Project 7 Solutions

Kevin Choe

Collaborators: (Collaborators listed here. Include names, which part of the project you gave or sought help with, and how you helped or were helped.)

TA help: Summeth Guda

Online resources used: (List of links/resources (if any) here. Include web addresses, which part of the project the resource helped with, and how you were helped.)

Question 1

```
import pandas as pd
```

```
businesses = pd.read_parquet("/class/datamine/data/yelp/data/parquet/businesses.parquet")
businesses[0:5]

#The names of the datasets are businesses, checkins, reviews, users, businesses_sample, photos, and tip

#The businesses includes the busseiness id and hours.
#The checkins includes the business id and the date.
#The review includes the review id and the date.
#The users includes the user id, name, compliment writer, and compliment photos.
#The businesses sample includesthe business id and hours.
#The photos include the photoid and label.
#The tips includes the userid and compliment count.
```

```
business_id ...  hours
0 f9NumwFMBDn751xgFiRbNA ...  {'Friday': '11:0-20:0', 'Monday': '10:0-18:0',...
1 YzvjgOSayhoZgCljUJRF9Q ...  None
2 XNoUzKckATkOD1hP6vghZg ...  None
3 6OAZjbxqM5ol29BuHsil3w ...  {'Friday': '7:0-16:0', 'Monday': '7:0-16:0', '...
4 51M2Kk903DFYI6gnB5I6SQ ...  {'Friday': '9:0-16:0', 'Monday': '0:0-0:0', 'S...

[5 rows x 14 columns]
```

Question 2

```
business = pd.read_parquet("/class/datamine/data/yelp/data/parquet/businesses.parquet")
len(business.loc[:, "attributes"].iloc[0].keys()) # 39
```

```
39
```

```
len(business.loc[:, "hours"].iloc[0].keys())
```

```
7
```

```
def has_attributes(business_id_number):
  returnval = False

  for i in range (0, business.shape[0]):
    if (business["business_id"][i] == business_id_number):

      if(business["attributes"][i] != None):
        returnval = True
  return returnval

print(has_attributes('f9NumwFMBDn751xgFiRbNA')) # True
```

```
True
```

```
print(has_attributes('XNoUzKckATkOD1hP6vghZg')) # False
```

```
False
```

```
print(has_attributes('YzvjgOSayhoZgCljUJRF9Q')) # True
```

```
True
```

```
print(has_attributes('7uYJJpwORUbCirC1mz8n9Q')) # False
```

```
False
```

Question 3

```
businesses.loc[0:5, "hours"].apply(pd.Series)
```

	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
0	11:0-20:0	10:0-18:0	11:0-20:0	13:0-18:0	11:0-20:0	11:0-20:0	10:0-18:0
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	7:0-16:0	7:0-16:0	None	None	7:0-16:0	7:0-16:0	7:0-16:0
4	9:0-16:0	0:0-0:0	None	None	9:0-16:0	9:0-16:0	9:0-16:0
5	7:0-18:0	7:0-18:0	7:0-15:0	None	7:0-18:0	7:0-18:0	7:0-18:0

```
businesses.loc[0:5, "attributes"].apply(pd.Series)
```

```
AcceptsInsurance AgesAllowed Alcohol ...  Smoking WheelchairAccessible WiFi
0 NaN NaN NaN ...  NaN NaN NaN
1 NaN NaN NaN ...  NaN NaN NaN
2 NaN NaN NaN ...  NaN NaN NaN
3 NaN NaN NaN ...  NaN NaN NaN
4 NaN NaN NaN ...  NaN NaN NaN
5 NaN NaN NaN ...  NaN NaN NaN

[6 rows x 39 columns]
```

```
from pathlib import Path

def fix_businesses_data(data_path: str, output_dir: str) -> None:
    """
    fix_data accepts a parquet file that contains data in a specific format.
    fix_data "explodes" the attributes and hours columns into 39+7=46 new
    columns.
```

```
    Args:
        data_path (str): Full path to a file in the same format as businesses.parquet.
        output_dir (str): Path to a directory where new_businesses.parquet should be output.
    """
    # read in original parquet file
    businesses = pd.read_parquet(data_path)

    # unnest the attributes column
    businesses = pd.concat([business.drop(columns=['attributes']), businesses.loc[:, 'attributes'].apply

    # unnest the hours column
    businesses = pd.concat([business.drop(columns=['hours']), businesses.loc[:, 'hours'].apply(pd.Series

    # output new file
    businesses.to_parquet(str(Path(f"{output_dir}").joinpath("new_businesses.parquet")))

    return None

attributesDF = businesses.loc[ : , "attributes"].apply(pd.Series)
hoursDF = businesses.loc[ : , "hours"].apply(pd.Series)
attributesDF.shape
```

```
(209393, 39)
```

```
hoursDF.shape
```

```
(209393, 7)
```

```
myDF = pd.concat([attributesDF,hoursDF], axis=1)
myDF.shape
```

```
(209393, 46)
```

```
p = Path(f"/scratch/scholar/choe29").glob('**/*')
files = [x for x in p if x.is_file()]
print(files)
```

```
[PosixPath('/scratch/scholar/choe29/2018.csv'),
PosixPath('/scratch/scholar/choe29/2018.feather'),
PosixPath('/scratch/scholar/choe29/2018.parquet')]
```

