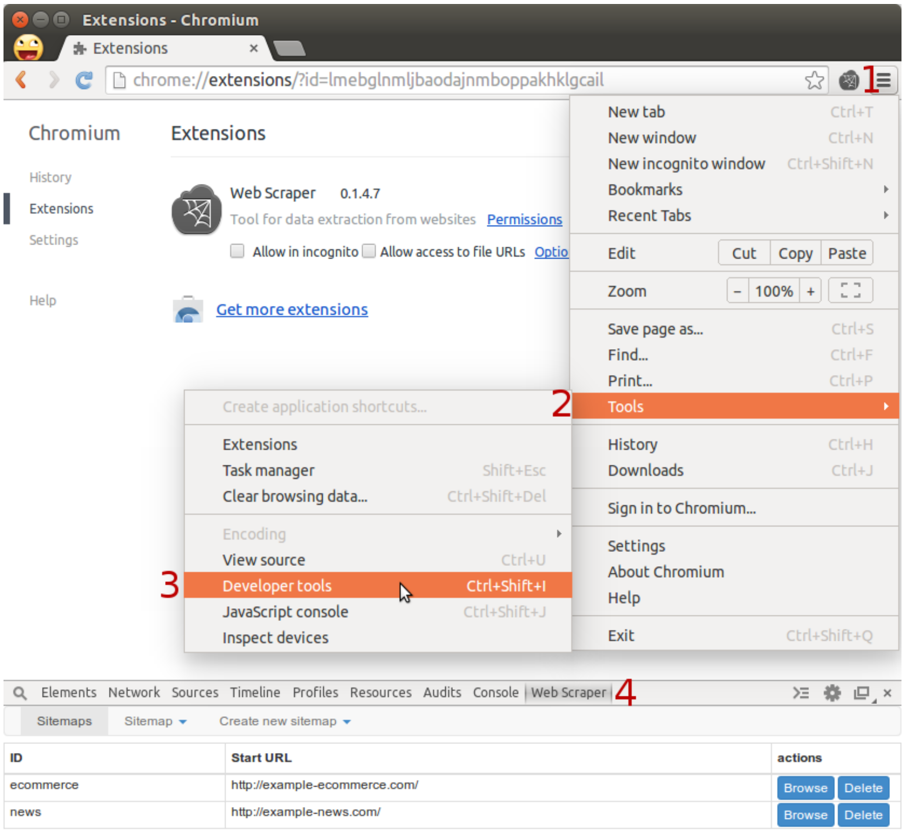
MLBox focuses on the below three points in particular in comparison to the other libraries:

- Drift Identification – A method to make the distribution of train data similar to the test data.
- Entity Embedding – A categorical features encoding technique inspired from word2vec.
- Hyperparameter Optimization

12.3 Installation and Usage (Tools: WebScraper)

12.3.1 Installation

You can install the extension from Chrome store. After installing it you should restart chrome to make sure the extension is fully loaded. If you don't want to restart Chrome then use the extension only in tabs that are created after installing it.



ID	Start URL	actions
ecommerce	http://example-ecommerce.com/	Browse Delete
news	http://example-news.com/	Browse Delete

Figure 12.3(a) Web Scraper

12.3.2 Requirements

The extension requires Chrome 49+. There are no OS limitations.

12.3.3 Open Web Scraper

Web Scraper is integrated into chrome Developer tools. Figure 1 shows how you can open it. You can also use these shortcuts to open Developer tools. After opening Developer tools open Web Scraper tab.

Shortcuts:

windows, linux: `Ctrl+Shift+I, f12, open Tools / Developer tools`

mac `Cmd+Opt+I, open Tools / Developer tools`

12.3.4 Scraping a site

Open the site that you want to scrape.

Create Sitemap

The first thing you need to do when creating a *sitemap* is specifying the start url. This is the url from which the scraping will start. You can also specify multiple start urls if the scraping should start from multiple places. For example if you want to scrape multiple search results then you could create a separate start url for each search result.

Specify multiple urls with ranges

In cases where a site uses numbering in pages URLs it is much simpler to create a range start url than creating *Link selectors* that would navigate the site. To specify a range url replace the numeric part of start url with a range definition - `[1-100]`. If the site uses zero padding in urls then add zero padding to the range definition - `[001-100]`. If you want to skip some urls then you can also specify incremental like this `[0-100:10]`.

Use range url like this `http://example.com/page/[1-3]` for links like these:

- http://example.com/page/1
- http://example.com/page/2
- http://example.com/page/3

Use range url with zero padding like this `http://example.com/page/[001-100]` for links like these:

- http://example.com/page/001
- http://example.com/page/002

Use range url with increment like this http://example.com/page/[0-100:10] for links like these:

- http://example.com/page/0
- http://example.com/page/10
- http://example.com/page/20

Create selectors

After you have created the *sitemap* you can add selectors to it. In the *Selectors* panel you can add new selectors, modify them and navigate the selector tree. The selectors can be added in a tree type structure. The web scraper will execute the selectors in the order how they are organized in the tree structure. For example there is a news site and you want to scrape all articles whose links are available on the first page. In image 1 you can see this example site.

