document describing the work done to clean and process the data.

- **ANALYSIS**

Dataset statistics

Number of variables	15
Number of observations	30205
Missing cells	6295
Missing cells (%)	1.4%
Duplicate rows	717
Duplicate rows (%)	2.4%

Main key points=

Dataset has 717 (2.4%) duplicate rows

`education` has 3024 (10.0%) missing values

`over_50k` has 3271 (10.8%) missing values

`id` is uniformly distributed

`capital_gain` has 27674 (91.6%) zeros

`capital_loss` has 28826 (95.4%) zeros

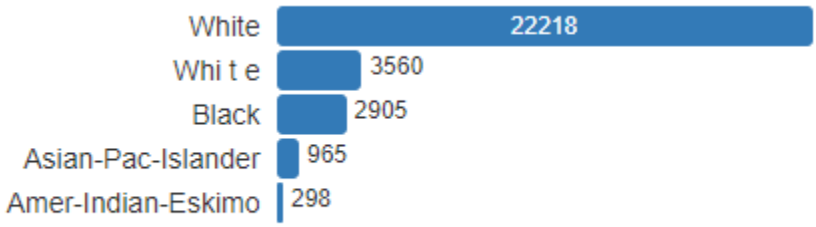
717 Rows are duplicated need to be removed

Target variable

Distinct count	2
Unique (%)	< 0.1%
Missing	3271
Missing (%)	10.8%
Memory size	236.1 KiB

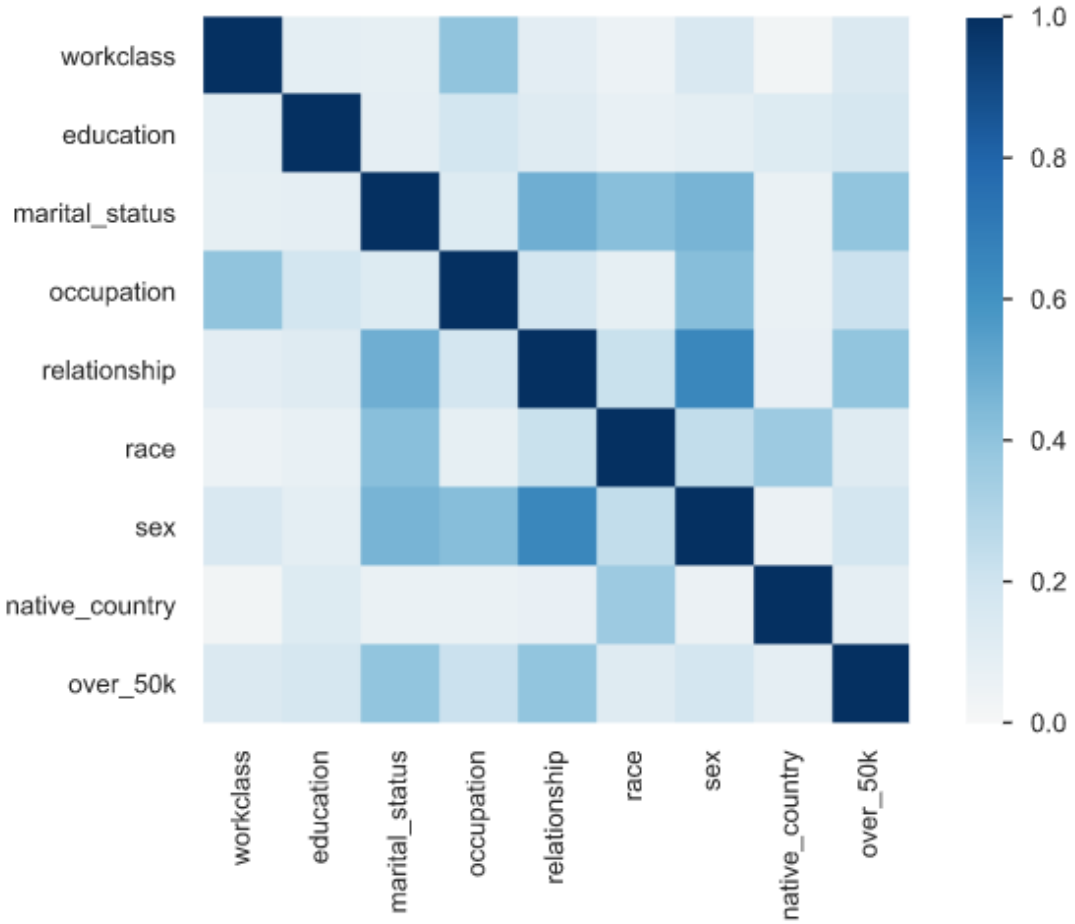
3271 Target variable are missing so needed to be removed

