**1. Overview**

1.1 **Introduction:** Accessories Management System Project is implemented for managing the dealer, customers, claims, Parts, Transactions and also This system also provides an the company to check or analyze the part is in warranty or not and also provides company to check the Monthly report of Dealers and parts that has been failed &Predict the parts that may fail according to the location, mileage and age. Company to understand Monthly report of top ten failed parts with the help of this system they can check Quality of the parts, Pattern of failure mostly reported failed part in the quarter/year. Predict the parts that may fail according to the location, mileage and age: company can analyzing the historical data, we need to predict the parts that can be expected to fail according to the age, km driven and location of the vehicle. This data helps the company to understand the manufacturing requirements of the part i.e. which part is to be manufactured in what quantity so as to avoid unavailability of the parts when claim is raised for that part. To determine whether the defective part is to be replaced or repaired. The model was developed to decide whether the failed part has to be repaired or replaced with the new one. For this, we had the last claimed data of the company. As this is the binary classification problem, Decision tree and Random Forest algorithms are used to build the model. Customer Retention Model. This model helps the company to understand the service provided by the dealers is satisfactory or not. The model analyses the performance of the dealers by considering the customers feedback, churn rate of that location, preference to the other locations by the customers, change of base location etc. The model helps the company to take some regulatory decisions based the performance of dealer. It also helps the company to decide, whether to introduce new schemes, offers etc. to the affected location or to change the dealer. Processing of variety of data. The data were coming from various locations like US, Canada, China, and India in huge amount with varying data-types it was required to club the data and then process it further for the analyzing. Processing large volume of data also caused memory issues which were then handled by memory management.

Accessories Management System is an application using [Google Co-laboratory](https://colab.research.google.com/notebooks/) in Python with data science. The feature of this application includes On the sale of the vehicle company provides one year base warranty on its parts and also provides optional one year extended warranty on additional cost. If the customer has not opted for the extended warranty option then after the one year in case of any failure of any part, the warranty is said to be expired and all the expenses has to be incurred by the customer itself and company will not take any responsibility for the same. On the other hand, if the vehicle is secured by extended warranty assurance, then after the one year from the sale i.e. after the base warranty period, if any failure happens then the affected part/parts is/are either repaired or replaced under the warranty terms of the company and all the expenses will be done by the company. This whole system is built using Python with data science programming language with the help of [Google Co laboratory](https://colab.research.google.com/notebooks/).

1.2 **Existing System**:  At present traders maintain their day to day company has to face many issue of any failure happens then the affected part/parts is/are either repaired or replaced under the warranty terms of the company and all the expenses will be done by the company. During this process it takes lots of time and dealer must wait till the process, only single system is used during this process.

.

* 1. Proposed System:

Modules: -

EDA:-Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations. (For this example pandas, numpy, matplotlib and seaborn) and loaded the data set.

Visualization of data: - Data visualization is the graphical representation of information and data. By using [visual elements like charts, graphs, and maps](https://www.tableau.com/learn/articles/data-visualization/glossary), data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

Grouping of datasheets: - function is used to split the data into groups based on some criteria. Pandas objects can be split on any of their axes. The abstract definition of grouping is to provide a mapping of labels to group names

The proposed system consists of following modules:

1.4 Feasibility Study:

ECONOMIC FEASIBILITY:-

TECHNICAL FEASIBILITY:-

OPERATIONAL FEASIBILITY:-

1.5 System Analysis:

- Hardware Requirements:

Processor : i3 processor

Memory : 4 GB RAM

Display : 14’’ LCD

Hard disk Drive : 120 GB

- Software Requirements : [Google Co laboratory](https://colab.research.google.com/notebooks/), Jupyter notebook

Operating System : Windows 7, 8.1,10.

1.5.1 **Introduction to Python with data science:** **Python** is a dynamic, object-oriented, high-level programming language that can be used for many kinds of software development. Python programming language was created by Guido van Rossum. Python got its name from the BBC comedy series “Monty Python’s Flying Circus”. Guido van Rossum needed a name that was short, unique, and slightly mysterious. Since he was a fan of the show he thought this name was great. Python is an interpreter, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. It supports multiple programming paradigms beyond object-oriented programming, such as procedural and functional programming. Python combines remarkable power with very clear syntax. It has interfaces to many system calls and libraries, as well as to various window systems, and is extensible in C or C++. It is also usable as an extension language for applications that need a programmable interface. Python is portable: it runs on many UNIX variants including Linux and macOS, and on Windows.

**Features of Python Programming language:**

At present, there are many programming languages, and everyone has its own unique and different feature. It is the feature which matters the most and helps planning of choosing the best programming language for a project. So, you should know about the Python Programming language before you decide which is better for you. For knowing a language, you should know the feature, below are the features that will clear your thoughts about why Python over R or any other roots.

1. Easy to code and Read
2. High-Level Programming Language
3. Portable
4. Expressive
5. Object Oriented
6. Free and Open Source
7. Interpreted
8. Extensible
9. Embeddable
10. Large and Standard Library
11. GUI programming
12. Dynamically Typed

1. Easy to code:   
Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, JavaScript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer-friendly language.

2. Free and Open Source:   
Since it is open-source; this means that source code is also available to the public. So you can download it as, use it as well as share it.

3. Object-Oriented Language:   
One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.

4. GUI Programming Support:   
Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python.  
PyQt5 is the most popular option for creating graphical apps with Python.

5. High-Level Language:   
Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6. Extensible feature:  
Python is a Extensible language. We can write us some Python code into C or C++ language and also we can compile that code in C/C++ language.

7. Python is Portable language:   
Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, UNIX, and Mac then we do not need to change it, we can run this code on any platform.

8. Python is Integrated language:  
Python is also an Integrated language because we can easily integrated python with other languages like c, c++, etc.

9. Interpreted Language:  
Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called byte code.

10. Large Standard Library  
Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers, etc.

11. Dynamically Typed Language:  
Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don’t need to specify the type of variable.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

# Applications for Python

Python is used in many application domains.

* The Python Package Index lists thousands of third party modules for Python.

# Web and Internet Development

Python offers many choices for web development:

* Frameworks such as Django and Pyramid.
* Micro-frameworks such as Flask and Bottle.
* Advanced content management systems such as [Plone](http://www.plone.org/) and Django CMS.

Python's standard library supports many Internet protocols:

* HTML and XML
* JSON
* E-mail processing.
* Support for FTP, IMAP, and other Internet protocols.
* Easy-to-use socket interface.

And the Package Index has yet more libraries:

* Requests, a powerful HTTP client library.
* Beautiful Soup, an HTML parser that can handle all sorts of oddball HTML.
* Feedparser for parsing RSS/Atom feeds.
* Paramiko, implementing the SSH2 protocol.
* Twisted Python, a framework for asynchronous network programming.
* Microsoft Foundation Classes through the win32 extensions

**Data science**

Data science is the study of data. Like biological sciences are a study of biology, physical sciences, it’s the study of physical reactions. Data is real, data has real properties, and we need to study them if we’re going to work on them. Data Science involves data and some signs.

