SUBMITTED BY:V.JAYAPRIYADHARSHINI

NM ID:au211521106068

# STEPS FOR CLEANING DATASET USING PYTHON:

1.Handle Missing Values: Locate and deal with any missing data points. Rows with missing data can either be eliminated or filled in utilizing methods like mean, median, or interpolation, depending on the circumstances.

2. Remove Duplicates: - Search for and eliminate records that are identical, particularly if the dataset was produced from various sources. Analysis results may be skewed by duplicates.

3. Data Type Conversion:- Ascertain that the data types are suitable for analysis. For instance, dates should be formatted as dates, and categorical variables should have the proper labels. Numerical data should also be presented in its proper format.

4. Check for Outliers: - Find anomalies in the numerical data that can have a big impact on the analysis. Choose whether to eliminate outliers or to change the data to lessen their influence.

5.Normalize and Standardize Data:- If you intend to employ methods that are sensitive to the scale of variables, standardize or normalize numerical data. It is possible to apply normalization (scaling values between 0 and 1) or standardization (subtracting the mean and dividing by the standard deviation).

6. Validate Categorical Data: Verify that categorical variables only contain valid values by validating them. Correct categories that are inconsistent or misspelled.

7. Address Inconsistent Data: Keep an eye out for errors made when entering data, particularly in text areas. For instance, when referring to the same area, "NY," "New York," and "New York City" should all be used consistently.

8. Check Integrity Constraints - Verify that the connections between various columns make sense. For instance, the time of arrival should be later than the time of departure.

9. Extract pertinent data from text fields using parsing and extraction software. For instance, if a field contains both a date and an hour, separate them into different columns.

10. Validate the coordinates in the dataset if it contains geographic information to make sure they are within the expected range for the area of interest.

11. Validate data across various fields via cross-field validation. Make sure the estimated speed is within acceptable bounds, for instance, if you have a distance field and a time field.

12. note Changes: Keep a note of any modifications that were made during cleaning. This documentation is useful for reproducibility and transparency.

13. Examine the Clean Dataset:On the cleaned dataset, run preliminary analysis to make sure the data behaves as predicted. This process helps identify any problems that might have gone unnoticed during cleaning.

# OBJECTIVES OF ANALYSIS OF DATASET:

1. Recognize Trends and Patterns:

The goal is to find trends, patterns, and connections in the data.

Why: To understand the relationships between variables and how they alter as a function of time, place, or other dimensions.

1. Make Forecasts or Predictions:

The goal is to create predictive models that can predict future trends or results.

Why: To foresee events in the future, make wise decisions, and develop plans based on what is likely to happen.

1. Process improvement

The goal is to locate process bottlenecks, inefficiencies, or potential improvement areas.

Why: To boost output, cut costs, and increase overall effectiveness in a variety of industries including manufacturing, logistics, or service provision.

1. Targeting and segmentation

Segment the data into groups based on traits and choose particular groups to focus marketing or intervention efforts on.

Why? To better comprehend various customer categories, hone marketing tactics, and raise client satisfaction.

1. Detection of Anomalies:

Goal: Spot any odd trends or anomalies in the data.

Why: To catch fraud, mistakes, flaws, or other odd occurrences that may need further care or investigation.

1. Resource allocation optimization

Determine the most efficient way to distribute resources like money, labor, or time.

Why: To ensure that resources are distributed where they are most needed and to make the most of the impact of those resources.

1. Analyze any policies or interventions:

Determine the effects of particular interventions, laws, or changes in variables.

Why: To assess the efficacy of activities performed, such as policy reforms, educational initiatives, or public health interventions.

1. Analysis of consumer behavior

Understanding consumer behavior, tastes, and shopping habits is the goal.

Why: To increase product and service offerings, better the consumer experience, and customize marketing methods.

1. Churn Forecast:

Determine which clients are most likely to discontinue utilizing a service.

Why: To put retention tactics in place, lower client attrition, and keep a steady customer base.

1. Systems of recommendations:

Create algorithms to provide users with recommendations for goods, services, or content.

Why: To promote customer engagement, boost revenues, and improve user experience on channels like social media, e-commerce, and streaming services.

1. Analysis of social networks:

Analyze connections and communications within networks of people or organizations.

Why? To comprehend social structures, recognize powerful nodes, and investigate information flow throughout networks.

1. Scientific Research:

Make scientific findings by analyzing experimental or observational data.

Why: To further scientific understanding, support theories, and derive important conclusions from empirical facts.

