Adult Salary Prediction

Presented by:

Omran Fallatah

Ghanim Alghanim

Dataset

Dataset Description

- The dataset is credited to Ronny Kohavi and Barry Becker and was drawn from the 1994 United States Census Bureau data and involves using personal details such as education level to predict whether an individual will earn more or less than $50,000 per year.

- The task is to predict whether a given adult makes more than $50,000 a year-based attributes such as education, hours of work per week, etc.

Dataset

The dataset was collected from UCI machine learning repository

The dataset provides ~50,000 observations and 14 input variables that are a mixture of categorical, ordinal, and numerical data types. The complete list of variables is as follows:

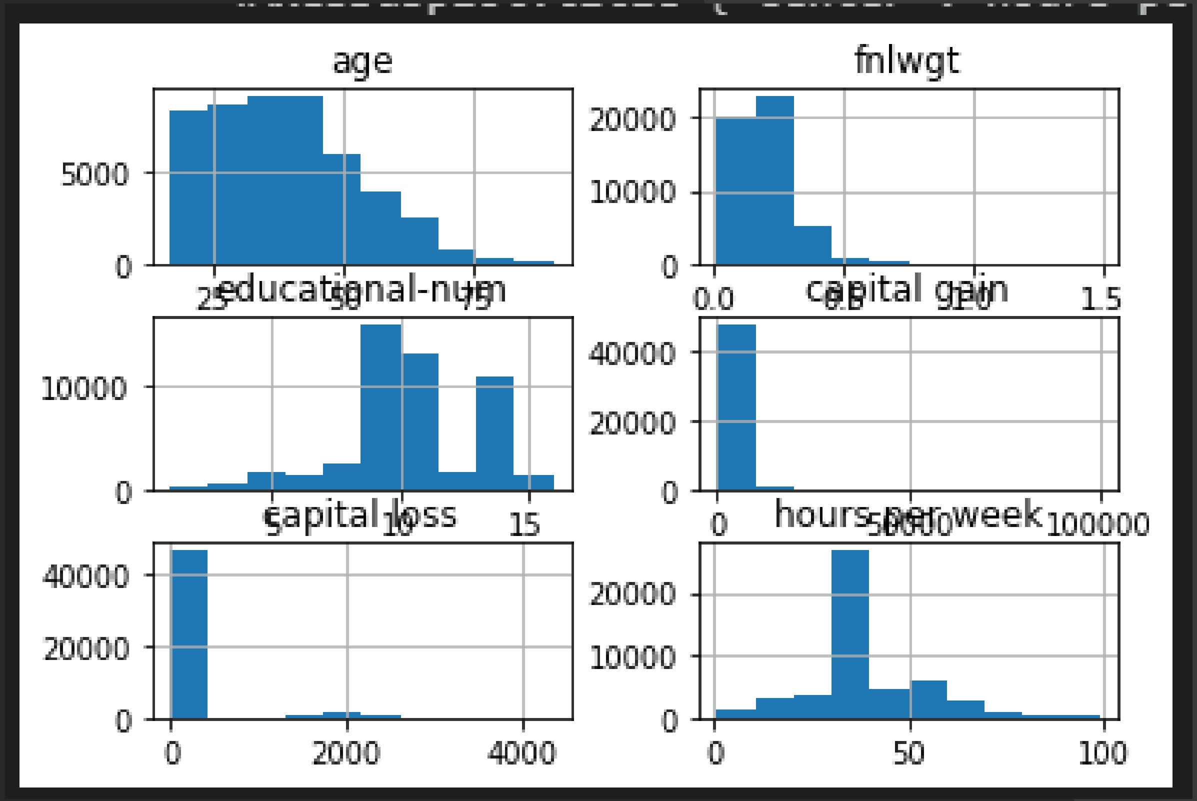
- Age.
- Workclass.
- Final Weight.
- Education.
- Education Number of Years.
- Marital-status.
- Occupation.
- Relationship.
- Race.
- Sex.
- Capital-gain.
- Capital-loss.
- Hours-per-week.
- Native-country.