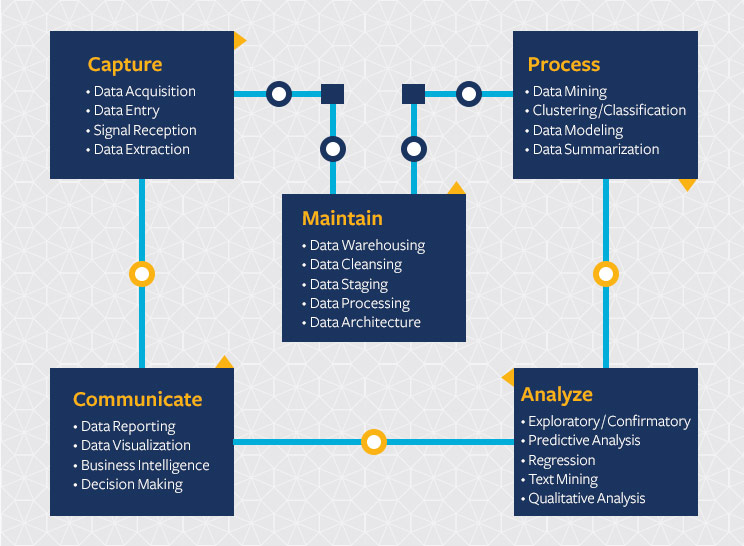
It is a process, not an event. It is the process of using data to understand too many different things, to understand the world. when you have a model or proposed explanation of a problem, and you try to validate that proposed explanation or model with your data.  
It is the skill of unfolding the insights and trends that are hiding (or abstract) behind data. It’s when you translate data into a story. So use storytelling to generate insight. And with these insights, you can make strategic choices for a company or an institution.

We can also define data science as a field which is about processes and systems to extract data of various forms and from various resources whether the data is unstructured or structured.

The definition and the name came up in the 1980s and 1990s when some professors, IT Professionals, scientist were looking into the statistics curriculum, and they thought it would be better to call it data science and then later on data analytics derived.

**Data science features and applications**

Data science continues to evolve as one of the most promising and in-demand career paths for skilled professionals. Today, successful data professionals understand that they must advance past the traditional skills of analyzing large amounts of data, data mining, and programming skills. In order to uncover useful intelligence for their organizations, data scientists must master the full spectrum of the data science life cycle and possess a level of flexibility and understanding to maximize returns at each phase of the process.

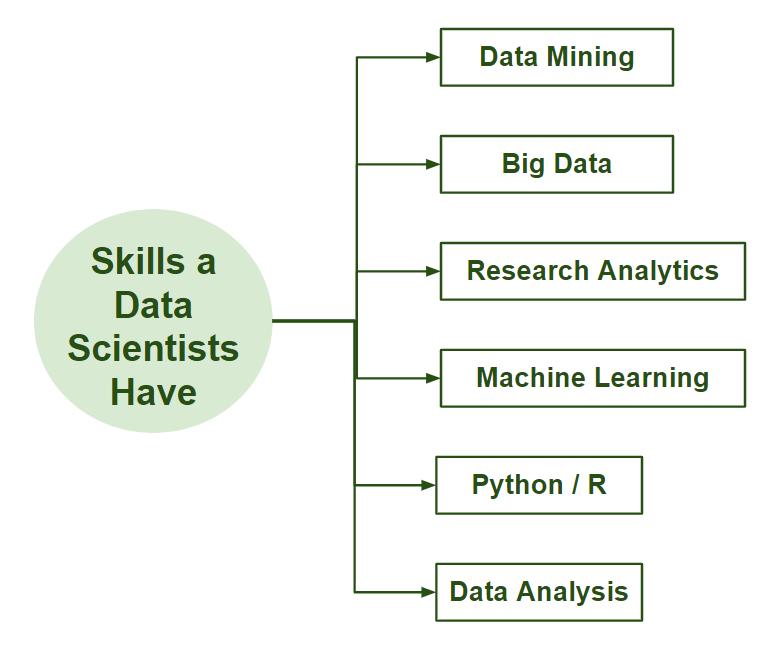


## ****Data Scientist****

There are several definitions available on Data Scientists. In simple words, a Data Scientist is one who practices the art of Data Science. The term “Data Scientist” has been coined after considering the fact that a Data Scientist draws a lot of information from the scientific fields and applications whether it is statistics or mathematics.

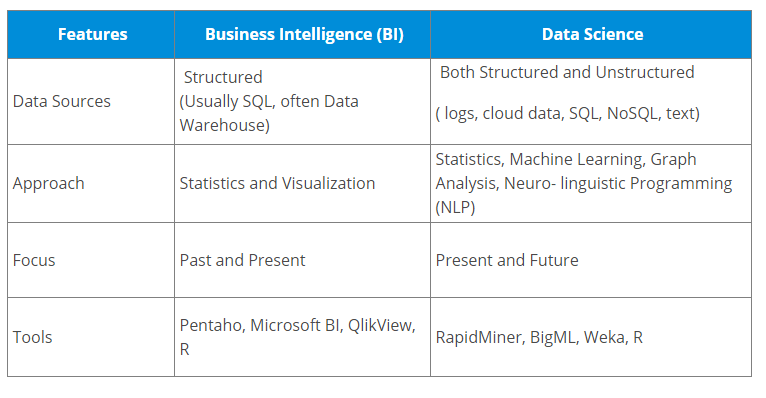
### ****What does a Data Scientist do?****

Data scientists are those who crack complex data problems with their strong expertise in certain scientific disciplines. They work with several elements related to mathematics, statistics, computer science, etc (though they may not be an expert in all these fields). They make a lot of use of the latest technologies in finding solutions and reaching conclusions that are crucial for an organization’s growth and development. Data Scientists present the data in a much more useful form as compared to the raw data available to them from structured as well as unstructured forms.



## Business Intelligence (BI) vs. Data Science

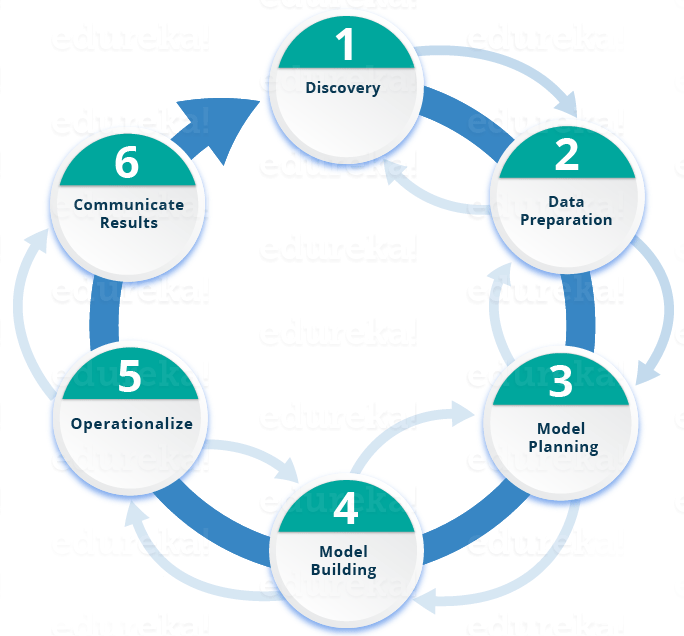
* Business Intelligence (BI) basically analyzes the previous data to find hindsight and insight to describe business trends. Here BI enables you to take data from external and internal sources, prepare it, run queries on it and create dashboards to answer questions like quarterly revenue analysis or business problems. BI can evaluate the impact of certain events in the near future.
* Data Science is a more forward-looking approach, an exploratory way with the focus on analyzing the past or current data and predicting the future outcomes with the aim of making informed decisions. It answers the open-ended questions as to “what” and “how” events occur.



A common mistake made in Data Science projects is rushing into data collection and analysis, without understanding the requirements or even framing the business problem properly. Therefore, it is very important for you to follow all the phases throughout the lifecycle of Data Science to ensure the smooth functioning of the project.

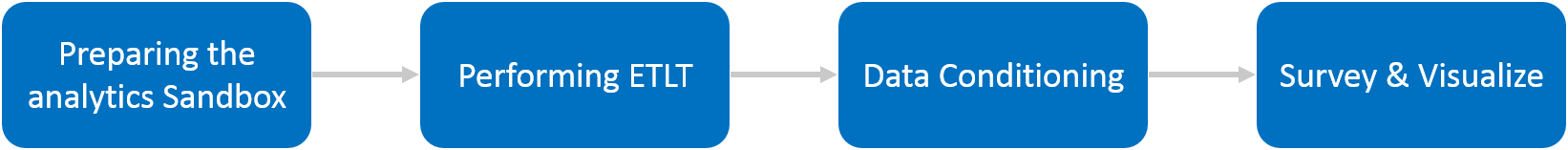
## Lifecycle of Data Science

Here is a brief overview of the main phases of the Data Science Lifecycle:



Phase 1—Discovery: Before you begin the project, it is important to understand the various specifications, requirements, priorities and required budget. You must possess the ability to ask the right questions. Here, you assess if you have the required resources present in terms of people, technology, time and data to support the project. In this phase, you also need to frame the business problem and formulate initial hypotheses (IH) to test.

