Data Science:
Term Project Specification

Term Project Requirements (1/4)

- The term project will be a team project.
- Prepare a short proposal and post it to CyberCampus.
  - Your proposal must include a statistical description of the dataset, project.
- Final presentation will be in the last class before the Final Exam.
- Must apply every step of the end-to-end Big Data process (except data curation and deployment)

Term Project Requirements (2/4)

- Study the list of possible datasets and the sample Case Study (using the WorldCup dataset) to get a feel for how to proceed.
- You may select a dataset from the list provided or find a suitable dataset on your own.
- However, for education purpose, the dataset must include
  - a reasonable number of records and features (attributes)
  - a reasonable amount of dirty data
  - a combination of numerical data and categorical data.

Term Project Requirements (3/4)

- Algorithms and Learning Models
  - You must use data scaling and encoding
  - You must use classification algorithm
  - You must use 1 of the following 2 types of algorithm
    - regression, clustering
  - You must NOT use learning models not taught in this course.
- Evaluation
  - You must use k-fold cross validation for testing classification models.
- Note 1: Term project grading will be based on the following:
  - Relevance (i.e. usefulness, practicality) and originality of the project objective
  - Demonstrating how well you have learned in this course.
  - (bonus) demonstrating things you have learned on your own (except the learning models).
- Note 2: Open Source software contribution to the community (explained shortly)

Term Project Requirements (4/4)

- Term project submission package
  - PPT presentation
  - Separate writeup that gives details behind the PPT presentation
  - Source code with detailed comments
  - Explanations for all modules/classes/libraries/functions/methods (along with the parameters) used that were not taught or used in Lab classes (* This is to prevent students from just copying code found on the Internet without actually learning anything. *)
  - Outputs (including plots, code execution results)
  - Dataset used
  - Peer review
- You may make use of code found on the Internet (blogs, Kaggle, GitHub, etc.).
  - However, in that case, you MUST cite the sources. Failure to cite them constitutes plagiarism.

Open Source SW Contribution (1/2)

- For a given cleaned dataset, do the following under a single top-level function (rather than repeating the same code many times).
- 1. Preprocessing
  - combination of various data scaling and categorical features encoding methods.
- 2. Learning Model training and testing
  - Different models(algorithms), combination of model parameters for each model
  - Different evaluation methods for each of the above.
- Find the top five and best combination of the above.

Open Source SW Contribution (2/2)

- Write the function and user manual/specification in the style of Pandas and Scikit-learn.
- Post it to GitHub or Kaggle

Dataset Finder

- Google dataset search
  https://toolbox.google.com/datasetsearch

- Kaggle
  https://www.kaggle.com/datasets
  examples)
  https://www.kaggle.com/residentmario/ramen-ratings
  https://www.kaggle.com/ncaa/ncaa-basketball
  https://www.kaggle.com/aaronschlegel/seattle-pet-licenses

- UCI Machine Learning Repository
  http://mlr.cs.umass.edu/ml/

- VisualData
  https://www.visualdata.io

- Find Datasets | CMU Libraries
  https://guides.library.cmu.edu/machine-learning/datasets

General Datasets: Public Government Datasets

- data.gov (https://www.data.gov)
- Food Environment Atlas
  https://catalog.data.gov/dataset/food-environment-atlas-f4a22
- School System Finances
  https://catalog.data.gov/dataset/annual-survey-of-school-system-finances
- The US National Center for Education Statistics
  https://nces.ed.gov
- The UK Data Service
  https://www.ukdataservice.ac.uk
- Data USA
  https://datausa.io

General Datasets: Housing Datasets

- Boston Housing Dataset
  https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html
  http://lib.stat.cmu.edu/datasets/boston

General Datasets: Finance & Economics Datasets

- Quandl
  https://www.quandl.com
- World Bank Open Data
  https://data.worldbank.org
- IMF Data
  https://www.imf.org/en/Data
- Financial Times Market Data
  https://markets.ft.com/data/
- Google Trends
  https://trends.google.com/trends/?q=google&ctab=0&geo=all&date=all&sort=0
- American Economic Association (AEA)
  https://www.aeaweb.org/resources/data/us-macro-regional

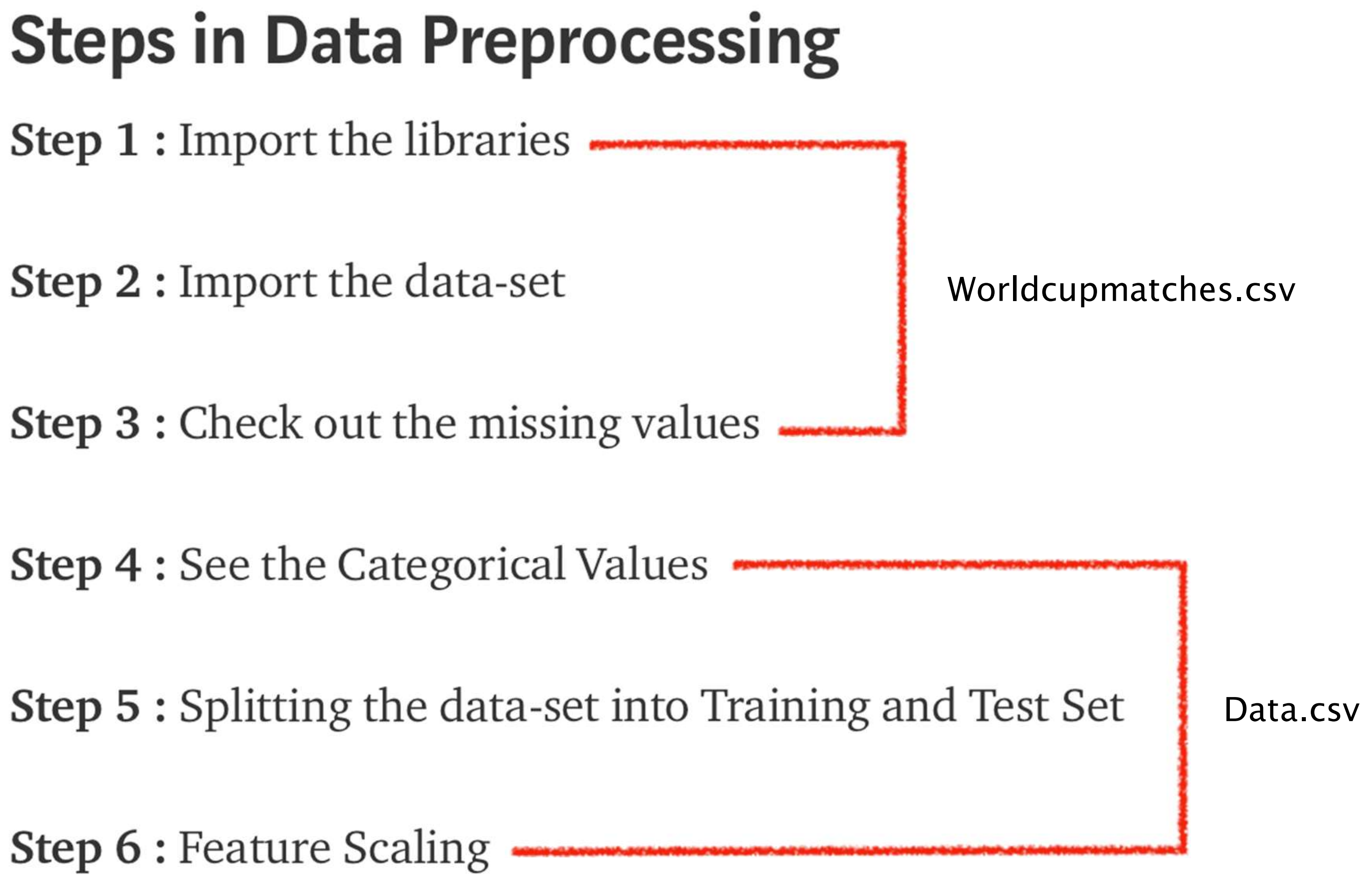
Machine Learning Datasets: Sentiment Analysis Datasets