Target filed: Income

EDA

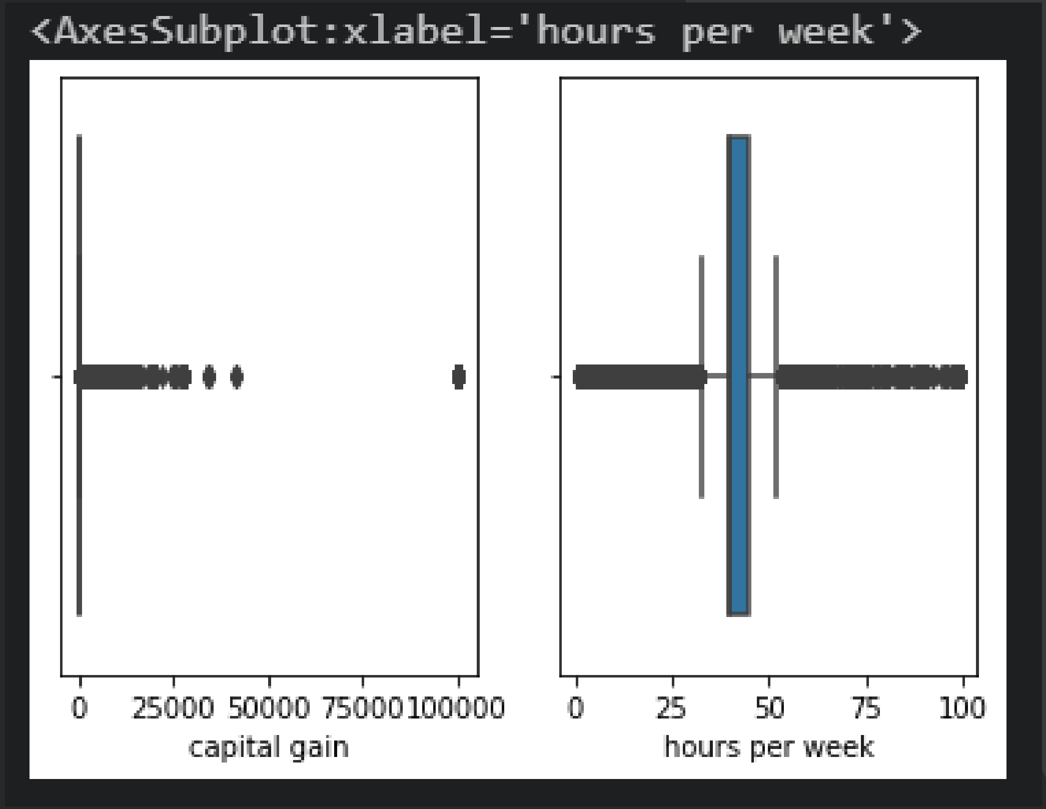
Exploratory Data Analysis

- We can see many different distributions, some with Gaussian-like distributions, others with seemingly exponential or discrete distributions. We can also see that they all appear to have a very different scale.
- Depending on the choice of modeling algorithms, we would expect scaling the distributions to the same range to be useful, and perhaps the use of some power transforms



Data Cleaning & Feature Engineering

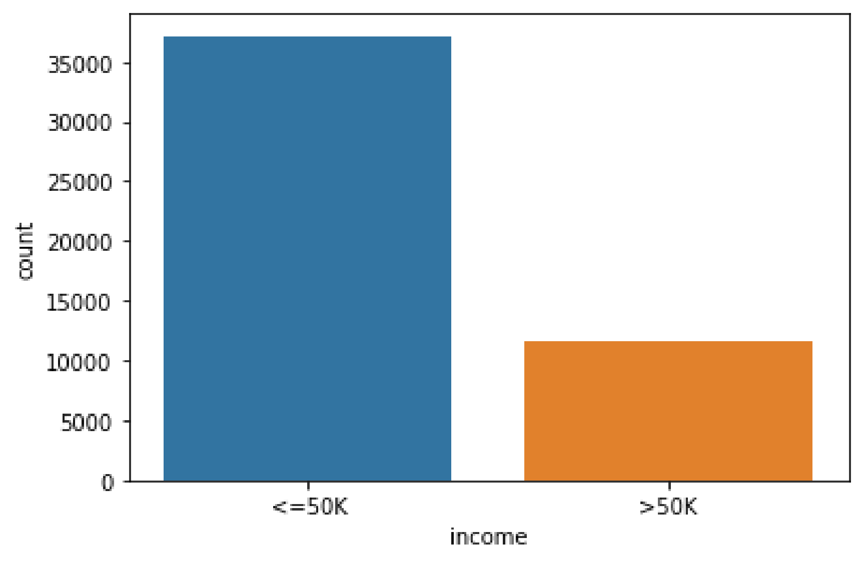
- A lot of outliers
- Convert categorical into maps and one hot encoding
- Dropping missing data that can't be filled due to lack of data.