 **Phase 2—Data preparation:**In this phase, you require analytical sandbox in which you can perform analytics for the entire duration of the project. You need to explore, preprocess and condition data prior to modeling. Further, you will perform ETLT (extract, transform, load and transform) to get data into the sandbox. Let’s have a look at the Statistical Analysis flow below.



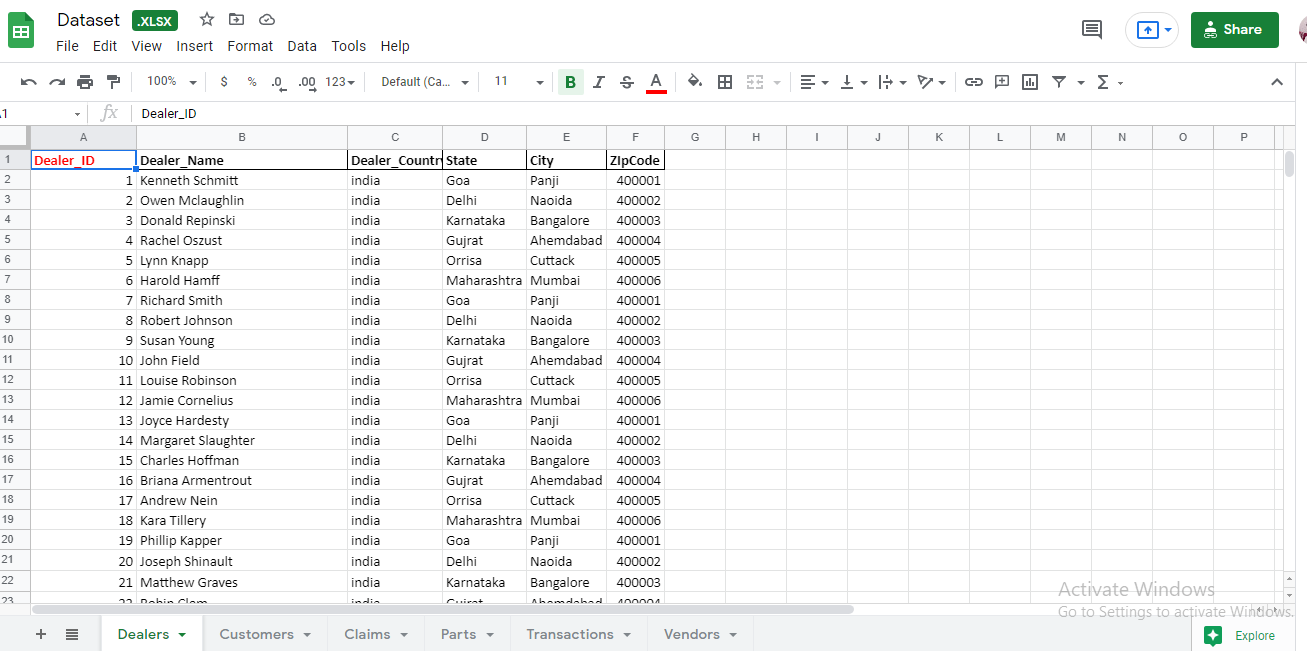
Phase 3—Model planning:  Here, you will determine the methods and techniques to draw the relationships between variables. These relationships will set the base for the algorithms which you will implement in the next phase. You will apply Exploratory Data Analytics (EDA) using various statistical formulas and visualization tools.

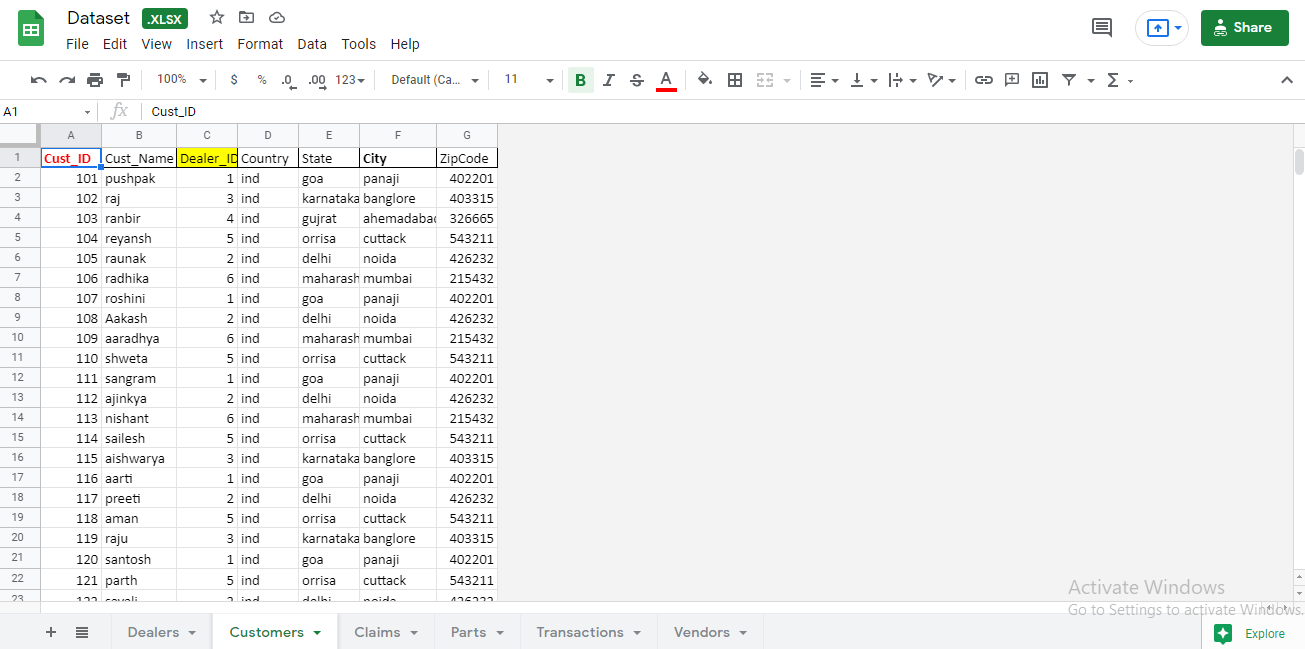
Phase 4—Model building: In this phase, you will develop datasets for training and testing purposes. Here you need to consider whether your existing tools will suffice for running the models or it will need a more robust environment (like fast and parallel processing).

