DATA WRANGLING

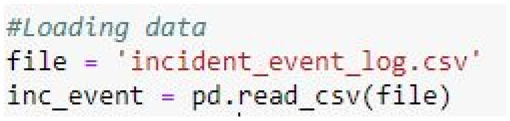
Dataset location : incident_managment_process_enriched_event_log
Jupyter Notebook : github_link

Above are all the relevant files used in this project. Incident management process enriched event log data set is raw information in csv format that has been taken from UCI repository as mentioned above.

This dataset as mentioned in UCI website has missing values and it needs to be corrected before performing any other operations. Below are the finds from initial findings from the raw dataset.

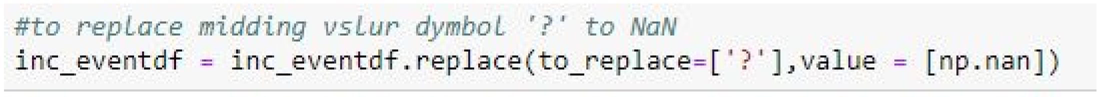
- Data set consists of 141712 rows and 36 columns.
- There are multiple columns with missing values marked in the dataset by ‘?’.
- Also found 5 different columns with inaccurate data types.
- As a subject matter column “incident_managment” one extra categorical value that was out of place, namely ‘-100’.

First course of action was to read the data.



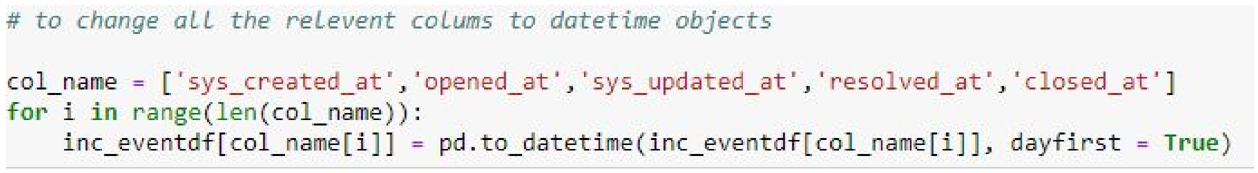
```
#Loading data
file = 'incident_event_log.csv'
inc_event = pd.read_csv(file)
```

Next step is to replace ‘?’ which symbolizes the missing values throughout the dataset with NaN for better maneuverability of data further on .



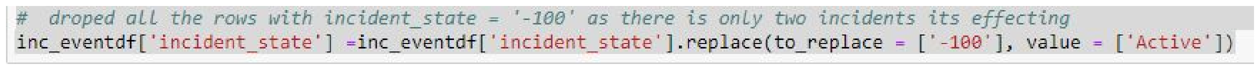
```
#to replace midding vslur dymbol '?' to NaN
inc_eventdf = inc_eventdf.replace(to_replace=['?'],value = [np.nan])
```

Moving forward to column data types we found earlier that there are 5 different columns was wrong data types. We solve the issue as shown below.



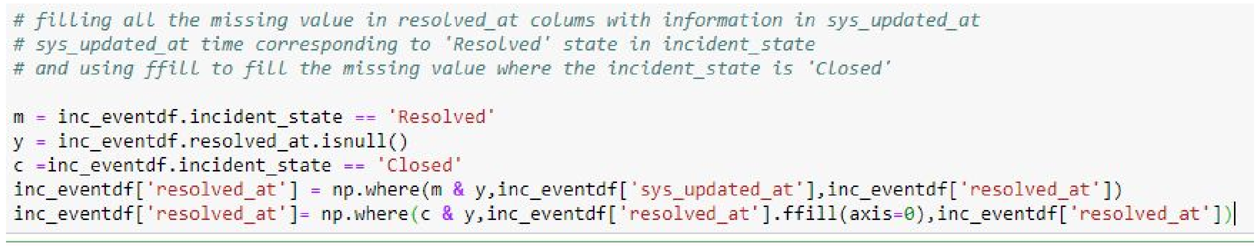
```
# to change all the relevent colums to datetime objects

col_name = ['sys_created_at','opened_at','sys_updated_at','resolved_at','closed_at']
for i in range(len(col_name)):
    inc_eventdf[col_name[i]] = pd.to_datetime(inc_eventdf[col_name[i]], dayfirst = True)
```

Next issue in our list was to correct information in the dataset. Such as the category “-100” in the “incident_state” column. Which was replaced based on subject matter knowledge.



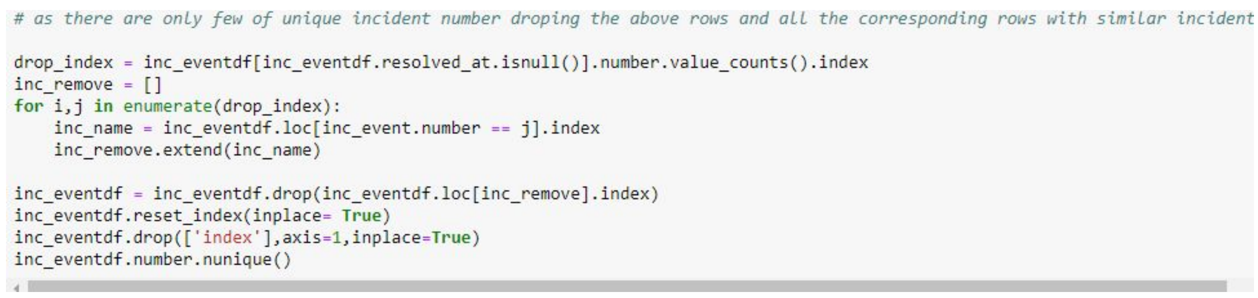
```
#  droped all the rows with incident_state = '-100' as there is only two incidents its effecting
inc_eventdf['incident_state'] =inc_eventdf['incident_state'].replace(to_replace = ['-100'], value = ['Active'])
```

Further action is required to address missing values in the dependent variable column “resolved_at” .



```
# filling all the missing value in resolved_at colums with information in sys_updated_at
# sys_updated_at time corresponding to 'Resolved' state in incident_state
# and using ffill to fill the missing value where the incident_state is 'Closed'

m = inc_eventdf.incident_state == 'Resolved'
y = inc_eventdf.resolved_at.isnull()
c =inc_eventdf.incident_state == 'Closed'
inc_eventdf['resolved_at'] = np.where(m & y,inc_eventdf['sys_updated_at'],inc_eventdf['resolved_at'])
inc_eventdf['resolved_at']= np.where(c & y,inc_eventdf['resolved_at'].ffill(axis=0),inc_eventdf['resolved_at'])
```

After performing all the above actions there are still some values that have missing entry, however they make up only 14 unique incidents out of 24918 hence are negligible and can be dropped.



```
# as there are only few of unique incident number droping the above rows and all the corresponding rows with similar incident

drop_index = inc_eventdf[inc_eventdf.resolved_at.isnull()].number.value_counts().index
inc_remove = []
for i,j in enumerate(drop_index):
    inc_name = inc_eventdf.loc[inc_event.number == j].index
    inc_remove.extend(inc_name)

inc_eventdf = inc_eventdf.drop(inc_eventdf.loc[inc_remove].index)
inc_eventdf.reset_index(inplace= True)
inc_eventdf.drop(['index'],axis=1,inplace=True)
inc_eventdf.number.nunique()
```