Pre-Model Fitting

- Unbalanced data
- Used SMOTE to fill the least filled target
- Used MinMaxScaler to normalize the data

```
244 outlier in the capital-gain
137 outlier in the hours-per-week
<=50K    0.760718
>50K     0.239282
Name: income, dtype: float64
```



Pre-Model Fitting

- Our model now consists of ~69000 observation
- split into 2 arrays one for training and testing : ration 80/20 % respectively
- Scaled between 0-1

```
X_up_train.shape (59448, 26)
X_test.shape (9769, 26)
y_up_train.value_counts()

0    29724
1    29724
Name: income, dtype: int64
```

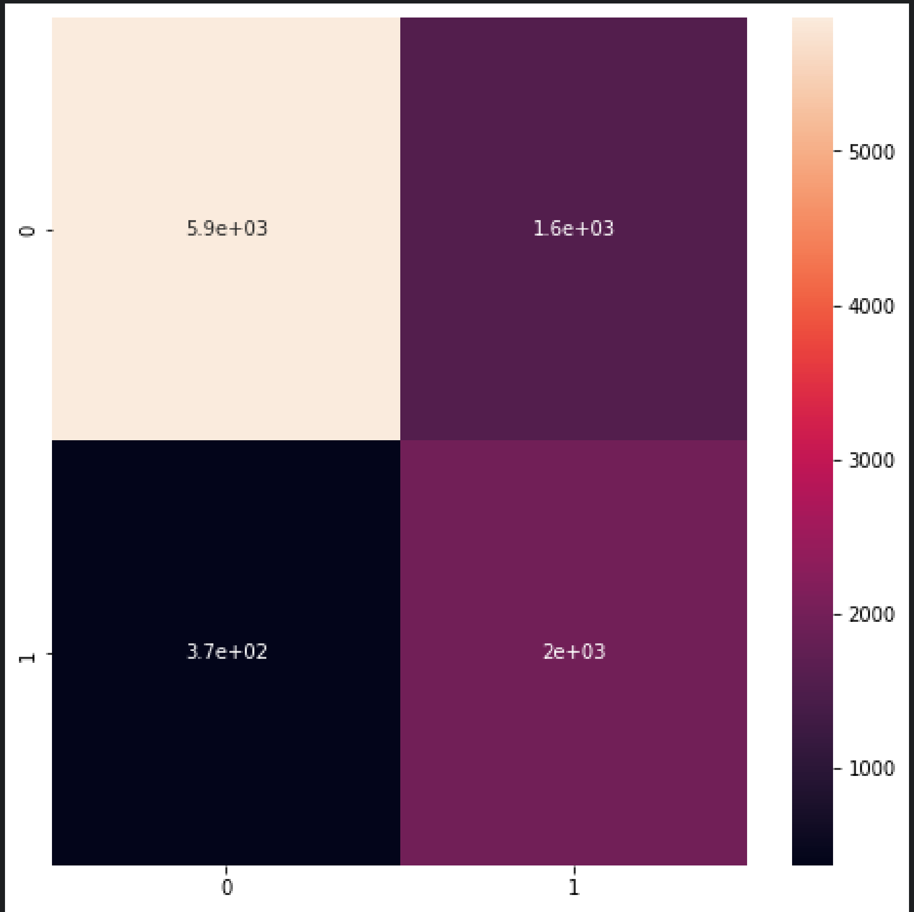
Models

Logistic Regression
Decision Tree

Best model esitmator LogisticRegression(C=10, solver='liblinear')