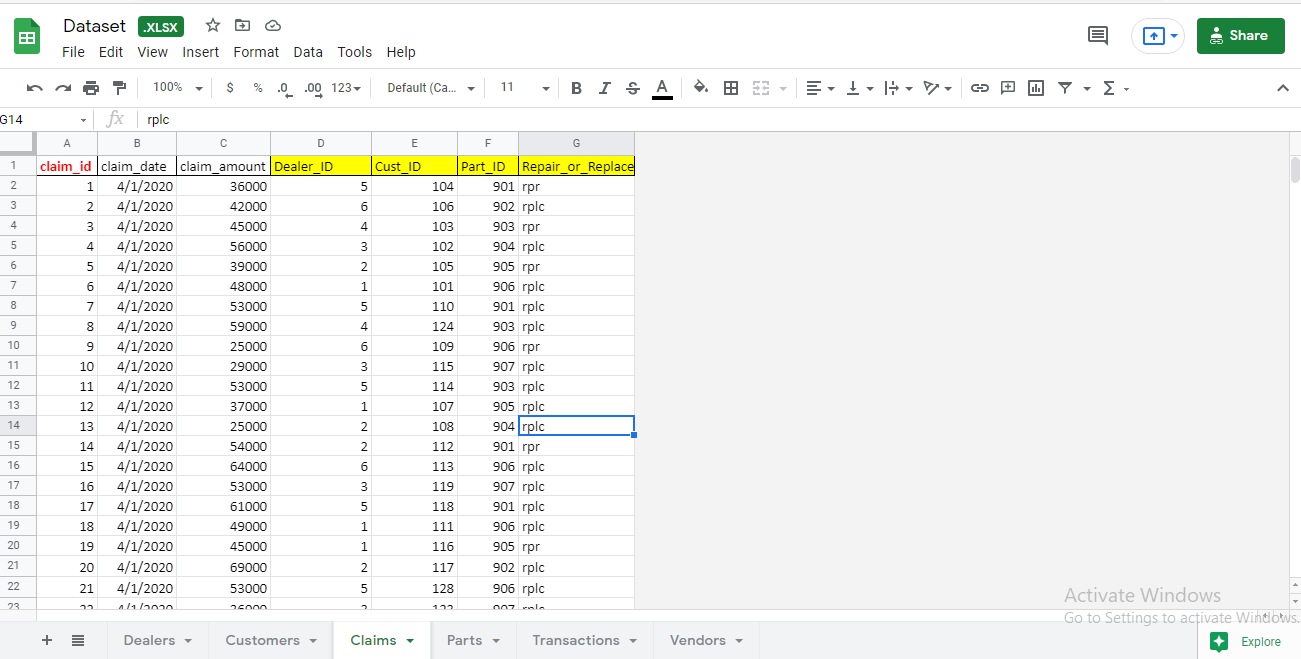
Phase 5—operationalize:   In this phase, you deliver final reports, briefings, and code and technical documents. In addition, sometimes a pilot project is also implemented in a real-time production environment. This will provide you a clear picture of the performance and other related constraints on a small scale before full deployment.

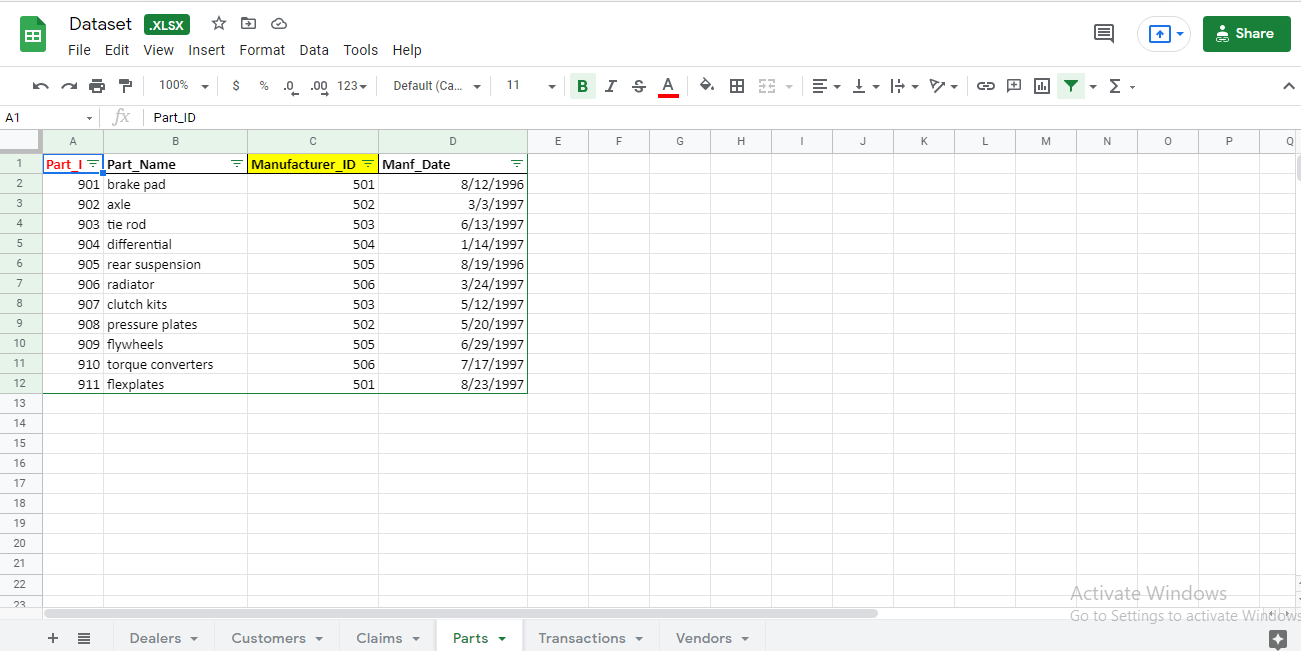
**Phase 6—Communicate results:**Now it is important to evaluate if you have been able to achieve your goal that you had planned in the first phase. So, in the last phase, you identify all the key findings, communicate to the stakeholders and determine if the results of the project are a success or a failure based on the criteria developed in Phase.