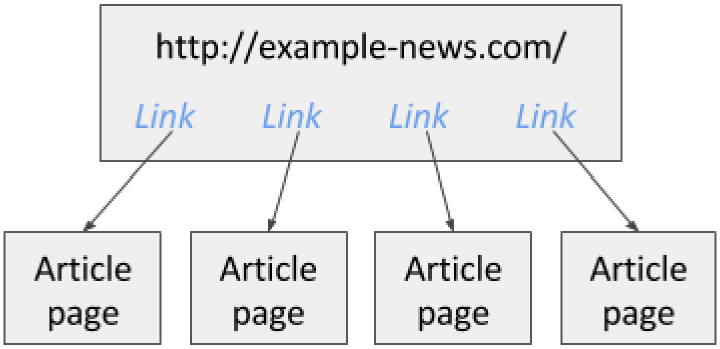


Fig. 12.3.4(a): News site

To scrape this site you can create a *Link selector* which will extract all article links in the first page. Then as a child selector you can add a *Text selector* that will extract articles from the article pages that the *Link selector* found links to. Image below illustrates how the *sitemap* should be built for the news site.

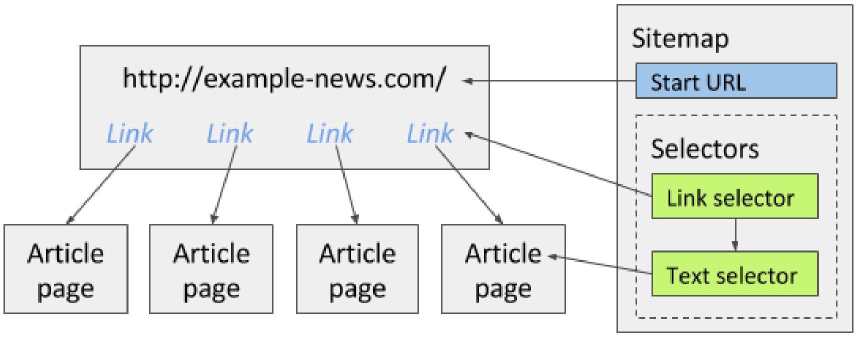


Fig. 12.3.4 (b): News site sitemap

Note that when creating selectors use Element preview and Data preview features to ensure that you have selected the correct elements with the correct data.

More information about selector tree building is available in selector documentation. You should atleast read about these core selectors:

- Text selector
- Link selector

Inspect selector tree

After you have created selectors for the *sitemap* you can inspect the tree structure of selectors in the Selector graph panel. Image below shows an example selector graph.

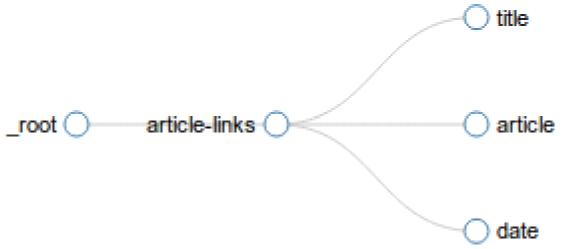


Fig. 12.3.4(c): News site selector graph

Scrape the site

After you have created selectors for the *sitemap* you can start scraping. Open *Scrape* panel and start scraping. A new popup window will open in which the scraper will load pages and extract data from them. After the scraping is done the popup window will close and you will be notified with a popup message. You can view the scraped data by opening *Browse* panel and export it by opening the *Export data as CSV* panel.

13. ANNEXURE

13.1 References

1. https://talentedge.in/blog/scope-future-data-analytics-india/
2. http://customerthink.com/accelerate-sales-with-machine-learning-andartificial-intelligence-what-you-need-to-know-to-get-started/
3. https://www.geeksforgeeks.org/regression-classification-supervised-machinelearning/
4. https://ketakirk.wordpress.com/2016/04/03/an-end-to-end-data-analysisworkflow/
5. https://krazytech.com/projects/sample-software-requirements-specificationsrsreport-airline-database
6. https://www.numpy.org/
7. https://matplotlib.org/users/pyplot_tutorial.html
8. https://www.scipy.org/
9. https://github.com/scikit-
10. https://towardsdatascience.com/5-data-science-project-every-e-commerce-company-should-do-8746c5ab4604
11. https://www.promptcloud.com/blog/how-to-scrape-data-with-web-scraper-chrome/
12. https://towardsdatascience.com/a-short-introduction-to-model-selection-bb1bb9c73376
13. https://www.datasciencecentral.com/profiles/blogs/how-the-economics-of-data-science-is-creating-new-sources-of
14. https://www.researchgate.net/publication/308960488_A_FEASIBILITY_STUDY_ON_BIG_DATA_INTEGRATION_AND_ITS_METHODOLOGIES_FOR_HADOOP_TECHNIQUES_USING_MAP_REDUCE_MODEL
15. https://en.wikipedia.org/wiki/Web_scraping
16. https://www.webscraper.io/tutorials
17. http://www.arithaconsulting.com/development-projects-in-data-science/
18. https://towardsdatascience.com/data-science-project-flow-for-startups-282a93d4508d
19. https://www.digitalocean.com/community/tutorials/how-to-write-modules-in-python-
20. https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f
21. https://www.draw.io/
22. https://medium.com/@jameschen_78678/data-science-x-project-planning-9cb0a2c3cfa7
23. https://github.com/rdpeng/courses/tree/master/05_ReproducibleResearch/Checklist
24. https://www.scikit-yb.org/en/latest/api/classifier/classification_report.html
25. https://buckwoody.wordpress.com/2017/08/17/a-data-science-microsoft-project-template-you-can-use-in-your-solutions/