%matplotlib inline

importnumpyasnp *#linearalgebra*

importpandasaspd *#dataprocessing,CSV fileI/O (e.g. pd.read\_csv)*

importmatplotlib.pyplotaspltimportdatetime

importos

from math import sqrtimportwarnings

*##ForMultipleOutputinsinglecell*

from IPython.core.interactiveshell import InteractiveShellInteractiveShell.ast\_node\_interactivity= "all"warnings.filterwarnings('ignore')

data=pd.read\_csv('/content/20140711.xslb.csv')data.shape

data.head(10)(82769,6)

TripIDRouteID StopID StopName

WeekBeginning\

0 23631 100 14156 181CrossRd30-06-2013

00:00

1 23631 100 14144 177CrossRd30-06-2013

00:00

2 23632 100 14132 175CrossRd30-06-2013

00:00

3 23633 100 12266Zone AArndaleInterchange 30-06-2013

00:00

4 23633 100 14147 178CrossRd30-06-2013

00:00

5 23634 100 13907 9AMarionRd30-06-2013

00:00

6 23634 100 14132 175CrossRd30-06-2013

00:00

7 23634 100 13335 9AHolbrooksRd30-06-2013

00:00

8 23634 100 13875 9MarionRd30-06-2013

00:00

9 23634 100 13045 206HolbrooksRd30-06-2013

00:00

NumberOfBoardings0 1.0

1 1.0

2 1.0

3 2.0

4 1.0

5 1.0

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 6 | 1.0 |
| 7 | 1.0 |
| 8 | 1.0 |
|  | 9 | 1.0 |

out\_geo= pd.read\_csv('/content/output\_geo.csv')out\_geo.shape

out\_geo.head()(4165,10)

accuracy formatted\_address

\

0 ROOFTOP 181 Cross Rd, Westbourne Park SA 5041, Australia1 ROOFTOP 177 Cross Rd, Westbourne Park SA 5041, Australia2 ROOFTOP 175 Cross Rd, Westbourne Park SA 5041, Australia3GEOMETRIC\_CENTERZoneA ArndaleInterchange-Southside, Kilke...

4 ROOFTOP 178 Cross Rd, Malvern SA5061, Australia

google\_place\_id input\_string latitude

\

1. ChIJKT7I9rbPsGoRVHMHkIy-Oyk 181CrossRd-34.966656
2. ChIJ-VFZ87bPsGoRyfVgC5qbPpE 177 Cross Rd -34.9666072ChIJIztlirbPsGoR38KRk76kPFI 175 Cross Rd -34.9667583ChIJn0C1hCPGsGoRIWvCdhF1RIgZone A Arndale Interchange -34.8751604ChIJycNiylvOsGoRdhfq9GKnpq0 178CrossRd-34.964960

longitudenumber\_of\_resultspostcodestatus \

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 138.592148 | 1 | 5041 | OK |  |
| 1 | 138.592301 | 1 | 5041 | OK |
| 2 | 138.592715 | 1 | 5041 | OK |
| 3 | 138.551628 | 1 | 5009 | OK |
| 4 | 138.611477 | 1 | 5061 | OK |
|  |  |  |  |  |
|  |  |  |  |  | type |
| 1. street\_address 2. street\_address 3. street\_address 4. bus\_station,establishment,point\_of\_interest,tr... 5. street\_address | | | | | |

*#DistanceFromCentre:Distancemeasurefromthecitycentre*

*#For Calculating Distance between centre with other bus stops by usingLongitude andLatitude*

*#wehaveused theHaversineformula*

from math import sin, cos, sqrt, atan2, radiansdef calc\_dist(lat1,lon1):

*##approximateradius of earthinkm*

R = 6373.0

dlon = radians(138.604801) - radians(lon1)dlat=radians(-34.921247)-radians(lat1)

a=sin(dlat/2)\*\*2+cos(radians(lat1))\*cos(radians(-34.921247))\*sin(dlon/2)\*\*2

c=2\*atan2(sqrt(a),sqrt(1- a))returnR\*c

out\_geo['dist\_from\_centre'] =out\_geo[['latitude','longitude']].apply(lambda x: calc\_dist(\*x),axis=1)

out\_geo.head()

accuracy formatted\_address

\

0 ROOFTOP 181 Cross Rd, Westbourne Park SA 5041, Australia1 ROOFTOP 177 Cross Rd, Westbourne Park SA 5041, Australia2 ROOFTOP 175 Cross Rd, Westbourne Park SA 5041, Australia3GEOMETRIC\_CENTERZoneA ArndaleInterchange-Southside, Kilke...