1.5.4 Dataset:

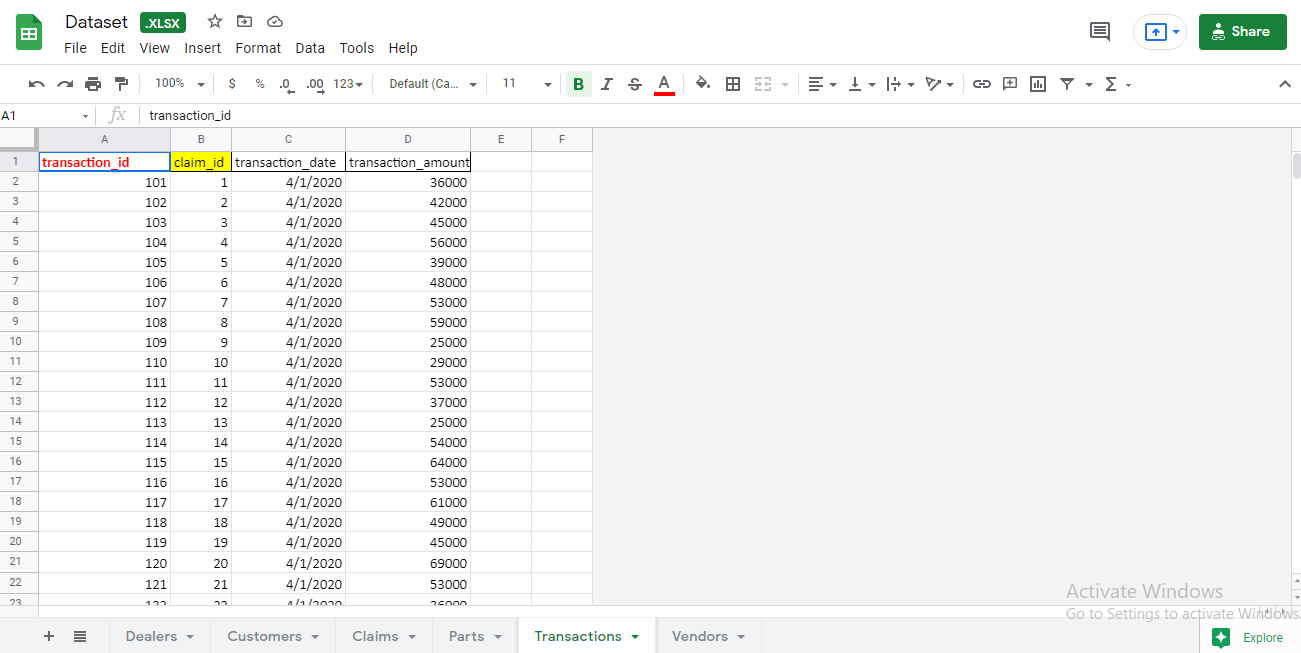
Dealers:- 

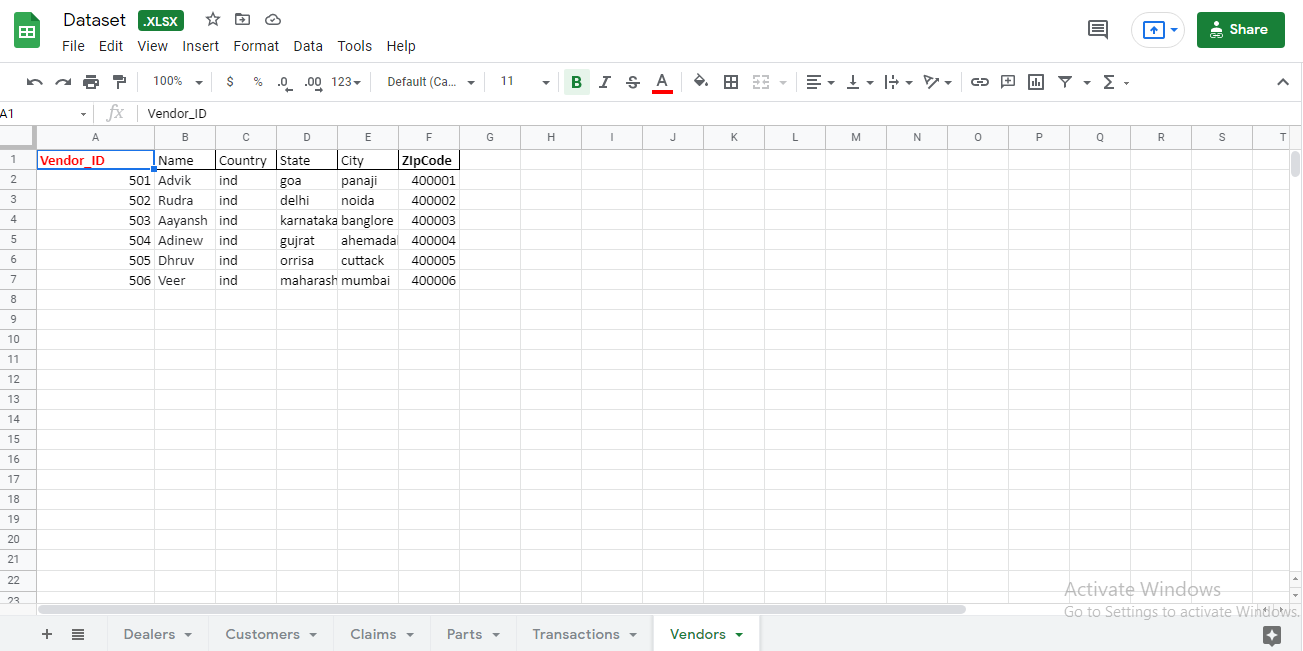
Customers:- 

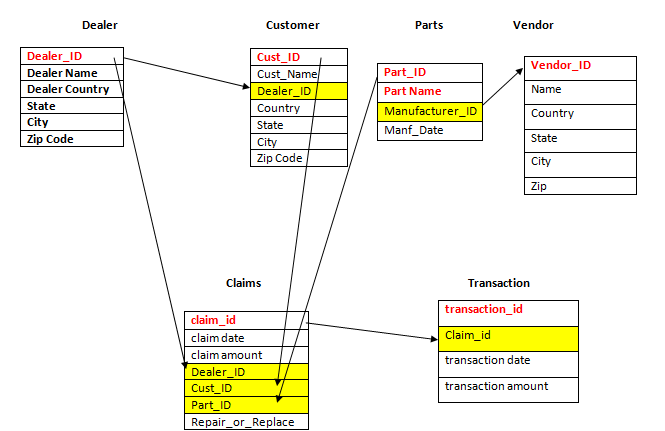
Claims:- 

Parts:- 

Transactions:-



Vendors:- 

Class-diagram:- 

**3. Source Code**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**Import seaborn as sns**

**from google.colab import files**

**Data\_files = files.upload()**

**dealer\_df = pd.read\_excel("Dataset.xlsx", sheet\_name = "Dealers")**

**Customers\_df = pd.read\_excel("Dataset.xlsx", sheet\_name = "Customers")**

**Claims\_df = pd.read\_excel("Dataset.xlsx", sheet\_name = "Claims")**

**Parts\_df = pd.read\_excel("Dataset.xlsx", sheet\_name = "Parts")**

**Transactions\_df = pd.read\_excel("Dataset.xlsx", sheet\_name = "Transactions")**

**Vendor\_df = pd.read\_excel("Dataset.xlsx", sheet\_name = "Vendors")**

**Parts\_df.head()**

**Claims\_df.head()**

**Customers\_df.head()**

**dealer\_df.head()**

**x = Claims\_df["Dealer\_ID"].value\_counts()**

**x.head(10)**

**Claims\_df["Part\_ID"].value\_counts().head(10)**

**Claims\_df.head()**

**Parts\_df.head()**

**Transactions\_df.head()**

**Vendor\_df.head()**

**mergedclaim\_transaction = pd.merge(Claims\_df,Transactions\_df, how = 'left',on ='claim\_id')**

**mergedclaim\_transaction.head()**

**sns.barplot(x="Part\_ID",y = 'transaction\_amount',hue ='Repair\_or\_Replace',data = mergedclaim\_transaction)**

**mergedclaim\_transaction["Vendor\_Location"] = mergedclaim\_transaction['Dealer\_ID'].map(dealer\_df['City'])**

**mergedclaim\_transaction.head()**

**sns.barplot(y='transaction\_amount',x = 'Vendor\_Location',hue = 'Repair\_or\_Replace',data = mergedclaim\_transaction)**

**mergedclaim\_transaction.groupby("City)['Part\_ID'].value\_counts().head()**

**mergedclaim\_transaction = pd.merge(mergedclaim\_transaction,Parts\_df,how ='left',on ='Part\_ID')**

**mergedclaim\_transaction['Age'] = mergedclaim\_transaction['claim\_date'] - mergedclaim\_transaction['Manf\_Date']**

**mergedclaim\_transaction**

**import datetime**

**mergedclaim\_transaction["Age"] = mergedclaim\_transaction["Age"].apply(lambda x: x.days)**

**sns.violinplot(y='Age',x = 'Part\_ID',hue = 'Repair\_or\_Replace',data = mergedclaim\_transaction)**

**sns.jointplot(y='Age',x = 'Part\_ID',hue = 'Repair\_or\_Replace',data = mergedclaim\_transaction)**

**mergedclaim\_transaction.columns**

**'Vendor\_Location\_Naoida','Vendor\_Location\_Panji']]**

**y=mergedclaim\_transaction['Repair\_or\_Replace\_rpr']**

**from sklearn.tree import DecisionTreeClassifier**

**classifier = DecisionTreeClassifier()**

**cross\_val\_score(classifier,x1,y1,cv=10,scoring='accuracy').mean()**

**from sklearn.ensemble import RandomForestClassifier**

**rfc = RandomForestClassifier(n\_estimators = 100)**

**cross\_val\_score(rfc,x1,y1,cv=10,scoring='accuracy').mean()**

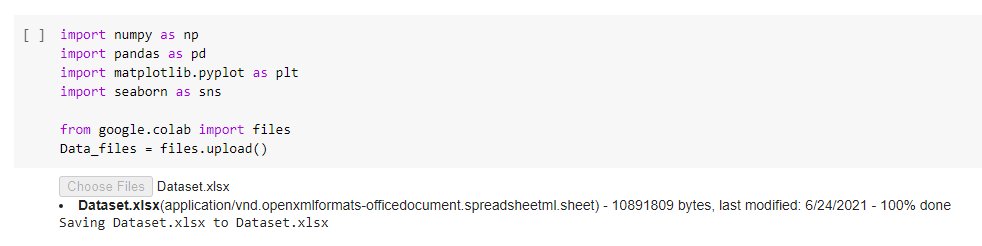
**from sklearn.tree import DecisionTreeClassifier**