- Multidomain sentiment analysis dataset
  http://www.cs.jhu.edu/~mdredze/datasets/sentiment/
- IMDB reviews
  http://ai.stanford.edu/~amaas/data/sentiment/
- Stanford sentiment Treebank
  https://nlp.stanford.edu/sentiment/code.html
- Sentiment140
  http://help.sentiment140.com/for-students/
- Twitter US Airline Sentiment
  https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Case Study for Term Project

The FIFA World Cup

Index



Steps in Data Preprocessing

Step 1 : Import the libraries

Step 2 : Import the data-set

Step 3 : Check out the missing values

Step 4 : See the Categorical Values

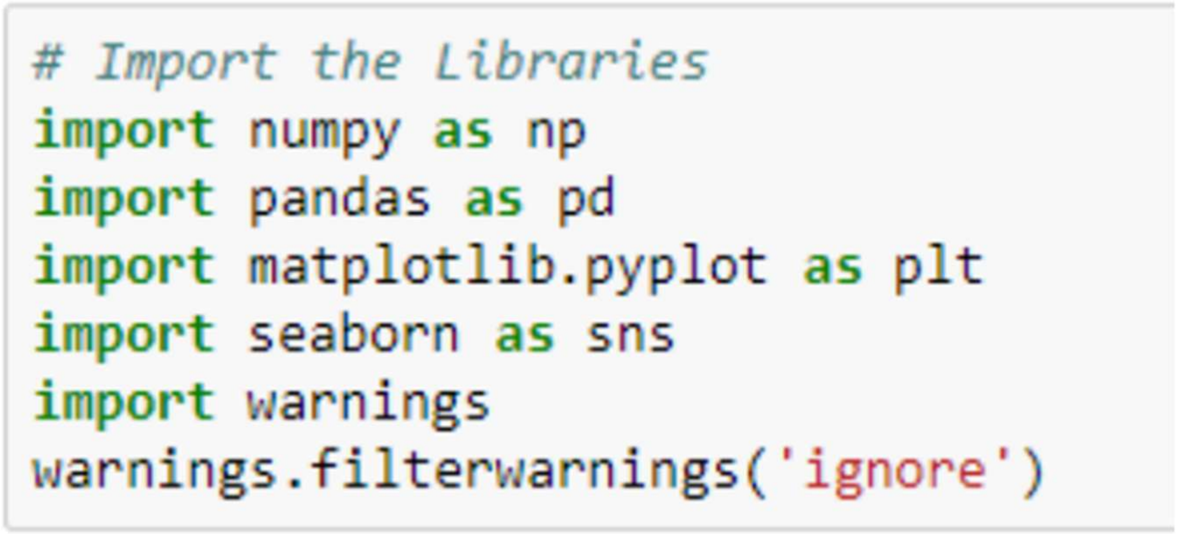
Step 5 : Splitting the data-set into Training and Test Set