Training Set Accuracy Score: 0.82

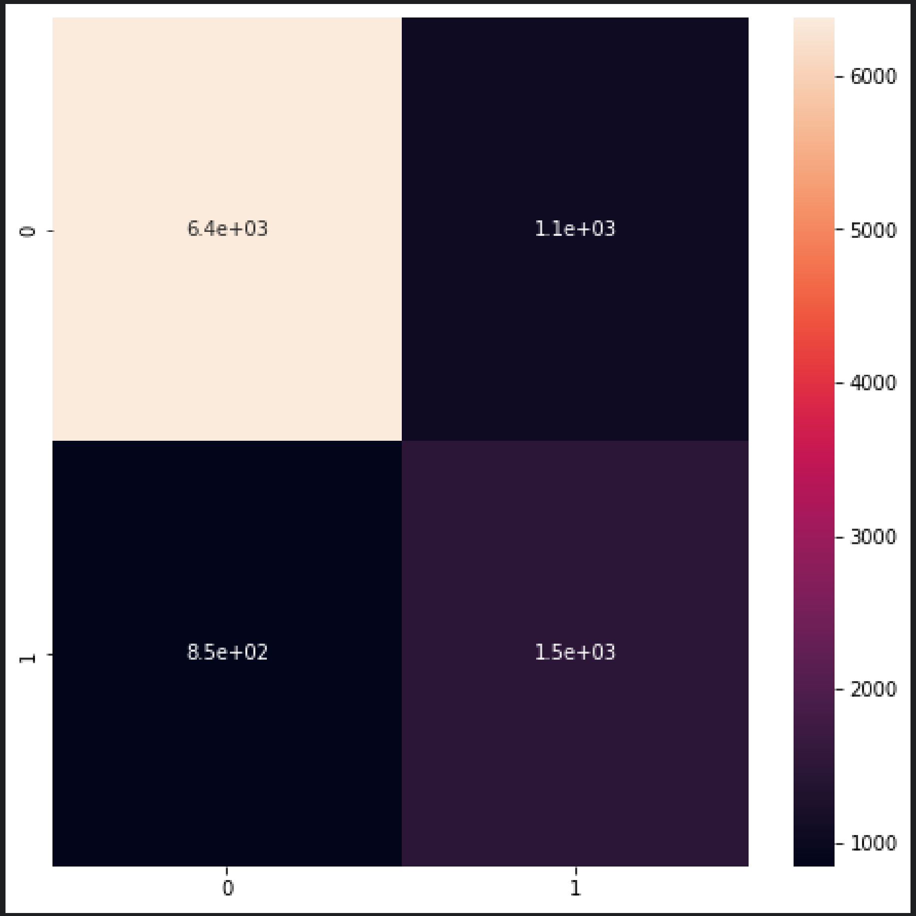
Testing Set Accuracy Score: 0.80



Best model esitmator DecisionTreeClassifier()