**classifier = DecisionTreeClassifier()**

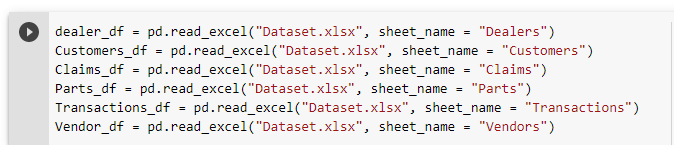
**cross\_val\_score(classifier,x2,y2,cv=10,scoring='accuracy').mean()**

**OUTPUT**

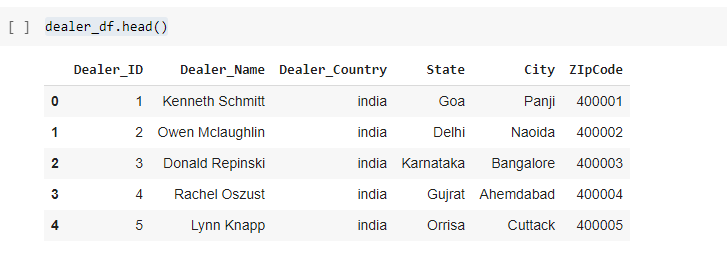
**Importing library**

****

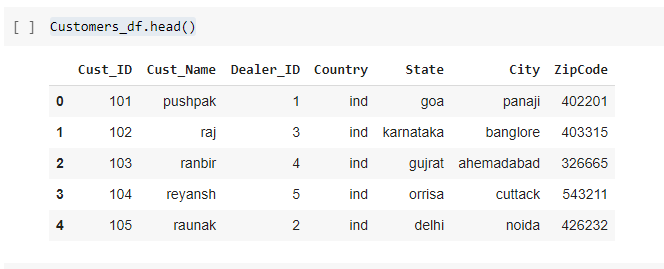
**Reading sheets**



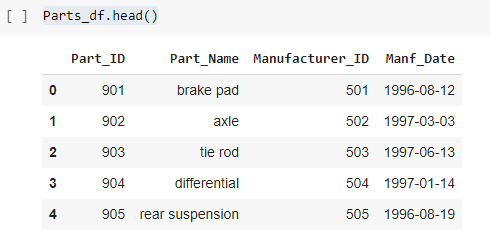
**Output of Dealer\_ID Sheet**



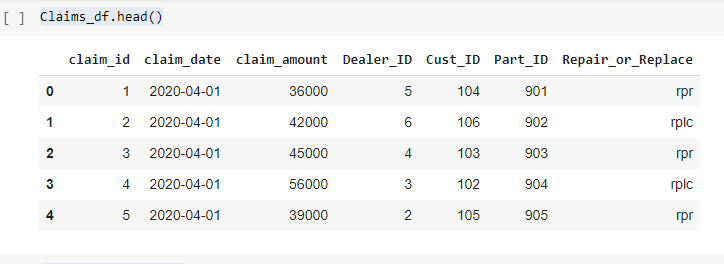
**Output of Customers Sheet**

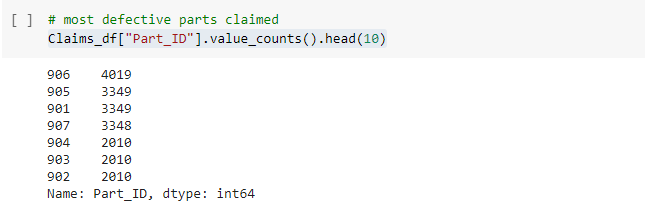