Step 6 : Feature Scaling

Worldcupmatches.csv

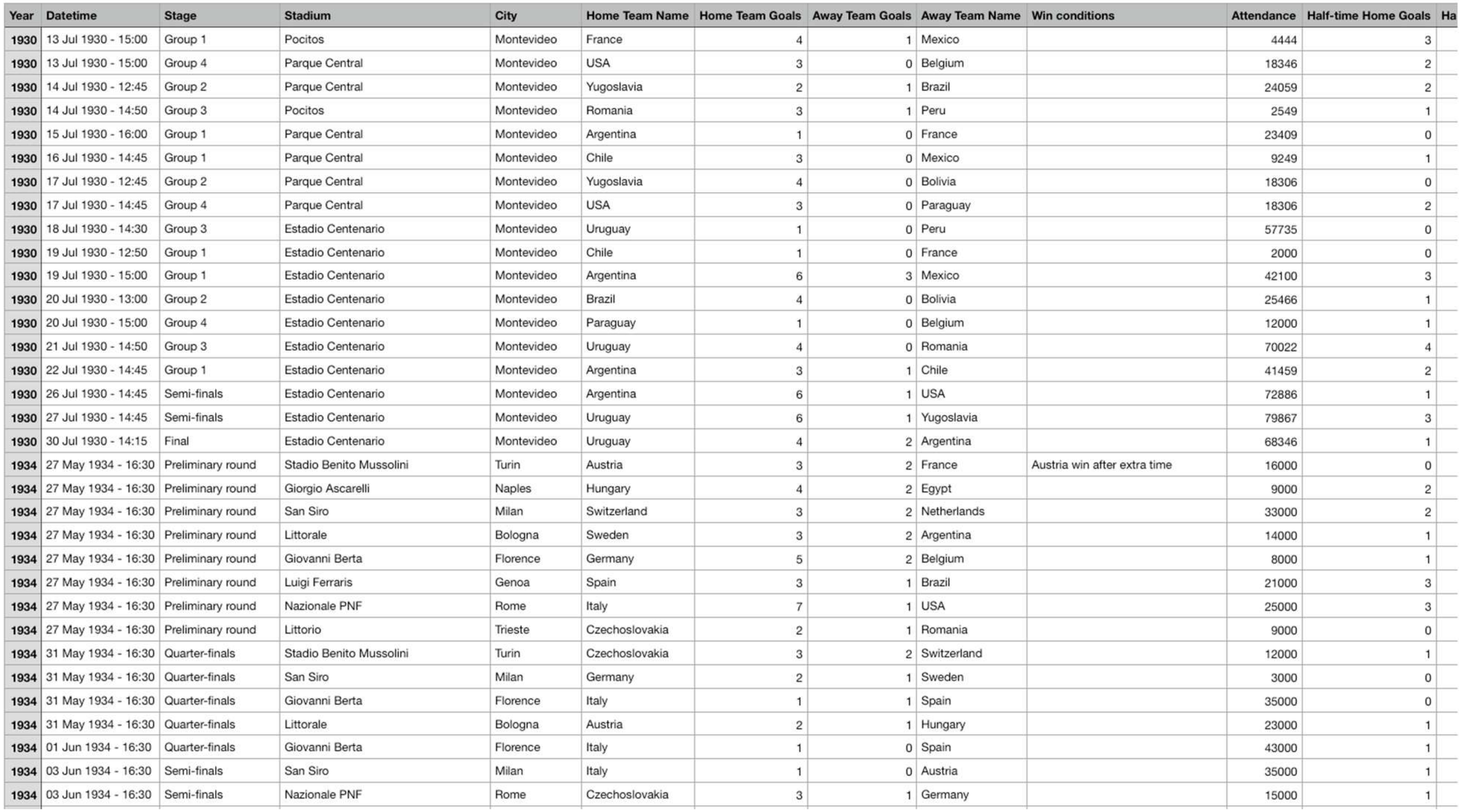
Data.csv

Step 1. Import the Libraries



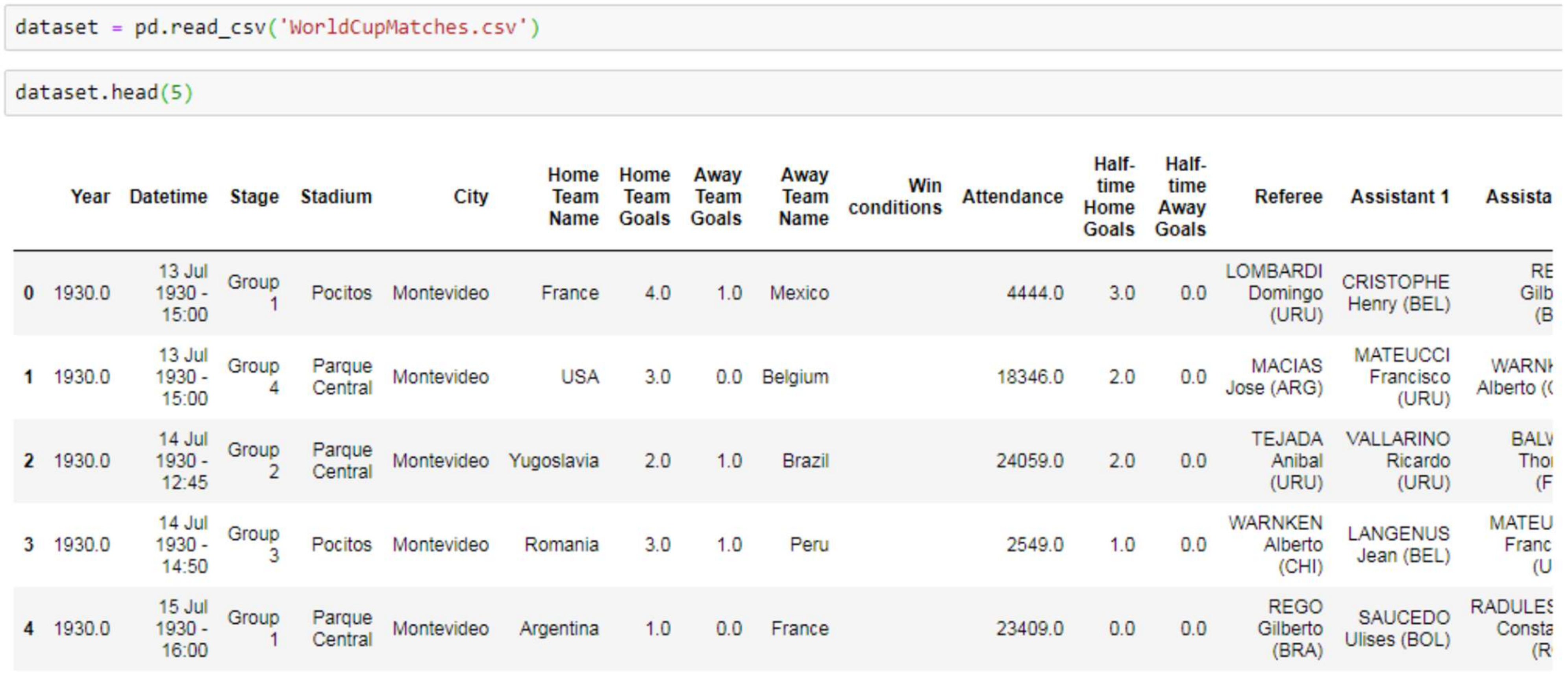
```python
# Import the Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Step 2. Dataset



Year	Datetime	Stage	Stadium	City	Home Team Name	Home Team Goals	Away Team Goals	Away Team Name	Win conditions	Attendance	Half-time Home Goals	Ha
1930	13 Jul 1930 - 15:00	Group 1	Pocitos	Montevideo	France	4	1	Mexico		4444	3	
1930	13 Jul 1930 - 15:00	Group 4	Parque Central	Montevideo	USA	3	0	Belgium		18346	2	
1930	14 Jul 1930 - 12:45	Group 2	Parque Central	Montevideo	Yugoslavia	2	1	Brazil		24059	2	
1930	14 Jul 1930 - 14:50	Group 3	Pocitos	Montevideo	Romania	3	1	Peru		2549	1	
1930	15 Jul 1930 - 16:00	Group 1	Parque Central	Montevideo	Argentina	1	0	France		23409	0	
1930	16 Jul 1930 - 14:45	Group 1	Parque Central	Montevideo	Chile	3	0	Mexico		9249	1	
1930	17 Jul 1930 - 12:45	Group 2	Parque Central	Montevideo	Yugoslavia	4	0	Bolivia		18306	0	
1930	17 Jul 1930 - 14:45	Group 4	Parque Central	Montevideo	USA	3	0	Paraguay		18306	2	
1930	18 Jul 1930 - 14:30	Group 3	Estadio Centenario	Montevideo	Uruguay	1	0	Peru		57735	0	
1930	19 Jul 1930 - 12:50	Group 1	Estadio Centenario	Montevideo	Chile	1	0	France		2000	0	
1930	19 Jul 1930 - 15:00	Group 1	Estadio Centenario	Montevideo	Argentina	6	3	Mexico		42100	3	
1930	20 Jul 1930 - 13:00	Group 2	Estadio Centenario	Montevideo	Brazil	4	0	Bolivia		25466	1	
1930	20 Jul 1930 - 15:00	Group 4	Estadio Centenario	Montevideo	Paraguay	1	0	Belgium		12000	1	
1930	21 Jul 1930 - 14:50	Group 3	Estadio Centenario	Montevideo	Uruguay	4	0	Romania		70022	4	
1930	22 Jul 1930 - 14:45	Group 1	Estadio Centenario	Montevideo	Argentina	3	1	Chile		41459	2	
1930	26 Jul 1930 - 14:45	Semi-finals	Estadio Centenario	Montevideo	Argentina	6	1	USA		72886	1	
1930	27 Jul 1930 - 14:45	Semi-finals	Estadio Centenario	Montevideo	Uruguay	6	1	Yugoslavia		79867	3	
1930	30 Jul 1930 - 14:15	Final	Estadio Centenario	Montevideo	Uruguay	4	2	Argentina		68346	1	
1934	27 May 1934 - 16:30	Preliminary round	Stadio Benito Mussolini	Turin	Austria	3	2	France	Austria win after extra time	16000	0	
1934	27 May 1934 - 16:30	Preliminary round	Giorgio Ascarelli	Naples	Hungary	4	2	Egypt		9000	2	
1934	27 May 1934 - 16:30	Preliminary round	San Siro	Milan	Switzerland	3	2	Netherlands		33000	2	
1934	27 May 1934 - 16:30	Preliminary round	Littorale	Bologna	Sweden	3	2	Argentina		14000	1	
1934	27 May 1934 - 16:30	Preliminary round	Giovanni Berta	Florence	Germany	5	2	Belgium		8000	1	
1934	27 May 1934 - 16:30	Preliminary round	Luigi Ferraris	Genoa	Spain	3	1	Brazil		21000	3	
1934	27 May 1934 - 16:30	Preliminary round	Nazionale PNF	Rome	Italy	7	1	USA		25000	3	
1934	27 May 1934 - 16:30	Preliminary round	Littorio	Trieste	Czechoslovakia	2	1	Romania		9000	0	
1934	31 May 1934 - 16:30	Quarter-finals	Stadio Benito Mussolini	Turin	Czechoslovakia	3	2	Switzerland		12000	1	
1934	31 May 1934 - 16:30	Quarter-finals	San Siro	Milan	Germany	2	1	Sweden		3000	0	
1934	31 May 1934 - 16:30	Quarter-finals	Giovanni Berta	Florence	Italy	1	1	Spain		35000	0	
1934	31 May 1934 - 16:30	Quarter-finals	Littorale	Bologna	Austria	2	1	Hungary		23000	1	
1934	01 Jun 1934 - 16:30	Quarter-finals	Giovanni Berta	Florence	Italy	1	0	Spain		43000	1	
1934	03 Jun 1934 - 16:30	Semi-finals	San Siro	Milan	Italy	1	0	Austria		35000	1	
1934	03 Jun 1934 - 16:30	Semi-finals	Nazionale PNF	Rome	Czechoslovakia	3	1	Germany		15000	1	

Step 2. Dataset (cont'd)

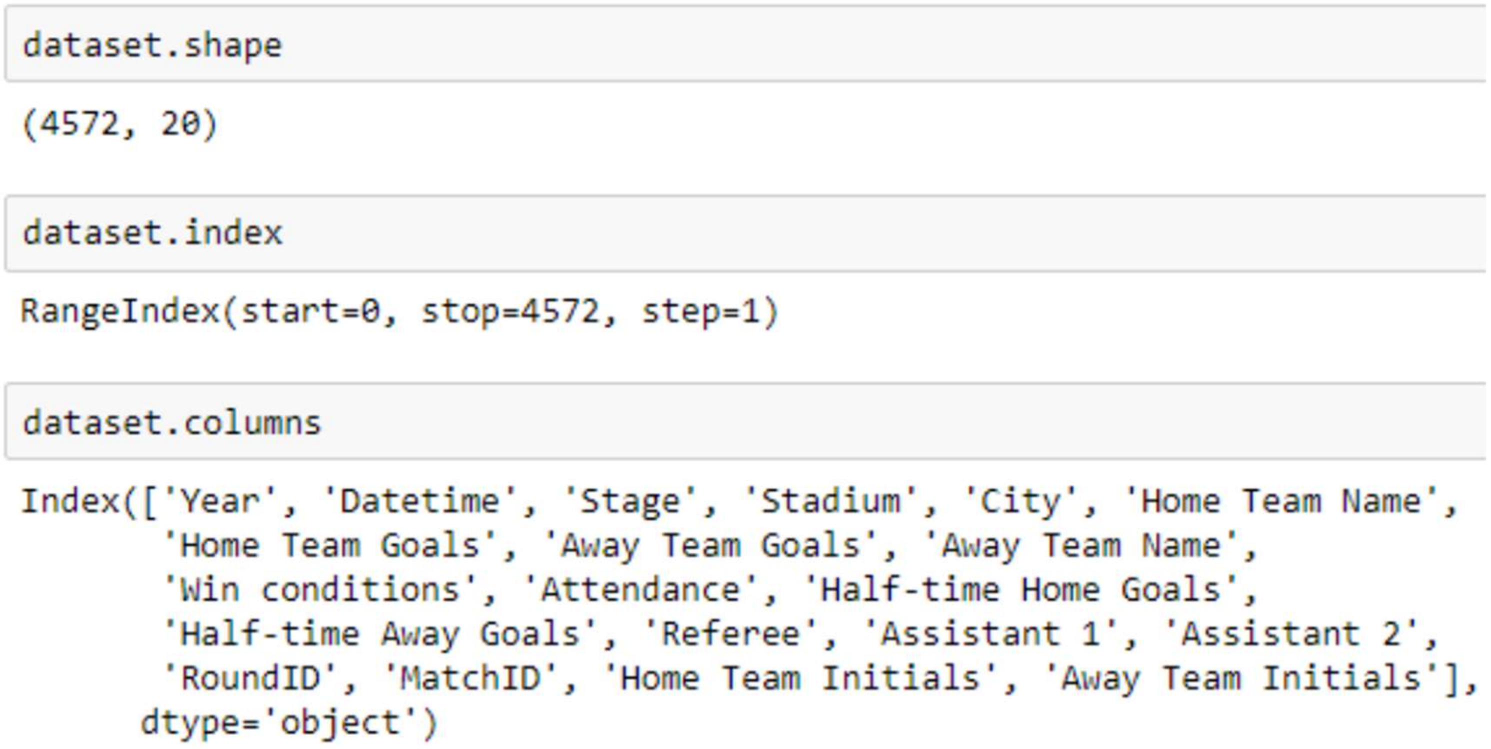