Question 4

```
def unnest(inputDF: pd.DataFrame, columns: list) -> pd.DataFrame:
  #inputDF = pd.DataFrame()
  for mycolumn in columns:
    tempDF = inputDF.loc[ : , mycolumn].apply(pd.Series)
    inputDF = pd.concat([inputDF,tempDF], axis=1)
  return inputDF

businesses = pd.read_parquet("/class/datamine/data/yelp/data/parquet/businesses.parquet")

new_businesses_df = unnest(businesses, ["attributes", ])
new_businesses_df.shape # (209393, 39)
```

```
(209393, 53)
```

```
new_businesses_df.head()
```

```
business_id name ... WheelchairAccessible WiFi
0 f9NumwFMBDn751xgFiRbNA The Range At Lake Norman ...  None None
1 YzvjgOSayhoZgCljUJRF9Q Carlos Santo, NMD ...  None None
2 XNoUzKckATkOD1hP6vghZg Felinus ...  NaN NaN
3 6OAZjbxqM5ol29BuHsil3w Nevada House of Hose ...  None None
4 51M2Kk903DFYI6gnB5I6SQ USE MY GUY SERVICES LLC ...  None None

[5 rows x 53 columns]
```

```
new_businesses_df = unnest(businesses, ["attributes", "hours"])
new_businesses_df.shape # (209393, 46)
```

```
(209393, 60)
```

```
new_businesses_df.head()
```

```
              business_id                        name  ...    Tuesday  Wednesday
0  f9NumwFMBDn751xgFiRbNA  The Range At Lake Norman  ...  11:0-20:0  10:0-18:0
1  YzvjgOSayhoZgCljUJRF9Q         Carlos Santo, NMD  ...        NaN        NaN
2  XNoUzKckATkOD1hP6vghZg                   Felinus  ...        NaN        NaN
3  6OAZjbxqM5ol29BuHsil3w      Nevada House of Hose  ...   7:0-16:0   7:0-16:0
4  51M2Kk903DFYI6gnB5I6SQ   USE MY GUY SERVICES LLC  ...   9:0-16:0   9:0-16:0

[5 rows x 60 columns]
```

Question 5

```
def unnest(inputDF: pd.DataFrame, columns: list) -> pd.DataFrame:
  myDF = pd.DataFrame()
  for mycolumn in columns:
    if mycolumn in inputDF.columns:
      mysum = 0
      for i in range(0,inputDF.shape[0]):
        if isinstance(inputDF[mycolumn][i],dict):
          mysum += 1
      if mysum > 0:

        tempDF = inputDF.loc[ : , mycolumn].apply(pd.Series)
        myDF = pd.concat([myDF,tempDF], axis=1)
  return myDF

businesses = pd.read_parquet("/class/datamine/data/yelp/data/parquet/businesses.parquet")


businesses['attributes'][2]
isinstance(business['attributes'][2],dict)
```

```
False
```

```
businesses = pd.read_parquet("/class/datamine/data/yelp/data/parquet/businesses.parquet")
results = unnest(businesses, ["doesntexist", "postal_code", "attributes"])
results.shape # (209393, 39)
```

```
(209393, 39)
```

```
results.head()
```

```
  AcceptsInsurance AgesAllowed Alcohol  ... Smoking WheelchairAccessible  WiFi
0             None        None    None  ...    None                 None  None
1             None        None    None  ...    None                 None  None
2              NaN         NaN     NaN  ...     NaN                  NaN   NaN
3             None        None    None  ...    None                 None  None
4             None        None    None  ...    None                 None  None

[5 rows x 39 columns]
```

Pledge

By submitting this work I hereby pledge that this is my own, personal work. I've acknowledged in the designated place at the top of this file all sources that I used to complete said work, including but not limited to: online resources, books, and electronic communications. I've noted all collaboration with fellow students and/or TA's. I did not copy or plagiarize another's work.

> As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together – We are Purdue.