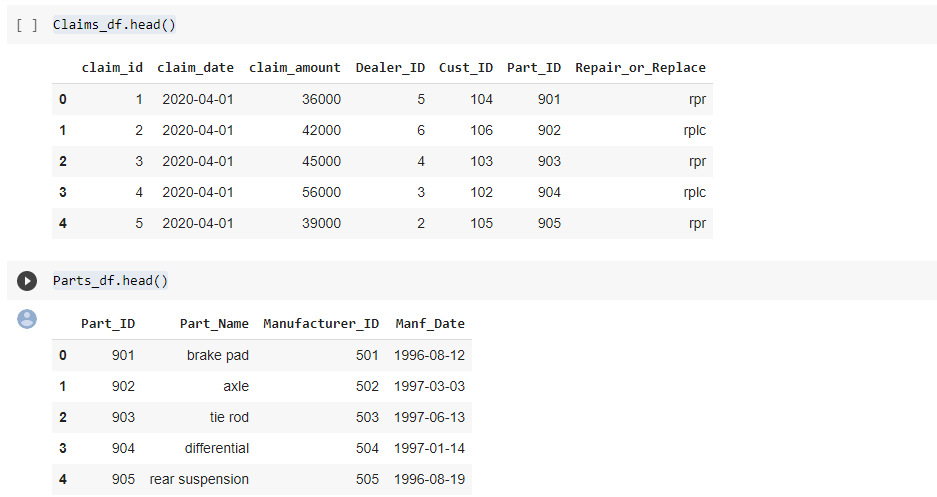
**Output Of Parts Sheet**

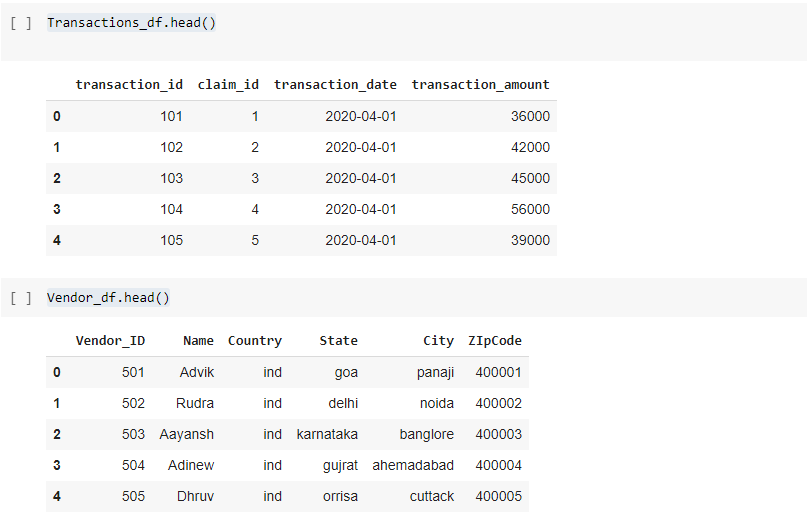


**Output of Claims Sheet**





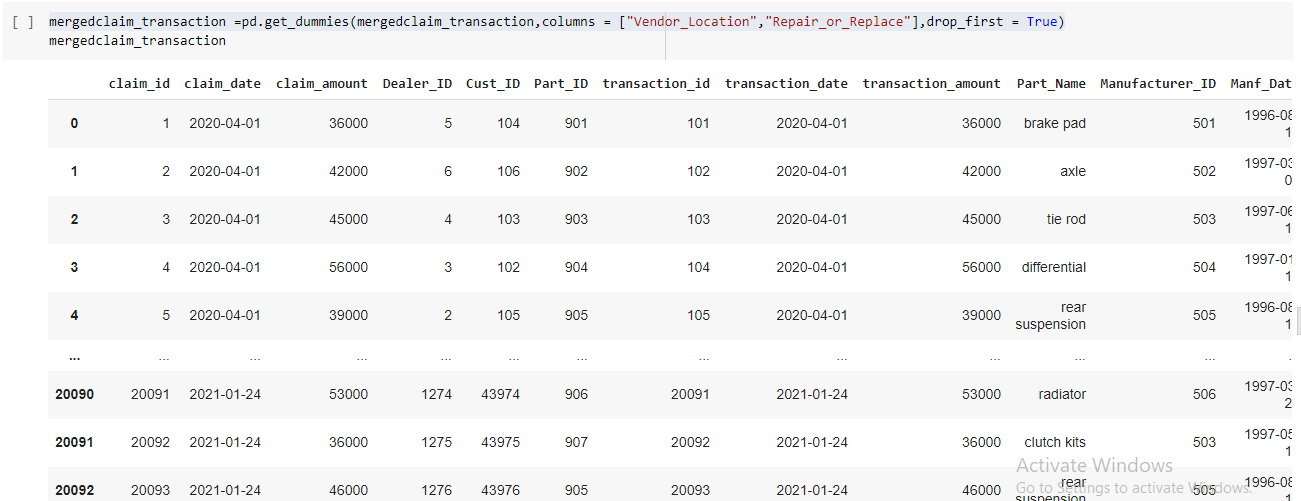
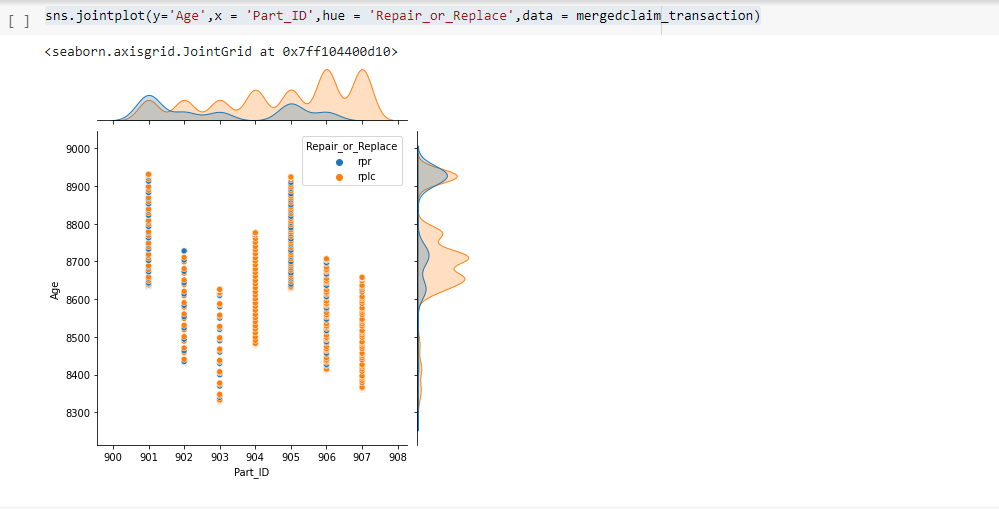
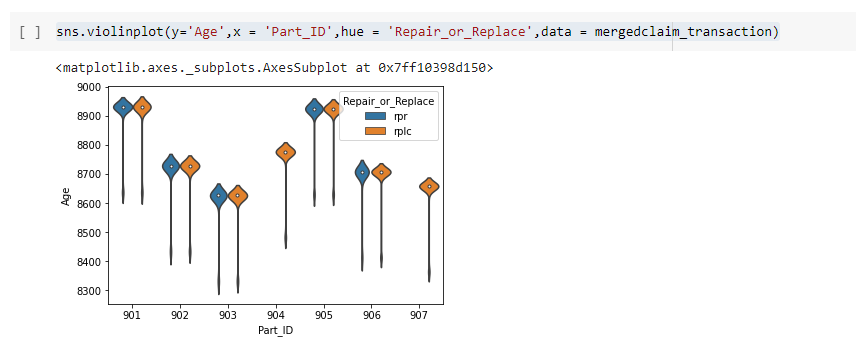
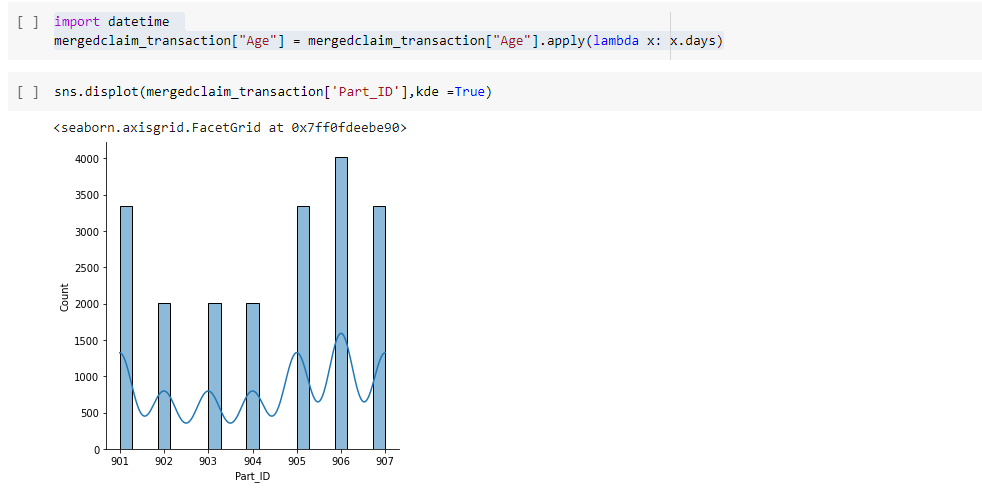
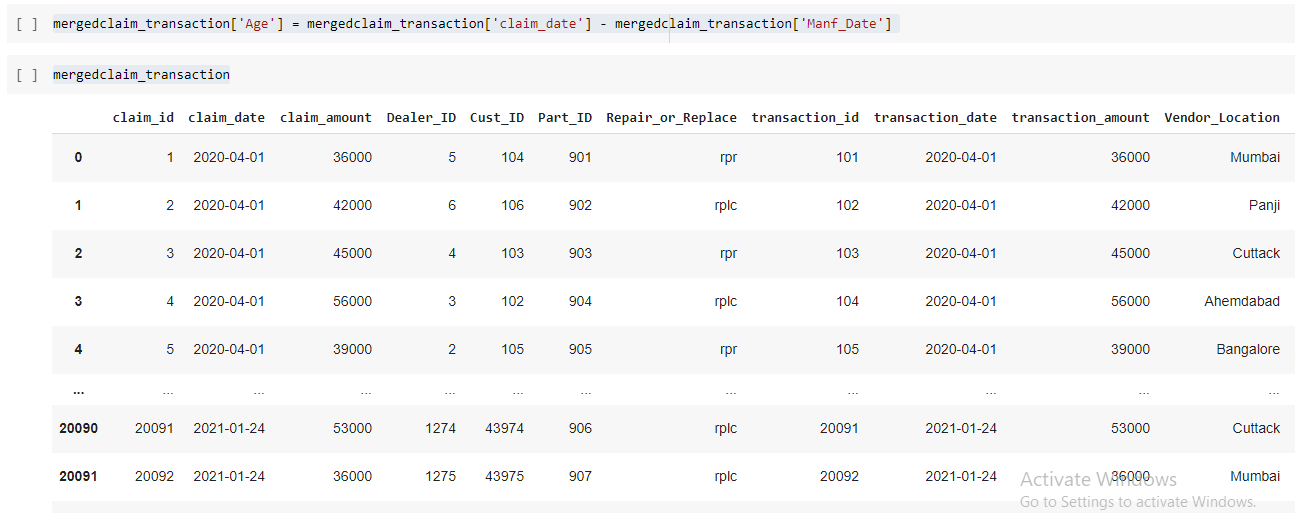
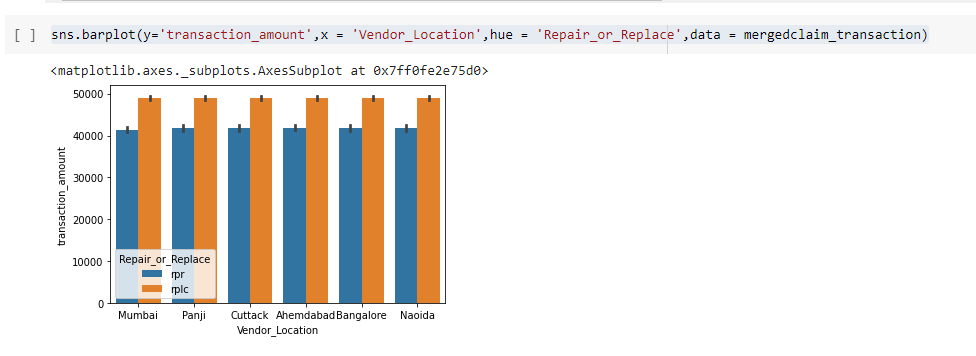
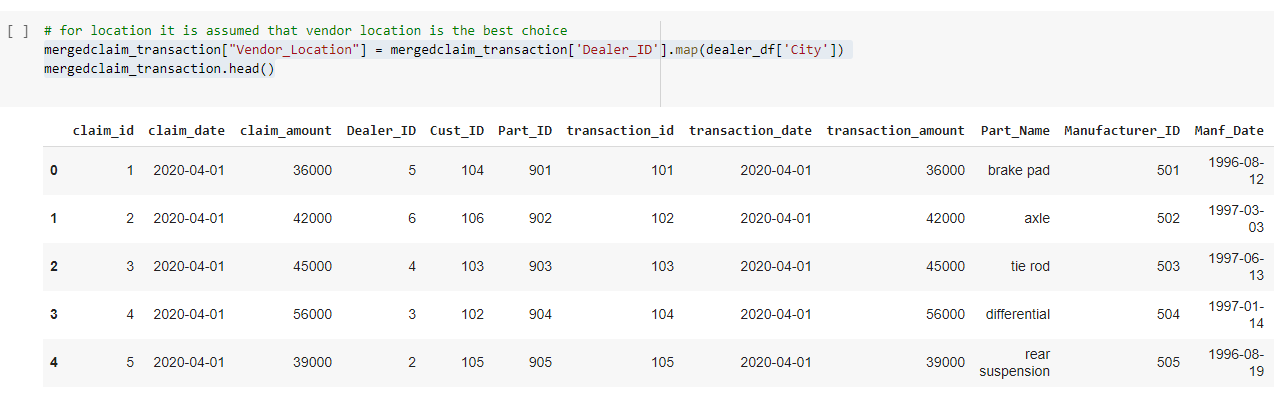
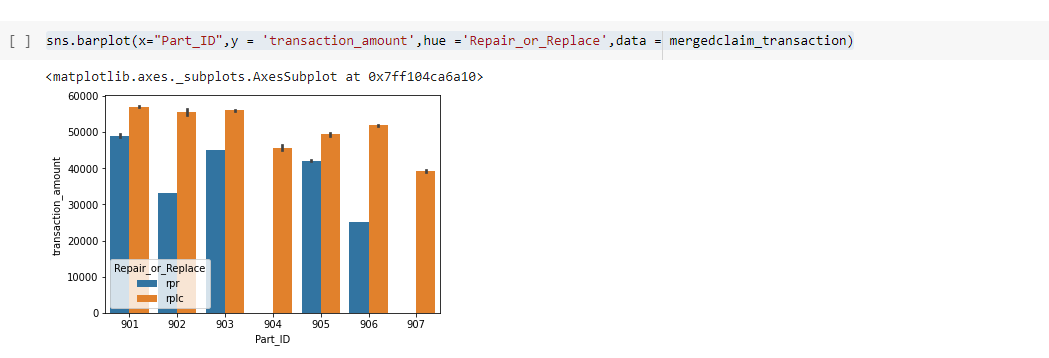
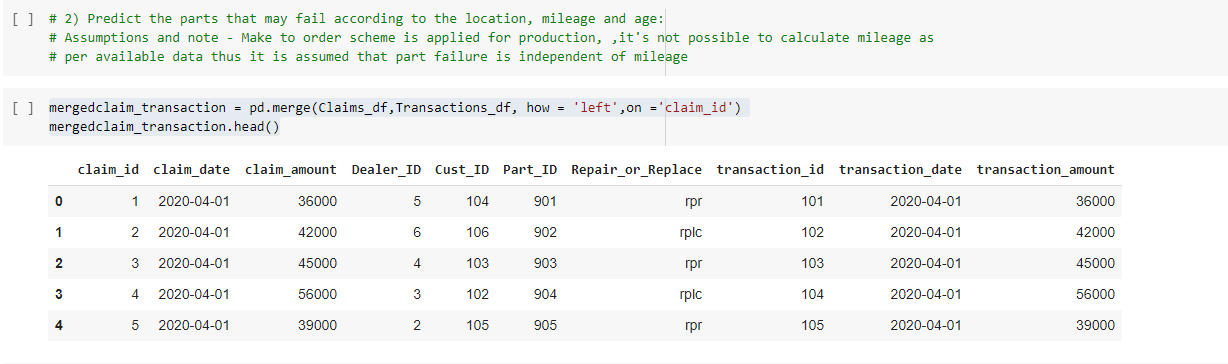