```
dataset = pd.read_csv('WorldCupMatches.csv')
```

```
dataset.head(5)
```

	Year	Datetime	Stage	Stadium	City	Home Team Name	Home Team Goals	Away Team Goals	Away Team Name	Win conditions	Attendance	Half-time Home Goals	Half-time Away Goals	Referee	Assistant 1	Assista
0	1930.0	13 Jul 1930 - 15:00	Group 1	Pocitos	Montevideo	France	4.0	1.0	Mexico		4444.0	3.0	0.0	LOMBARDI Domingo (URU)	CRISTOPHE Henry (BEL)	RE Gilb (B
1	1930.0	13 Jul 1930 - 15:00	Group 4	Parque Central	Montevideo	USA	3.0	0.0	Belgium		18346.0	2.0	0.0	MACIAS Jose (ARG)	MATEUCCI Francisco (URU)	WARNI Alberto (
2	1930.0	14 Jul 1930 - 12:45	Group 2	Parque Central	Montevideo	Yugoslavia	2.0	1.0	Brazil		24059.0	2.0	0.0	TEJADA Anibal (URU)	VALLARINO Ricardo (URU)	BALV Thoi (F
3	1930.0	14 Jul 1930 - 14:50	Group 3	Pocitos	Montevideo	Romania	3.0	1.0	Peru		2549.0	1.0	0.0	WARNKEN Alberto (CHI)	LANGENUS Jean (BEL)	MATEU Franc (U
4	1930.0	15 Jul 1930 - 16:00	Group 1	Parque Central	Montevideo	Argentina	1.0	0.0	France		23409.0	0.0	0.0	REGO Gilberto (BRA)	SAUCEDO Ulises (BOL)	RADULES Consta (R

Step 2. Dataset (cont'd)



```
dataset.shape
```

```
(4572, 20)
```

```
dataset.index
```

```
RangeIndex(start=0, stop=4572, step=1)
```

```
dataset.columns
```

```
Index(['Year', 'Datetime', 'Stage', 'Stadium', 'City', 'Home Team Name',
       'Home Team Goals', 'Away Team Goals', 'Away Team Name',
       'Win conditions', 'Attendance', 'Half-time Home Goals',
       'Half-time Away Goals', 'Referee', 'Assistant 1', 'Assistant 2',
       'RoundID', 'MatchID', 'Home Team Initials', 'Away Team Initials'],
      dtype='object')
```

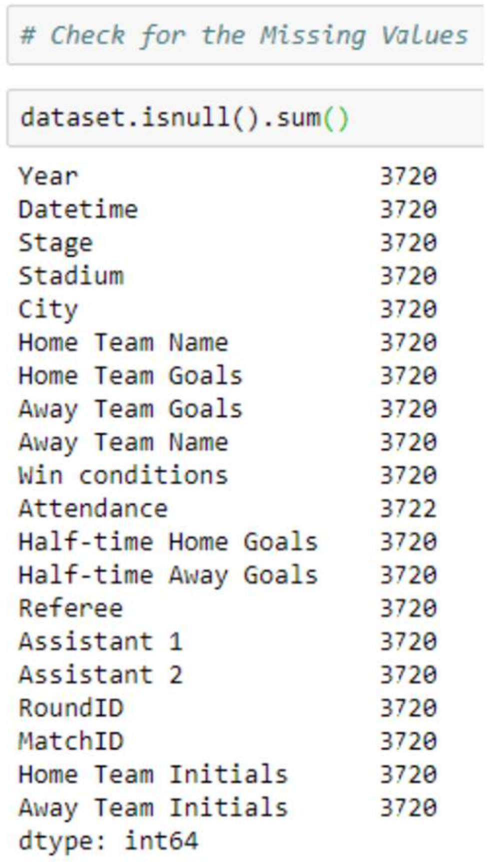
Step 3. Missing Values

- Two ways to handle missing values

1. Delete a particular row/column if there are enough samples in data set
   (removing the data will lead to loss of information)
2. calculate mean, median, or mode of the feature and use it to replace the missing values

Step 3. Missing Values (cont'd)

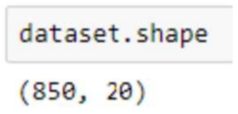
- Delete the missing values



