310256: Data Science and Big Data Analytics Laboratory

Data Wrangling, I

Perform the following operations using Python on any open source dataset (e.g., data.csv)

1. Import all the required Python Libraries.
2. Locate an open source data from the web (e.g., https://www.kaggle.com). Provide a clear description of the data and its source (i.e., URL of the web site).
3. Load the Dataset into pandas dataframe.
4. Data Preprocessing: check for missing values in the data using pandas isnull(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.
5. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.
6. Turn categorical variables into quantitative variables in Python.

In addition to the codes and outputs, explain every operation that you do in the above steps and explain everything that you do to import/read/scrape the data set.

1. Import all the required Python Libraries.

```
import pandas as pd
```

2. Locate an open source data from the web (e.g., https://www.kaggle.com). Provide a clear description of the data and its source (i.e., URL of the web site).

✓ 0s completed at 3:14 PM

first go to our account page on Kaggle to generate an API token. On the account page, we scroll down to API section.

open a new notebook in colab and run the following command.

```
!pip install -q kaggle
```

```
import os
os.environ['KAGGLE_CONFIG_DIR'] = "/content/gdrive/MyDrive/DataSets/Kaggle/"
```

upload kaggle.json file.

```
from google.colab import files
files.upload()
```

Choose “kaggle.json” downloaded from Kaggle. We need save this file into a directory named kaggle. Run the following commands to accomplish this task:

```
!mkdir ~/.kaggle
!cp kaggle.json ~/.kaggle/
```

```
mkdir: cannot create directory ‘/root/.kaggle’: File exists
```

change the permission of the file using the following command:

```
!chmod 600 ~/.kaggle/kaggle.json
```

see the datasets available on kaggle:

```
!kaggle datasets list
```

```
ref                                                            title                                                 
-------------------------------------------------------------  ----------------------------------------------------  
ahsan81/hotel-reservations-classification-dataset              Hotel Reservations Dataset                            
googleai/musiccaps                                             MusicCaps                                             
themrityunjaypathak/most-subscribed-1000-youtube-channels      Most Subscribed 1000 Youtube Channels                 
nitishsharma01/olympics-124-years-datasettill-2020             Olympics 124 years Dataset(till 2020)                 
thedevastator/medical-student-mental-health                    Medical Student Mental Health                         
senapatirajesh/netflix-tv-shows-and-movies                     Latest Netflix TV shows and movies                    
thedevastator/predicting-credit-card-customer-attrition-with-m  Predicting Credit Card Customer Segmentation        
kane6543/most-watched-stocks-of-past-decade20132023            Most Watched Stocks of Past Decade(2013-2023)         
karkavelrajaj/amazon-sales-dataset                             Amazon Sales Dataset                                  
thedevastator/us-film-industry-top-movies-directors            US Film Industry Top Movies & Directors               
salimwid/latest-top-3000-companies-ceo-salary-202223           CEO vs Worker Pay in Top 3000 US Companies [2023]     
tymekurban/new-cars-usa-202223-dataset                         New Cars USA 2022/23 dataset                          
abhishek14398/salary-dataset-simple-linear-regression          Salary Dataset - Simple linear regression             
salimwid/technology-company-layoffs-20222023-data              Technology Company Layoffs (2022-2023)                
thedevastator/higher-education-predictors-of-student-retention  Predict students' dropout and academic success       
rhugvedbhojane/fifa-world-cup-2022-players-statistics          FIFA World Cup 2022 Players Statistics                
karimabdulnabi/car-price                                       Car Price                                             
rishikeshkonapure/home-loan-approval                           Home Loan Approval                                    
rakkesharv/spotify-top-10000-streamed-songs                    Spotify Top 10000 Streamed Songs                      
thedevastator/canine-intelligence-and-size                     Dogs Intelligence and Size                            
```

To download a particular one:

!kaggle datasets download -d "name_of_the_dataset"

For instance:

```
!kaggle datasets download -d ahsan81/hotel-reservations-classification-dataset --force
```

```
Downloading hotel-reservations-classification-dataset.zip to /content/gdrive/MyDrive/DataSets/Kaggle
  0% 0.00/480k [00:00<?, ?B/s]
100% 480k/480k [00:00<00:00, 46.3MB/s]
```

3.Load the Dataset into pandas dataframe.

```
df=pd.read_csv("/content/gdrive/MyDrive/DataSets/Kaggle/hotel-reservations-classification-dataset.zip")
```

```
df.head()
```

We don’t have to go through all of these steps everytime. Once these steps are completed, next time we just upload kaggle.json to colab and then download the dataset.

4. Data Preprocessing:

check for missing values in the data using pandas isnull(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.

```
df.head(5)
```

```
df.tail(5)
```

```
df.describe()
```

```
df.describe(include="all")
```

```
df.size
```

```
689225
```

```
df.shape
```

```
(36275, 19)
```

```
df.columns
```

```
Index(['Booking_ID', 'no_of_adults', 'no_of_children', 'no_of_weekend_nights',
       'no_of_week_nights', 'type_of_meal_plan', 'required_car_parking_space',
       'room_type_reserved', 'lead_time', 'arrival_year', 'arrival_month',
       'arrival_date', 'market_segment_type', 'repeated_guest',
       'no_of_previous_cancellations', 'no_of_previous_bookings_not_canceled',
       'avg_price_per_room', 'no_of_special_requests', 'booking_status'],
      dtype='object')
```

```
df['Booking_ID']
```

```
0        INN00001
1        INN00002
2        INN00003
3        INN00004
4        INN00005
```

```
              ...
36270    INN36271
36271    INN36272
36272    INN36273
36273    INN36274
36274    INN36275
Name: Booking_ID, Length: 36275, dtype: object
```

```
df[0:2]
```

```
df.loc[0:2]
```

```
df.iloc[0:2]
```

```
df.loc[0:2,"Booking_ID":"type_of_meal_plan"]
```

```
df.iloc[0:2,1:5]
```