2) **Predict the parts that may fail according to the location, mileage and age**:

- In this case, by analyzing the historical data, we need to predict the parts that can be expected to fail according to the age, km driven and location of the vehicle.

- This data helps the company to understand the manufacturing requirements of the part i.e. which part is to be manufactured in what quantity so as to avoid unavailability of the parts when claim is raised for that part.

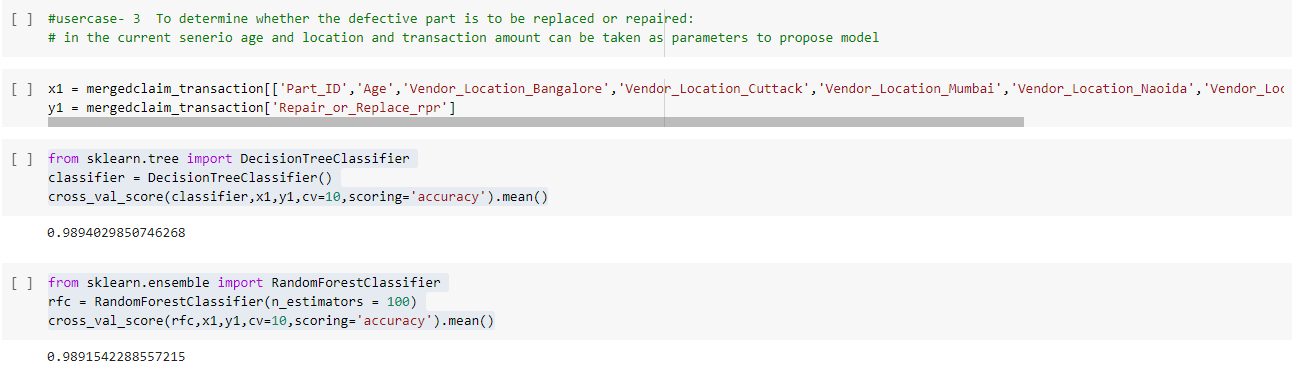


3) **To determine whether the defective part is to be replaced or repaired**:

- The model was developed to decide whether the failed part has to be repaired or replaced with the new one.

- For this, we had the last claimed data of the company.

- As this is the binary classification problem, Decision tree and Random Forest algorithms are used to build the model.



4) **Processing of variety of data: -** The data were coming from various locations like US, Canada, China, India in huge amount with varying data-types. - It was required to club the data and then process it further for the analyzing.