```
# Check for the Missing Values
```

```
dataset.isnull().sum()
```

```
Year                    3720
Datetime                3720
Stage                   3720
Stadium                 3720
City                    3720
Home Team Name          3720
Home Team Goals         3720
Away Team Goals         3720
Away Team Name          3720
Win conditions          3720
Attendance              3722
Half-time Home Goals    3720
Half-time Away Goals    3720
Referee                 3720
Assistant 1             3720
Assistant 2             3720
RoundID                 3720
MatchID                 3720
Home Team Initials      3720
Away Team Initials      3720
dtype: int64
```

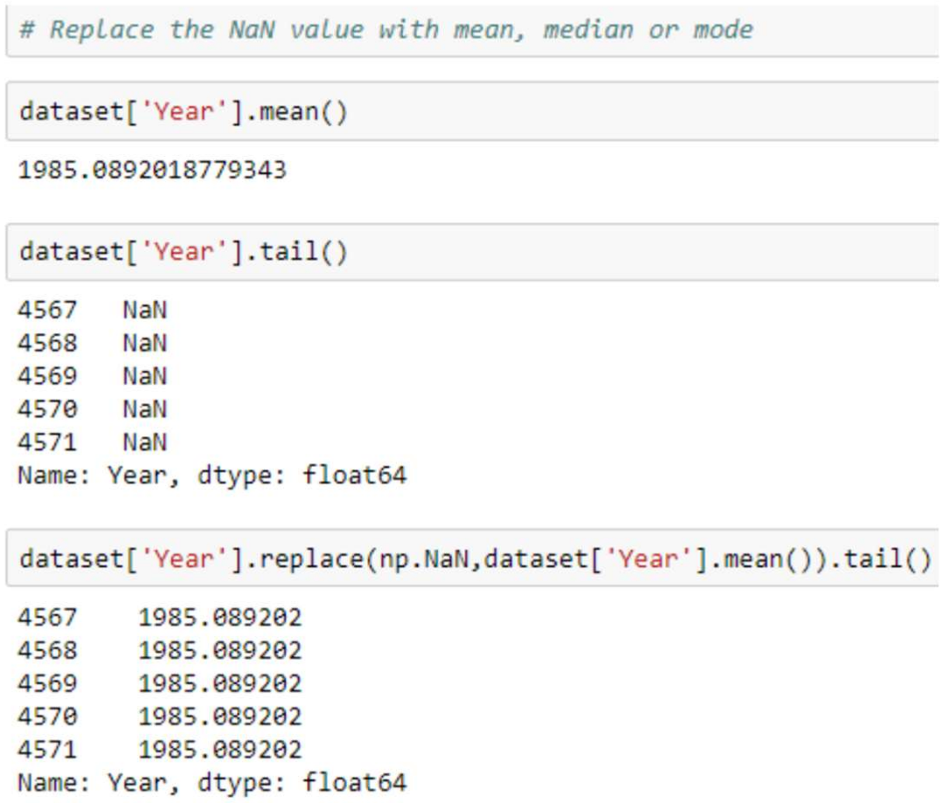


```
dataset.shape
```

```
(850, 20)
```

Step 3. Missing Values (cont'd)

- Calculate mean and use it to replace the missing values



```
# Replace the NaN value with mean, median or mode
```

```
dataset['Year'].mean()
```

```
1985.0892018779343
```

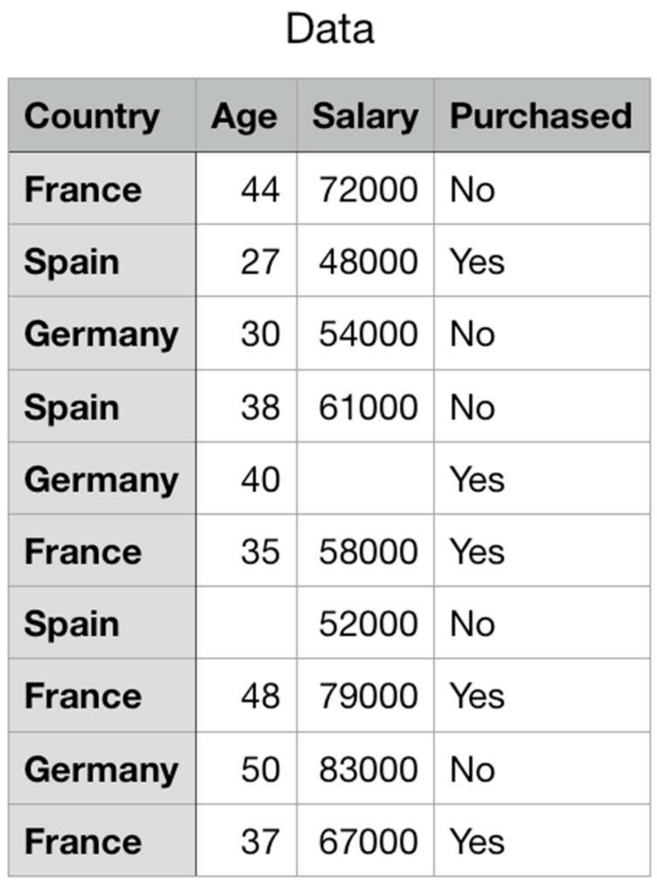
```
dataset['Year'].tail()
```

```
4567    NaN
4568    NaN
4569    NaN
4570    NaN
4571    NaN
Name: Year, dtype: float64
```

```
dataset['Year'].replace(np.NaN,dataset['Year'].mean()).tail()
```

```
4567     1985.089202
4568     1985.089202
4569     1985.089202
4570     1985.089202
4571     1985.089202
Name: Year, dtype: float64
```

Step 4. Categorical Value

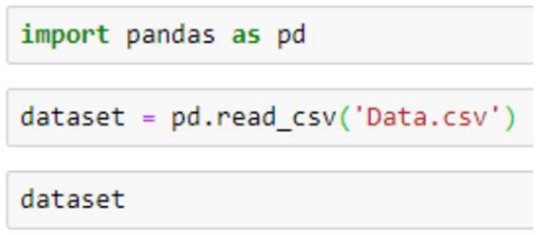


Data

Country	Age	Salary	Purchased
France	44	72000	No
Spain	27	48000	Yes
Germany	30	54000	No
Spain	38	61000	No
Germany	40		Yes
France	35	58000	Yes
Spain		52000	No
France	48	79000	Yes
Germany	50	83000	No
France	37	67000	Yes

Step 4. Categorical Value (cont'd)

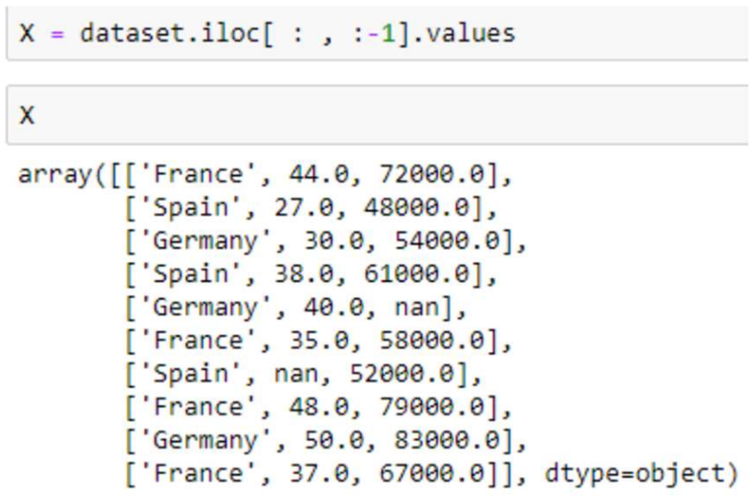
- Read csv file



```
import pandas as pd
```

```
dataset = pd.read_csv('Data.csv')
```

```
dataset
```



```
X = dataset.iloc[ : , :-1].values
```

```
X
```

```
array([['France', 44.0, 72000.0],
       ['Spain', 27.0, 48000.0],
       ['Germany', 30.0, 54000.0],
       ['Spain', 38.0, 61000.0],
       ['Germany', 40.0, nan],
       ['France', 35.0, 58000.0],
       ['Spain', nan, 52000.0],
       ['France', 48.0, 79000.0],
       ['Germany', 50.0, 83000.0],
       ['France', 37.0, 67000.0]], dtype=object)
```

-

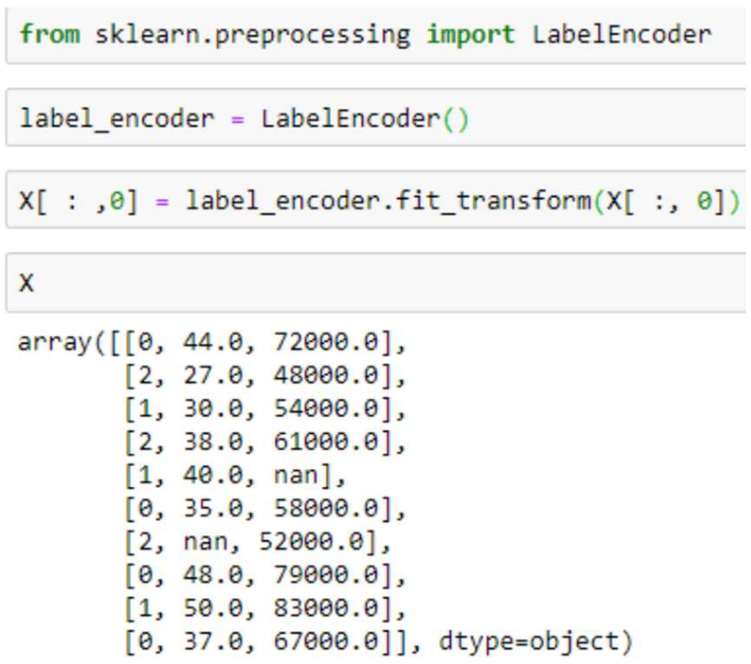
```
imputer = Imputer(missing_values = "NaN", strategy = "mean", axis =
0)
```