4 ROOFTOP 178 Cross Rd, Malvern SA5061, Australia

google\_place\_id input\_string latitude

\

1. ChIJKT7I9rbPsGoRVHMHkIy-Oyk 181CrossRd-34.966656
2. ChIJ-VFZ87bPsGoRyfVgC5qbPpE 177 Cross Rd -34.9666072ChIJIztlirbPsGoR38KRk76kPFI 175 Cross Rd -34.9667583ChIJn0C1hCPGsGoRIWvCdhF1RIgZone A Arndale Interchange -34.8751604ChIJycNiylvOsGoRdhfq9GKnpq0 178CrossRd-34.964960

longitudenumber\_of\_resultspostcodestatus \

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 138.592148 | 1 | 5041 | OK |
| 1 | 138.592301 | 1 | 5041 | OK |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | |
| 2 | 138.592715 | 1 | 5041 | OK |
| 3 | 138.551628 | 1 | 5009 | OK |  |  |
| 4 | 138.611477 | 1 | 5061 | OK |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | type | dist\_from\_centre |

|  |  |
| --- | --- |
| 0 street\_address | 5.180961 |
|  |  |
| 1 street\_address | 5.172525 |
|  |  |
| 2 street\_address | 5.180709 |
|  |  |
| 3bus\_station,establishment,point\_of\_interest,tr... | 7.057549 |
|  |  |
| 4 street\_address | 4.900099 |

*#exp\_data=out\_geo.head(10)*

*##Fill the missing valueswithmode*

out\_geo['type'].fillna('street\_address',inplace=True)

out\_geo['type']=out\_geo['type'].apply(lambdax:str(x).split(',')[-1])

out\_geo['type'].unique()

array(['street\_address','transit\_station','premise','political',

'school', 'route','intersection','point\_of\_interest',

'subpremise', 'real\_estate\_agency', 'university','travel\_agency',

'restaurant', 'supermarket', 'store', 'post\_office'],dtype=object)

data.info()

<class 'pandas.core.frame.DataFrame'>RangeIndex: 82769 entries, 0 to 82768Data columns(total6columns):

#Column Non-NullCount Dtype

* 1. TripID 82769non-nullint64
  2. RouteID 82769non-nullint64
  3. StopID 82769non-nullint64
  4. StopName 82769non-null object
  5. WeekBeginning 82768non-null object

5 NumberOfBoardings82768 non-nullfloat64dtypes:float64(1),int64(3), object(2)

memory usage:3.8+MBdata.head(3)

TripIDRouteIDStopID StopName WeekBeginningNumberOfBoardings

0 23631 100 14156181 Cross Rd30-06-2013 00:00

1.0

1 23631 100 14144177 Cross Rd30-06-2013 00:00

1.0

2 23632 100 14132175 Cross Rd30-06-2013 00:00

1.0

data.tail(3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TripID | RouteID | StopID | StopName | WeekBeginning | \ |
| 82766 13354 | 100 | 14152 | 179Cross Rd | 01-06-201400:00 |  |
| 82767 13354 | 100 | 12352 | Woodville | 01-06-201400:00 |  |
| 82768 13354 | 100 | 13767 | 8H | NaN |  |

NumberOfBoardings

82766 1.0

82767 1.0

82768 NaN

data.describe()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | TripID | RouteID | StopID | NumberOfBoardings |
| count | 82769.000000 | 82769.0 | 82769.000000 | 82768.000000 |
| mean | 28094.843166 | 100.0 | 13475.915923 | 3.604545 |
| std | 18674.361134 | 0.0 | 745.937586 | 7.038205 |
| min | 5605.000000 | 100.0 | 12213.000000 | 1.000000 |
| 25% | 5651.000000 | 100.0 | 12839.000000 | 1.000000 |
| 50% | 44679.000000 | 100.0 | 13627.000000 | 2.000000 |
| 75% | 44704.000000 | 100.0 | 14099.000000 | 4.000000 |
| max | 44729.000000 | 100.0 | 17881.000000 | 181.000000 |

data.isna().sum()TripID 0

RouteID 0

StopID 0

StopName 0

WeekBeginning 1

NumberOfBoardings 1

dtype: int64

data.dropna(inplace=True)data.isna().sum()

TripID 0

RouteID 0

StopID 0

StopName 0

WeekBeginning 0

NumberOfBoardings 0

dtype: int64

data[data.duplicated()]

EmptyDataFrame

Columns: [TripID, RouteID, StopID, StopName, WeekBeginning,NumberOfBoardings]

Index:[]

data['TripID']=data['TripID'].astype(float)data.head(3)

TripIDRouteIDStopID StopName WeekBeginningNumberOfBoardings

023631.0 100 14156181 CrossRd30-06-201300:00

1.0

123631.0 100 14144177 CrossRd30-06-201300:00

1.0

223632.0 100 14132175 CrossRd30-06-201300:00

1.0

data['StopName'] = data['StopName'].str.strip()data['WeekBeginning'] =pd.to\_datetime(data['WeekBeginning'])from sklearn.preprocessingimportStandardScaler

scaler=StandardScaler()

data[['StopID']] = scaler.fit\_transform(data[['StopID']])data.to\_csv('cleaned\_data.csv',index=False)

from google.colab import drivedrive.mount('/content/drive')

Mounted at/content/drive



