13.2 About The Organization (Company Information)

Fusion Informatics is a Leading ISO Certified 9001:2015 Software development service provider in India, UAE and USA that combines creativity with utility. We offer a plenty of services like Enterprise Mobile App Development, Internet of Things (IoT) Development, Artificial Intelligence Development, Blockchain Development, Cloud
Solutions, Smart Device Development, iPhone app Development & Android app Development, Business Process Solutions, and other IT related designs. As a best mobile app development company in India (Bangalore, Ahmedabad, Mumbai and Delhi), we constantly strive to design successful inclinations for startups to fortune 500 companies that intend to deliver an exceptional level of achievement with no compromising the quality.

We do consider quality is most prominent in the competition. Therefore, we modify ourselves by presenting what we assure. Although we attempt, advanced technologies at unbelievably handsome packages, High quality is our weapon we never negotiate on. The development team at Fusion Informatics develops scalable and reliable mobile apps as per your custom demands that are extremely complex and operative for both platforms on Android App Development and iPhone App Development.

What We Do:

- Enterprise Mobile App Development: Bringing mobility within an enterprise is what we do, by helping stakeholder’s process business tasks through mobile or cloud.
- Internet of Things (IoT) App Development: Our IoT app development, applied for all devices through automation and allowing them to operate independently, to accomplish human tasks.
- Artificial Intelligence Development: AI services for boosting sales, enhancing customer experiences, delivering innovative marketing solutions and much more.
- Blockchain Development: The Fusion Informatics workshop has access to state of the art technology and the best of the resources need for Blockchain development.
- Cloud Solutions: Clients save big time on infrastructural costs, by letting them access shared solutions on a cloud, accessible anywhere anytime.
- Smart Device Development: Apps we create, help to communicate with smart devices, making them function as you want, and do real smart processing.
- App Design & Development: Are you looking for a tailor-made web application or a customized mobile app for your business need? Create an identity to expand your business.
- Business Process Solutions: We help businesses work upon operational hurdles, and get their business processes atomized, achieving strategic goals over time.
- Data Science: Data science applications allows companies to create smart decisions based on various aspects of Big data.

13.3 About College (U. V. Patel College of Engineering, Ganpat University)

Ganpat University as a well reputed State Private University established in 2005 through the State Legislative act no 19 of 2005, Government of Gujarat and recognized by the UGC under the section 2(f) of the UGC Act, 1956 having campus spread over more than 300 acres of land with world class infrastructure and more than 10,000 students on campus. The University offers Diplomas, Under Graduate, Post – Graduate and Research Programs under the Faculties of Engineering and Technology, Pharmacy, Management, Computer Applications, Sciences, Education, Humanities and Social Science and Human Potential Development. Ganpat University and the township of Ganpat Vidyanagar, a high-tech education campus is a joint initiatives; purely for philanthropy; of a large number of industrialists and technocrats, noble farmers and affluent businessmen; having a mission of "Social Upliftment through Education"

Ganpat University-U. V. Patel College of Engineering (GNU-UVPCE) is situated in Ganpat Vidyanagar campus. It was established in September-1997 and it is one of the constituent colleges of Ganpat University with a view of educating and training young talented students of Gujarat in the field of Engineering and Technology to meet the needs of Industries in Gujarat and across globe.

The College is named after Shri Ugarchandbhai Varanasibhai Patel, a leading industrialist of Gujarat, for his generous support. It is a self-financed institute approved by All India Council for Technical Education (AICTE), New Delhi and the Commissionerate of Technical Education, Government of Gujarat.

The College is spread over 25 acres of land and is a part of Ganpat Vidyanagar Campus. It has six ultra modern buildings of architectural splendor, class rooms, tutorial rooms, seminar halls, offices, drawing hall, workshop, library, well equipped departmental laboratories and several computer laboratories with internet connectivity through 10Gbps Fiber link, satellite link education center with two-way audio and one-way video link.

The Institute, at present, offers ten undergraduate programmes, ten postgraduate programmes and Ph.D. programme.

Placement plays key role in shaping the future of the students and keeping this in mind the institute has created healthy relations with the prominent industries also. This in turn is reciprocally advantageous. The industries gets a chance to exploit the resources of the institute for their R & D work and in return extend every possible help to the institute. As part of this initiative, Incubation Centre/Start-up activities have been developed.