```
imputer = imputer.fit(X[:,1:3])
```

```
X[:, 1:3] = imputer.transform(X[:, 1:3])
```

Step 4. Categorical Value (cont'd)

- Label encoding



```
from sklearn.preprocessing import LabelEncoder
```

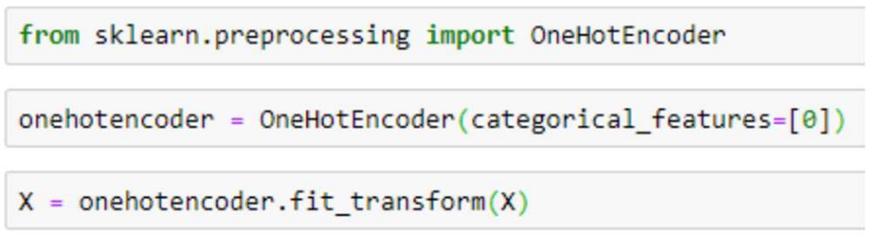
```
label_encoder = LabelEncoder()
```

```
X[ : ,0] = label_encoder.fit_transform(X[ :, 0])
```

```
X
```

```
array([[0, 44.0, 72000.0],
       [2, 27.0, 48000.0],
       [1, 30.0, 54000.0],
       [2, 38.0, 61000.0],
       [1, 40.0, nan],
       [0, 35.0, 58000.0],
       [2, nan, 52000.0],
       [0, 48.0, 79000.0],
       [1, 50.0, 83000.0],
       [0, 37.0, 67000.0]], dtype=object)
```

- OneHotencoder



```
from sklearn.preprocessing import OneHotEncoder
```

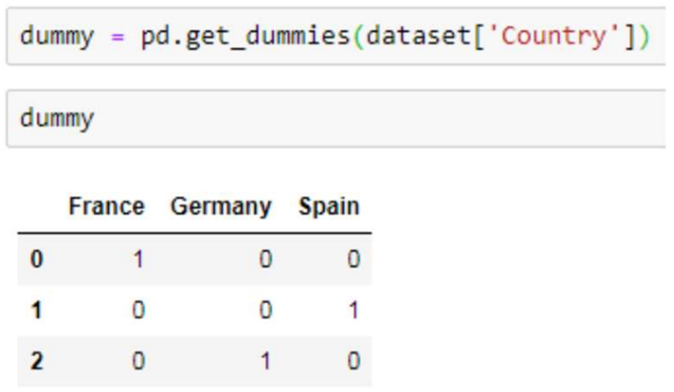
```
onehotencoder = OneHotEncoder(categorical_features=[0])
```

```
X = onehotencoder.fit_transform(X)
```

Step 4. Categorical Value (cont'd)

- Dummy variables
  : 0 or 1 to indicate the absence or presence of some categorical effect

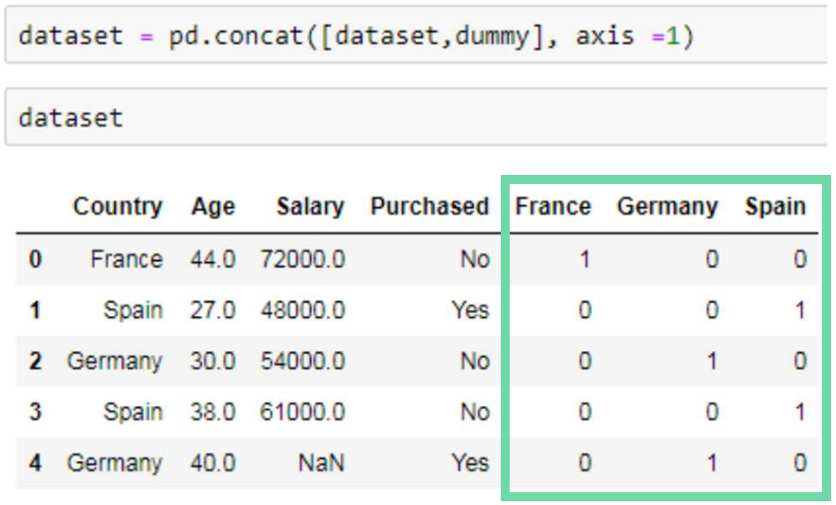
- Number of Columns = Number of Categories



```
dummy = pd.get_dummies(dataset['Country'])
```

```
dummy
```

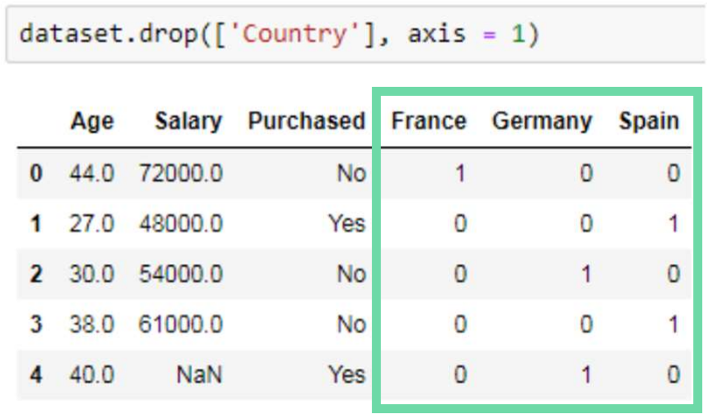
	France	Germany	Spain
0	1	0	0
1	0	0	1
2	0	1	0



```
dataset = pd.concat([dataset,dummy], axis =1)
```

```
dataset
```

	Country	Age	Salary	Purchased	France	Germany	Spain
0	France	44.0	72000.0	No	1	0	0
1	Spain	27.0	48000.0	Yes	0	0	1
2	Germany	30.0	54000.0	No	0	1	0
3	Spain	38.0	61000.0	No	0	0	1
4	Germany	40.0	NaN	Yes	0	1	0

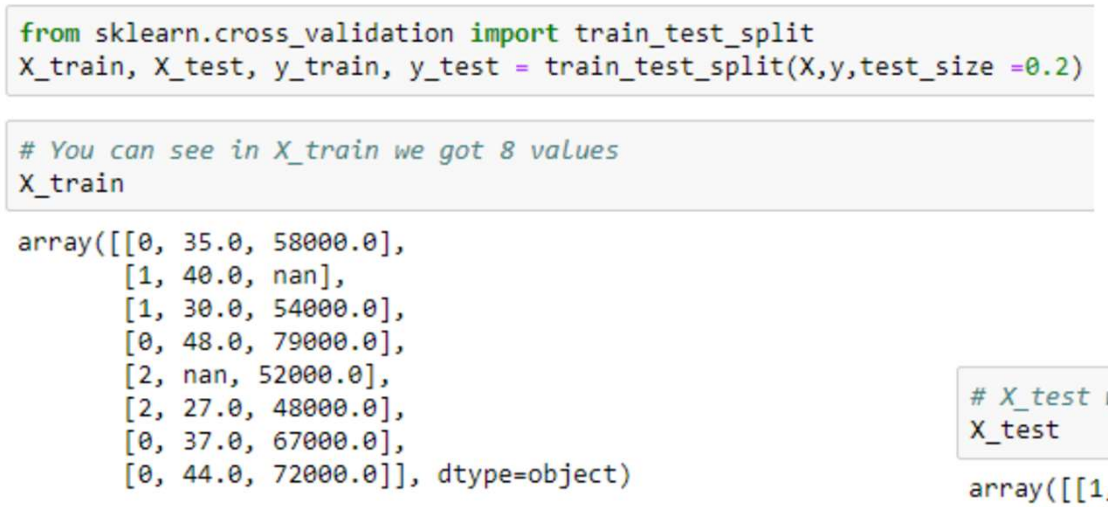