check for missing values in the data using pandas isnull()

```
df.isnull()
```

```
df.isna()
```

```
df.isnull().any()
```

```
Booking_ID                            False
no_of_adults                          False
no_of_children                        False
no_of_weekend_nights                  False
no_of_week_nights                     False
type_of_meal_plan                     False
required_car_parking_space            False
room_type_reserved                    False
lead_time                             False
arrival_year                          False
arrival_month                         False
arrival_date                          False
market_segment_type                   False
repeated_guest                        False
no_of_previous_cancellations          False
no_of_previous_bookings_not_canceled  False
avg_price_per_room                    False
no_of_special_requests                False
booking_status                        False
dtype: bool
```

```
df.isnull().sum()
```

```
Booking_ID                            0
no_of_adults                          0
no_of_children                        0
no_of_weekend_nights                  0
```

```
no_of_week_nights                     0
type_of_meal_plan                     0
required_car_parking_space            0
room_type_reserved                    0
lead_time                             0
arrival_year                          0
arrival_month                         0
arrival_date                          0
market_segment_type                   0
repeated_guest                        0
no_of_previous_cancellations          0
no_of_previous_bookings_not_canceled  0
avg_price_per_room                    0
no_of_special_requests                0
booking_status                        0
dtype: int64
```

count of missing values of a specific column.

```
df.Booking_ID.isnull().sum()
```

```
0
```

5. Data Formatting and Data Normalization:

Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.

Data Formatting:

```
df.dtypes
```

```
Booking_ID                            object
no_of_adults                           int64
```

```
no_of_children                          int64
no_of_weekend_nights                    int64
no_of_week_nights                       int64
type_of_meal_plan                      object
required_car_parking_space              int64
room_type_reserved                     object
lead_time                               int64
arrival_year                            int64
arrival_month                           int64
arrival_date                            int64
market_segment_type                    object
repeated_guest                          int64
no_of_previous_cancellations            int64
no_of_previous_bookings_not_canceled    int64
avg_price_per_room                    float64
no_of_special_requests                  int64
booking_status                         object
dtype: object
```

```
df.arrival_date=df.arrival_date.astype("float")
```

```
df.dtypes
```

```
Booking_ID                             object
no_of_adults                            int64
no_of_children                          int64
no_of_weekend_nights                    int64
no_of_week_nights                       int64
type_of_meal_plan                      object
required_car_parking_space              int64
room_type_reserved                     object
lead_time                               int64
arrival_year                            int64
arrival_month                           int64
arrival_date                          float64
market_segment_type                    object
repeated_guest                          int64
no_of_previous_cancellations            int64
no_of_previous_bookings_not_canceled    int64
avg_price_per_room                    float64
```

```
no_of_special_requests                int64
booking_status                       object
dtype: object
```

Data Normalization

```
from sklearn import preprocessing
```

```
df.arrival_date
```

```
0         2.0
1         6.0
2        28.0
3        20.0
4        11.0
         ...
36270     3.0
36271    17.0
36272     1.0
36273    21.0
36274    30.0
Name: arrival_date, Length: 36275, dtype: float64
```

```
min_max_scaler = preprocessing.MinMaxScaler()
```

```
x=df.arrival_date
```

```
x_scaled = min_max_scaler.fit_transform(x)
```

```
df_normalized = pd.DataFrame(min_max_scaler.fit_transform(x))
```

```
df_normalized
```

6. Turn categorical variables into quantitative variables in Python.

There are many ways to convert categorical data into numerical data. Here the three most used methods are discussed.

i. Label Encoding:

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. It is an important preprocessing step for the structured dataset in supervised learning.

```
from sklearn import preprocessing
```

```
df['type_of_meal_plan'].unique()
```

```
array(['Meal Plan 1', 'Not Selected', 'Meal Plan 2', 'Meal Plan 3'],
      dtype=object)
```

```
label_encoder = preprocessing.LabelEncoder()
```

```
df['type_of_meal_plan']= label_encoder.fit_transform(df['type_of_meal_plan'])
```

```
df['type_of_meal_plan'].unique()
```

```
array([0, 3, 1, 2])
```

ii. One-Hot Encoding:

```
from sklearn import preprocessing
```

```
df['type_of_meal_plan'].unique()
```

```
array([0, 3, 1, 2])
```

```
one_hot_df = pd.get_dummies(df, prefix="type_of_meal_plan",columns=['type_of_meal_plan'], drop_first=True)
```

```
one_hot_df
```

iii. Replace()

```
df.head(5)
```

```
df.market_segment_type.unique()
```

```
array(['Offline', 'Online', 'Corporate', 'Aviation', 'Complementary'],
      dtype=object)
```

**Conclusion- **

In this way we have explored the functions of the python library for Data Preprocessing, Data Wrangling Techniques and How to Handle missing values on Iris Dataset.

In addition to the codes and outputs, explain every operation that you do in the above steps and explain everything that you do to import/read/scrape the data set.

Colab paid products - Cancel contracts here