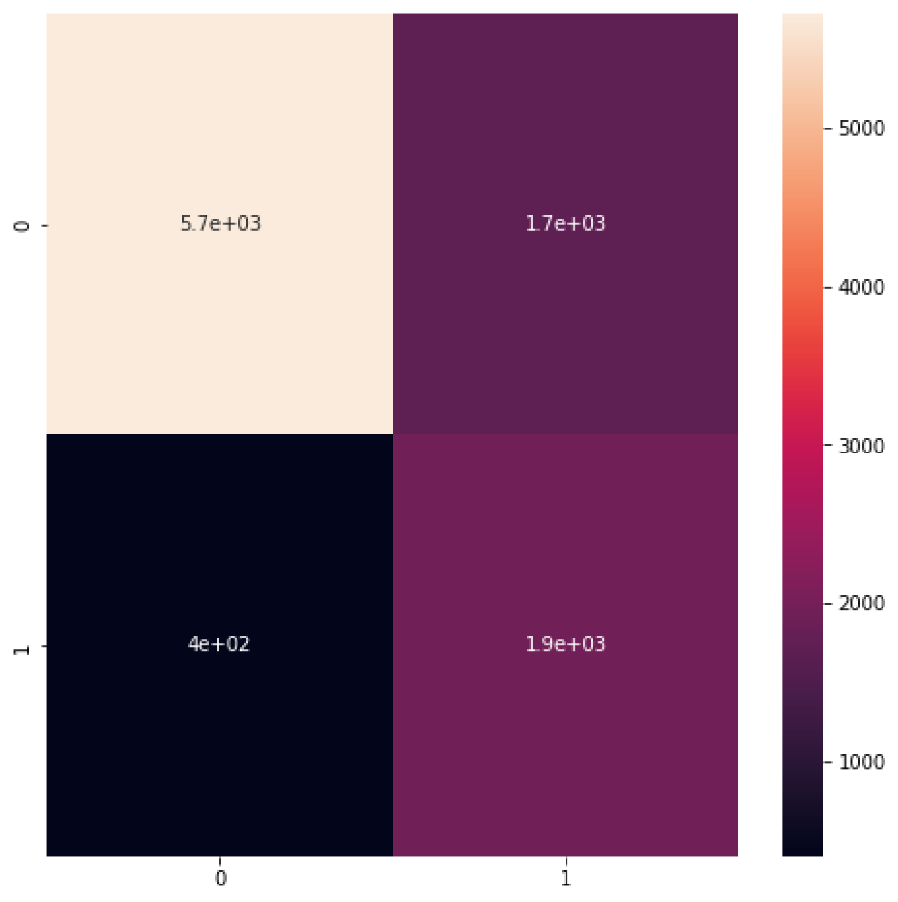
Training Set Accuracy Score: 1.00

Testing Set Accuracy Score: 0.81

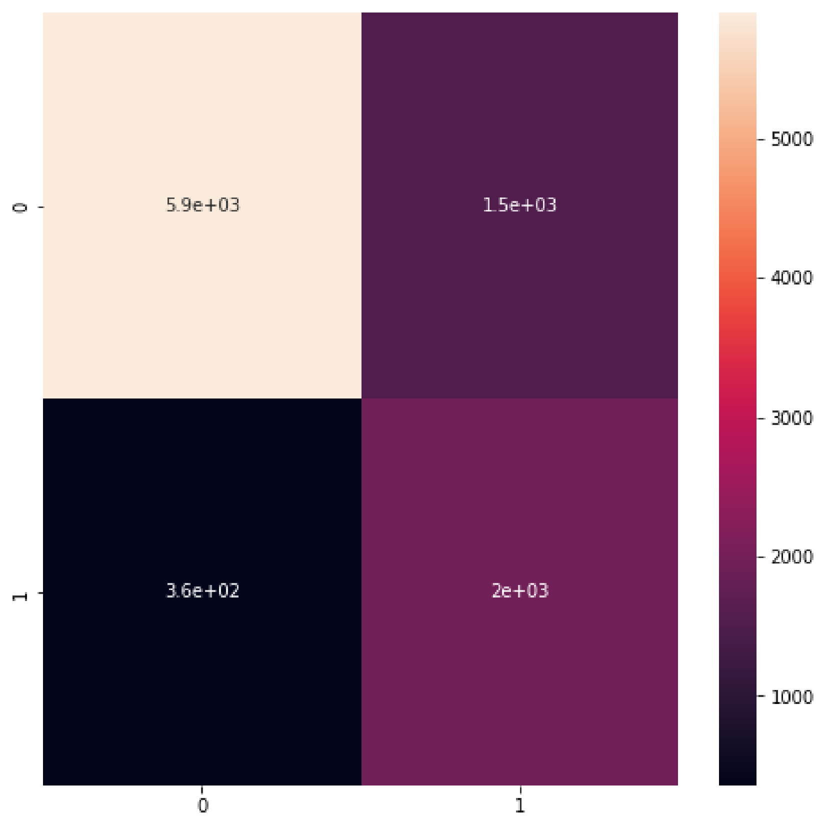


Random Forest
K-Nearest Neighbors

```
Best model esitmator KNeighborsClassifier(n_neighbors=23)
Training Set Accuracy Score: 0.85
Testing Set Accuracy Score: 0.78
```