```
dataset.drop(['Country'], axis = 1)
```

	Age	Salary	Purchased	France	Germany	Spain
0	44.0	72000.0	No	1	0	0
1	27.0	48000.0	Yes	0	0	1
2	30.0	54000.0	No	0	1	0
3	38.0	61000.0	No	0	0	1
4	40.0	NaN	Yes	0	1	0

Step 5. Training & Test Set

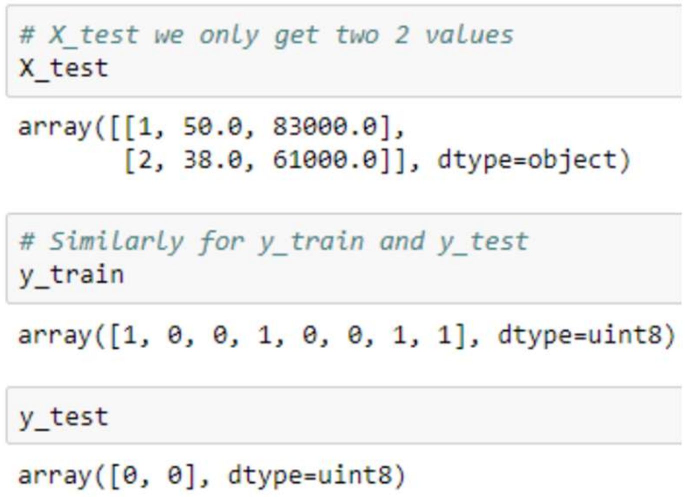
- split the dataset into training and test set



```
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size =0.2)
```

```
# You can see in X_train we got 8 values
X_train
```

```
array([[0, 35.0, 58000.0],
       [1, 40.0, nan],
       [1, 30.0, 54000.0],
       [0, 48.0, 79000.0],
       [2, nan, 52000.0],
       [2, 27.0, 48000.0],
       [0, 37.0, 67000.0],
       [0, 44.0, 72000.0]], dtype=object)
```



```
# X_test we only get two 2 values
X_test
```

```
array([[1, 50.0, 83000.0],
       [2, 38.0, 61000.0]], dtype=object)
```

```
# Similarly for y_train and y_test
y_train
```

```
array([1, 0, 0, 1, 0, 0, 1, 1], dtype=uint8)
```

```
y_test
```

```
array([0, 0], dtype=uint8)
```

Step 6. Feature Scaling

- feature scaling
  : the method to limit the range of variables so that they can be compared on common grounds

- age & salary : not same scale
  → Euclidean distance

$$d(A,B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

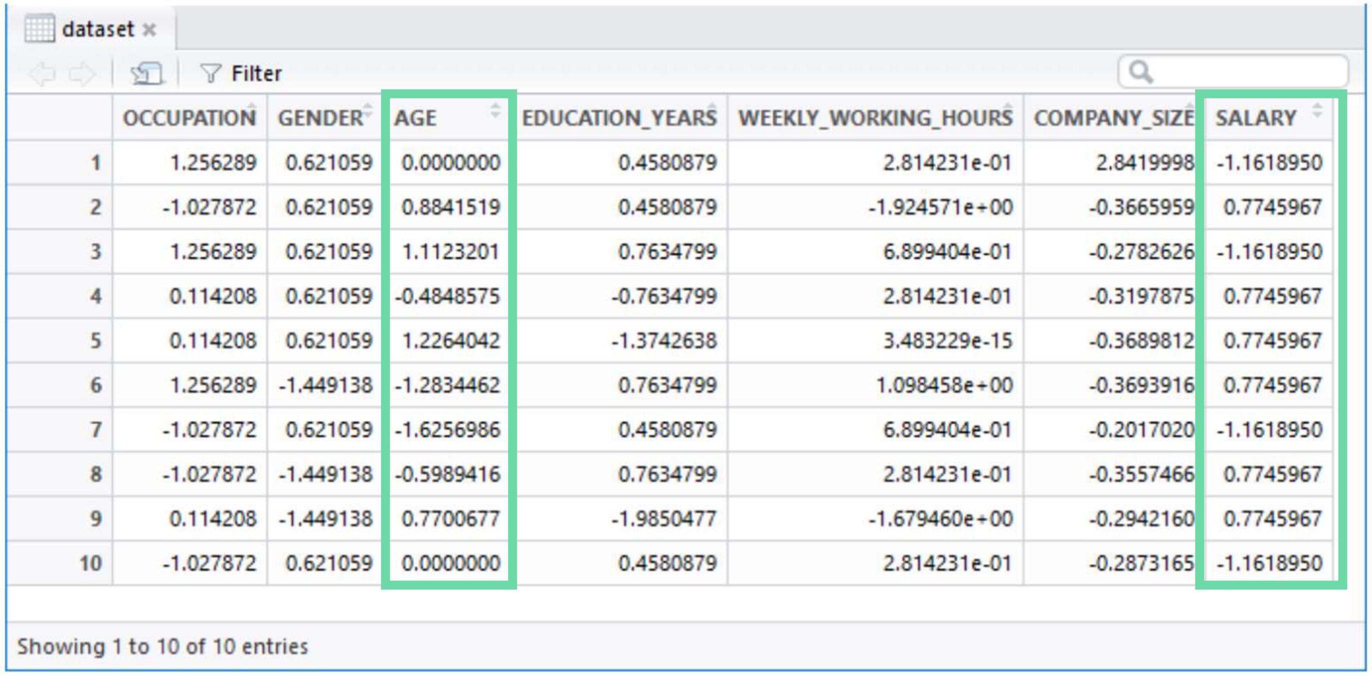
- Other methods
  1. Rescaling (min-max normalization) $\rightarrow$ $x' = \frac{x - \min(x)}{\max(x) - \min(x)}$
  2. Mean normalization $\rightarrow$ $x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)}$
  3. Standardization $\rightarrow$ $x' = \frac{x - \bar{x}}{\sigma}$

Step 6. Feature Scaling (cont'd)

- feature scaling (standard scaler)

```
from sklearn.preprocessing import StandardScaler
standard_X = StandardScaler()
```