Race category



In Race category Whi t e need to be renamed

Relationship between given categorical variable



- **Cleaning steps –**

data loading

to load from your file change path in config.ini

```
[path]
InputPath = upwork_data_1.csv
OutputPath = clean_data.csv
```

raw_data = pd.read_csv(r"path\file.csv")

also check if the file has header if not than read this way

raw_data = pd.read_csv(r"path\file.csv", header=None)

```
raw_data = pd.read_csv(input_path)
print(raw_data.shape)
raw_data.head()
```

```
(30205, 15)
```

	id	age	workclass	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	nativ
0	12106	32	Private	HS-grad	9	Divorced	Adm-clerical	Other-relative	Whi t e	Female	0	0	40	Un
1	28951	43	State-gov	Some-college	10	Divorced	Adm-clerical	Unmarried	Whi t e	Female	0	0	40	Un

Checking and removing duplicate rows

```
duplicate=raw_data[raw_data.duplicated(subset=None, keep='first')]
print(duplicate.shape)
#duplicate
```

```
(717, 15)
```

```
unique_data=raw_data.drop_duplicates( keep='first', inplace=False)
print(unique_data.shape)
#unique_data

```

```
(29488, 15)
```

Removing na in Target variable

```
output_na=unique_data[unique_data['over_50k'].isna()]
output_na['over_50k'].isna().count()
```

```
3211
```

```
input_data=unique_data[~unique_data['over_50k'].isna()]
input_data.shape
```

```
(26277, 15)
```

This is the actual usable data for us - 26277 rows

correcting wrong label for white in race

```
input_data['race'][input_data.race == 'Whi t e'] = 'White'

print(input_data.groupby('race')['id'].count())
```

```
race
Amer-Indian-Eskimo      290
Asian-Pac-Islander      944
Black                  2837
Other                   251
White                 21955
Name: id, dtype: int64
```

dealing with wrong data of age “-1”

out of 2824 -1 values 2672 never married .
by checking distribution we have replaced these value with random age between 15-35.and the remaining null rows are dropped as 2821 out of 2824 are having target less than 50k (causing data imbalance)

Dealing with “?” in data

Converting them to Nan

```
input_data['native_country'][input_data.native_country == '?'] = np.nan
input_data['workclass'][input_data.workclass == '?'] = np.nan
input_data['occupation'][input_data.occupation == '?'] = np.nan
```

Dealing with nan values in data

dropping the nan data ROWS with target variable less than 50k (causing data imbalance)

```
print(input_data['id'].count())
dropped_data=input_data[~
    ((
        (input_data['occupation'].isnull())
           |  (input_data['workclass'].isnull() )
           |  (input_data['native_country'].isnull())
           |  (input_data['education'].isnull())
    )

       |
        & (input_data['over_50k']=='<=50K'))
]
dropped_data['id'].count()
```

```
26127
```

```
22185
```

Dropping off the common row in workclass and occupation NA

As they are only 101 and its better to drop as there are 4 variable(these 2 ,education,native_country) to fill

```
print(input_data['id'].count())
dropped_data=dropped_data[~
    ((   dropped_data['occupation'].isnull())
          & (dropped_data['workclass'].isnull() ) )
]
dropped_data['id'].count()
```

```
26127
```

```
22084
```

fill the remaining nan value with mode

```
print("nan left",dropped_data.isna().sum().sum())
mod=dropped_data['native_country'].mode()
dropped_data['native_country'][dropped_data.native_country.isnull()] = 'United-States'
print("nan left",dropped_data.isna().sum().sum())

dropped_data = dropped_data.fillna(dropped_data['education'].value_counts().index[0])
print("nan left",dropped_data.isna().sum().sum())
```

```
nan left 459
nan left 395
nan left 0
```

saving data to csv

```
data.to_csv(output_path,index=False)
```

to save data file change path in config.ini

or **just put your pathname\filename.csv like**

```
data.to_csv(r'path\filename.csv',index=False)
```

```
data.to_csv(r'D:\Shivam\Upwork\JoleneMartin\1\data\processed\datacleaned.csv',index=False)
```