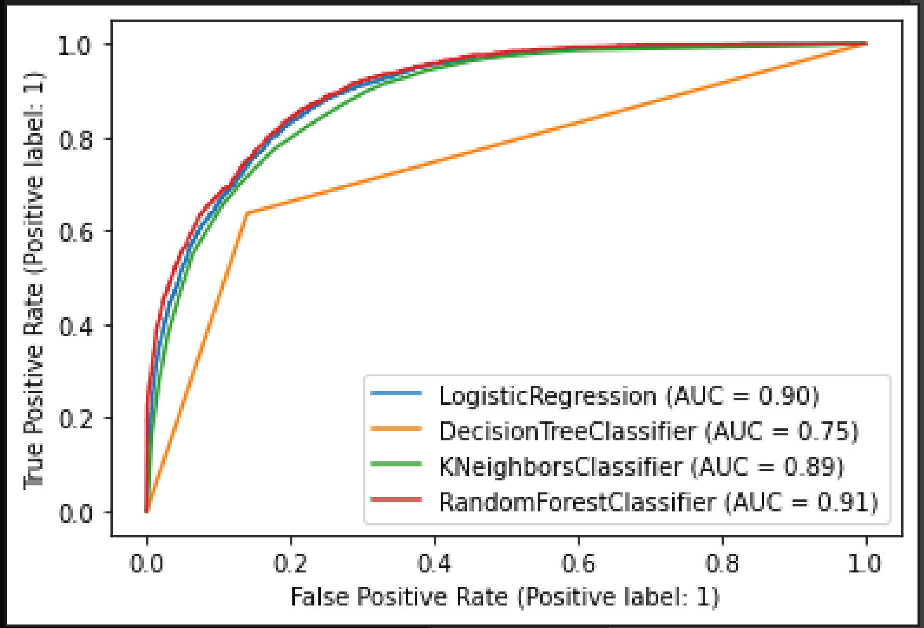

```
Best model esitmator RandomForestClassifier(max_depth=9, max_features=9, random_state=0
Training Set Accuracy Score: 0.86
Testing Set Accuracy Score: 0.81
```



Cross Validation with 10 K folds

- Best Model is Random Forest
- with best R Squared Score 0.85

```
Logistic Regression    0.791900
Decision Tree          0.813890
KNN                    0.795749
Random Forest          0.854981
Name: CV Mean, dtype: float64
```



Prediction example

```
#gender    / 'Female':0, 'Male':1
#race   /   'White':0, 'Black':1, 'Asian-Pac-Islander':2, 'Amer-Indian-Eskimo':3
#marital  / 'Widowed':0, 'Divorced':1, 'Separated':2,'Never-married':3,'Married-civ-spouse':4, 'Married-spouse-absent':5, 'Married-AF-spouse':6
#relationship / 'Not-in-family':0, 'Unmarried':0, 'Own-child':0, 'Other-relative':0,'Husband':1, 'Wife':1
#workclass /  ?':0, 'Private':1, 'State-gov':2, 'Federal-gov':3,'Self-emp-not-inc':4, 'Self-emp-inc': 5, 'Local-gov': 6,'Without-pay':7, 'Never-worked':8
#country    /    x:    x == "United-States" 1 else 0)
#income     /    <=50K':0, '>50K': 1
X['country']
random_for.predict(np.array([20,6,4856,9,0,1,1,100,0,15,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0]).reshape(1, -1))
```

```
array([1], dtype=int64)
```

```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 26 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   age                 48842 non-null  int64  
 1   workclass           48842 non-null  float64
 2   fnlwgt              48842 non-null  int64  
 3   educational-num     48842 non-null  int64  
 4   marital             48842 non-null  int64  
 5   relationship        48842 non-null  int64  
 6   race                48842 non-null  float64
 7   gender              48842 non-null  int64  
 8   capital gain        48842 non-null  float64
 9   capital loss        48842 non-null  int64  
 10  hours per week      48842 non-null  float64
 11  country             48842 non-null  int64  
 12  Adm-clerical        48842 non-null  uint8  
 13  Armed-Forces        48842 non-null  uint8  
 14  Craft-repair        48842 non-null  uint8  
 15  Exec-managerial     48842 non-null  uint8  
 16  Farming-fishing     48842 non-null  uint8  
 17  Handlers-cleaners   48842 non-null  uint8  
 18  Machine-op-inspct   48842 non-null  uint8  
 19  Other-service       48842 non-null  uint8  
 20  Priv-house-serv     48842 non-null  uint8  
 21  Prof-specialty      48842 non-null  uint8  
 22  Protective-serv     48842 non-null  uint8  
 23  Sales               48842 non-null  uint8  
 24  Tech-support        48842 non-null  uint8  
 25  Transport-moving    48842 non-null  uint8  
dtypes: float64(4), int64(8), uint8(14)
memory usage: 5.1 MB
```

Thank you

Questions ?