```
X_train = standard_X.fit_transform(X_train)
X_test = standard_X.fit_transform(X_test)
```



dataset

Filter

	OCCUPATION	GENDER	AGE	EDUCATION_YEARS	WEEKLY_WORKING_HOURS	COMPANY_SIZE	SALARY
1	1.256289	0.621059	0.0000000	0.4580879	2.814231e-01	2.8419998	-1.1618950
2	-1.027872	0.621059	0.8841519	0.4580879	-1.924571e+00	-0.3665959	0.7745967
3	1.256289	0.621059	1.1123201	0.7634799	6.899404e-01	-0.2782626	-1.1618950
4	0.114208	0.621059	-0.4848575	-0.7634799	2.814231e-01	-0.3197875	0.7745967
5	0.114208	0.621059	1.2264042	-1.3742638	3.483229e-15	-0.3689812	0.7745967
6	1.256289	-1.449138	-1.2834462	0.7634799	1.098458e+00	-0.3693916	0.7745967
7	-1.027872	0.621059	-1.6256986	0.4580879	6.899404e-01	-0.2017020	-1.1618950
8	-1.027872	-1.449138	-0.5989416	0.7634799	2.814231e-01	-0.3557466	0.7745967
9	0.114208	-1.449138	0.7700677	-1.9850477	-1.679460e+00	-0.2942160	0.7745967
10	-1.027872	0.621059	0.0000000	0.4580879	2.814231e-01	-0.2873165	-1.1618950

Showing 1 to 10 of 10 entries

After Feature Scaling all values comes into same scale

Dataset

"worldcupmatches.csv"

https://medium.com/datadriveninvestor/data-preprocessing-for-machine-learning-188e9eef1d2c

Dataset

“data.csv”

https://hackernoon.com/what-steps-should-one-take-while-doing-data-preprocessing-502c993e1caa

Columns description of WorldCupMatches.csv

Integer
Year: The year in which the match was played

Date
Datetime: The Date on which the match was played along with a 24 hour format time

Numeric
Stage: The stage at which the match was played

String
Stadium: Stadium name where the match was held

String
City: The city name, where the match was played

Country
Home Team Name: Home team country name

Numeric
Home Team Goals: Total goals scored by the home team by the end of the match

Numeric
Away Team Goals: Total goals scored by the away team by the end of the match

String
Away Team Name: Away team country name

String
Win conditions: Special win condition (if any)

Numeric
Attendance: Total crowd present at the stadium

Numeric
Half-time Home Goals: Goals scored by the home team until half time

Numeric
Half-time Away Goals: Goals scored by the away team until half time

String
Referee: Name of the first referee

String
Assistant 1: Name of the first assistant referee (linesman)

String
Assistant 2: Name of the second assistant referee (linesman)

Numeric
RoundID: Unique ID of the Round

Numeric
MatchID: Unique ID of the match

String
Home Team Initials: Home team country's three letter initials

String
Away Team Initials: Away team country's three letter initials

Columns description of WorldCupPlayer.csv

RoundID: Unique ID of the round

Numeric
MatchID: Unique ID of the match

String
Team Initials: Player's team initials

String
Coach Name: Name and country of the team coach

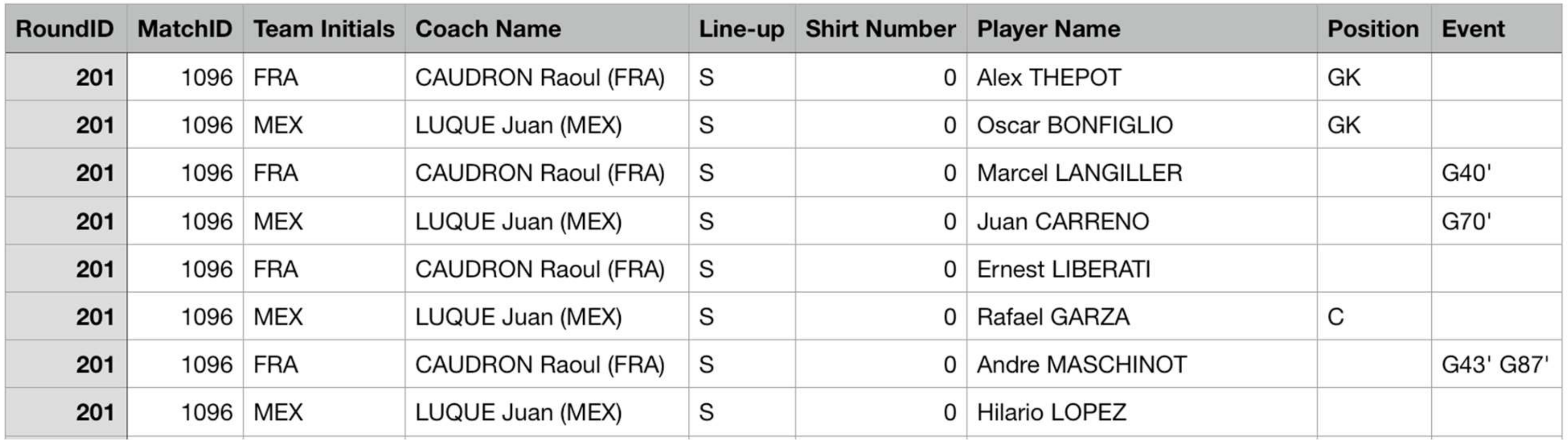
String
Line-up: S=Line-up, N=Substitute

Numeric
Shirt Number: Shirt number if available

String
Player Name: Name of the player

String
Position: C=Captain, GK=Goalkeeper

String
Event: G=Goal, OG=Own Goal, Y=Yellow Card, R=Red Card, SY = Red Card by second yellow, P=Penalty, MP=Missed Penalty, I = Substitution In, O=Substitute Out



RoundID	MatchID	Team Initials	Coach Name	Line-up	Shirt Number	Player Name	Position	Event
201	1096	FRA	CAUDRON Raoul (FRA)	S	0	Alex THEPOT	GK	
201	1096	MEX	LUQUE Juan (MEX)	S	0	Oscar BONFIGLIO	GK	
201	1096	FRA	CAUDRON Raoul (FRA)	S	0	Marcel LANGILLER		G40'
201	1096	MEX	LUQUE Juan (MEX)	S	0	Juan CARRENO		G70'
201	1096	FRA	CAUDRON Raoul (FRA)	S	0	Ernest LIBERATI		
201	1096	MEX	LUQUE Juan (MEX)	S	0	Rafael GARZA	C	
201	1096	FRA	CAUDRON Raoul (FRA)	S	0	Andre MASCHINOT		G43' G87'
201	1096	MEX	LUQUE Juan (MEX)	S	0	Hilario LOPEZ		

Columns description of WorldCup.csv

Numeric
Year: Year of the worldcup

String
Country: Country of the worldcup

String
Winner: Team who won the worldcup

String
Runners-Up: Team who was the second place

String
Third: Team who was the third place

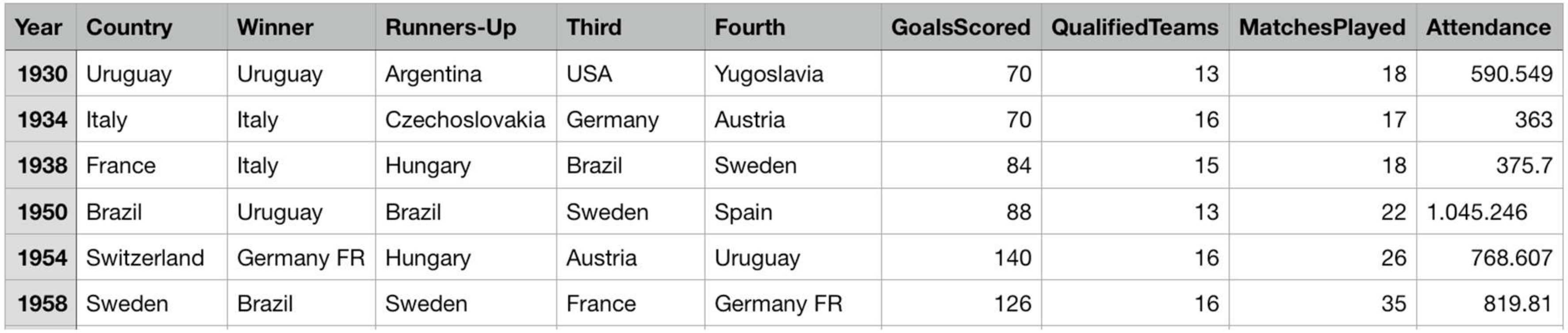
String
Fourth: Team who was the fourth place

Numeric
GoalsScored: Total goals scored in the worldcup

Numeric
QualifiedTeams: Total participating teams

Numeric
MatchesPlayed: Total matches played in the cup

Numeric
Attendance: Total attendance of the worldcup



Year	Country	Winner	Runners-Up	Third	Fourth	GoalsScored	QualifiedTeams	MatchesPlayed	Attendance
1930	Uruguay	Uruguay	Argentina	USA	Yugoslavia	70	13	18	590.549
1934	Italy	Italy	Czechoslovakia	Germany	Austria	70	16	17	363
1938	France	Italy	Hungary	Brazil	Sweden	84	15	18	375.7
1950	Brazil	Uruguay	Brazil	Sweden	Spain	88	13	22	1.045.246
1954	Switzerland	Germany FR	Hungary	Austria	Uruguay	140	16	26	768.607
1958	Sweden	Brazil	Sweden	France	Germany FR	126	16	35	819.81

“column description of worldcup data”

https://www.kaggle.com/abecklas/fifa-world-cup#WorldCups.csv “world cup 1930 - 2014 data analysis”

https://www.kaggle.com/pavanraj159/fifa-world-cup-1930-to-2014-data-analysis “world cup 2018 prediction”

https://www.kaggle.com/angps95/fifa-world-cup-2018-prediction

Reference

- The Best Public Datasets for Data Science
  - https://towardsdatascience.com/the-50-best-public-datasets-for-machine-learning-d80e9f030279

End of Term Project Guidelines