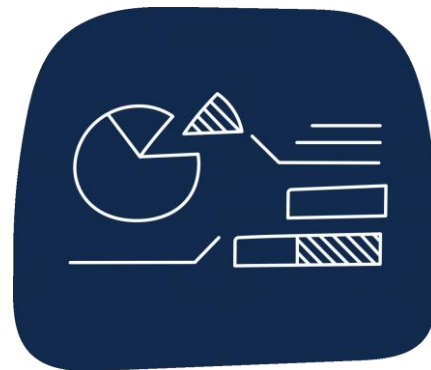
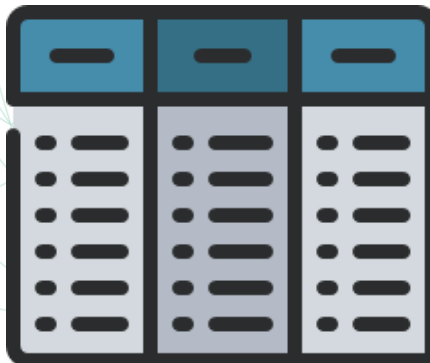


Python for Data Science



WHAT IS PANDAS?



Why and where should we use Pandas?



WHERE?

If you are facing any project that involves data where you have to visualize, analyze or do any kind of operations with data



WHY?

“fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive.”



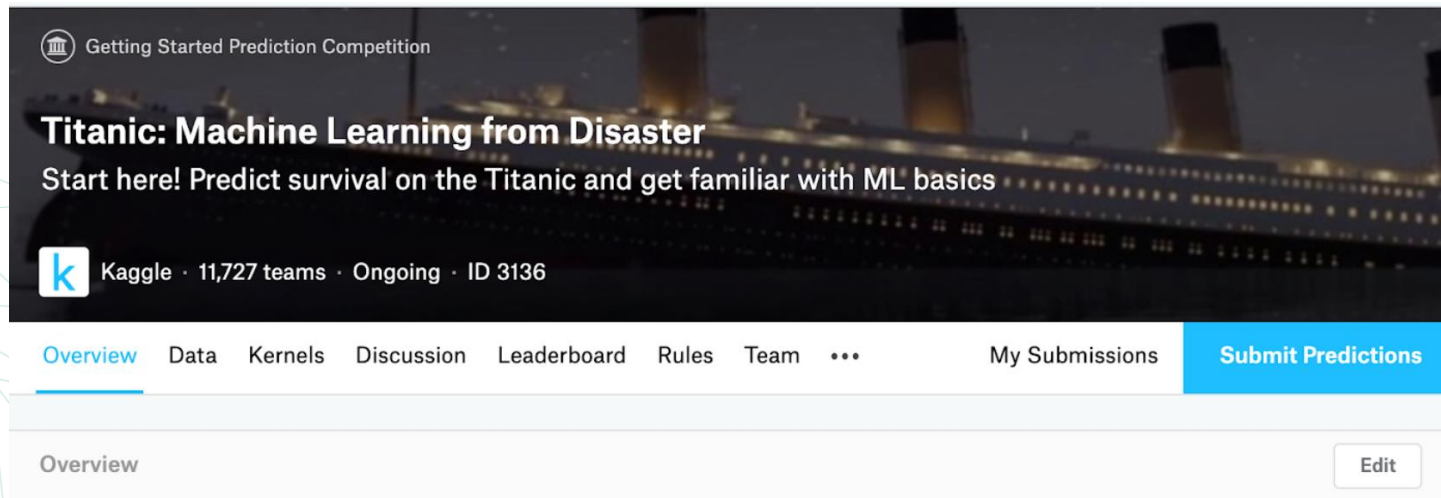
WHY?

implemented in C, giving it an extra speed boost

Let's see what it can do




THE DATA: TITANIC DATASET



Getting Started Prediction Competition

Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics

 Kaggle · 11,727 teams · Ongoing · ID 3136

[Overview](#) [Data](#) [Kernels](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [...](#) [My Submissions](#) [Submit Predictions](#)

Overview [Edit](#)

First Step

Reading the data and first insights

DataFrame
Table with Columns and Rows.
Series



First Step

READING THE DATA AND FIRST INSIGHTS

```
import pandas as pd
```

```
df = pd.read_csv("train.csv")
```

```
df.head()
```

```
# notice how no n is given, to get the first 10 items it would be:
```

```
# df.head(10)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

First Step READING THE DATA AND FIRST INSIGHTS

df.info()

Titanic Dataframe

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 891 entries, 0 to 890  
Data columns (total 12 columns):  
PassengerId    891 non-null int64  
Survived       891 non-null int64  
Pclass         891 non-null int64  
Name           891 non-null object  
Sex            891 non-null object  
Age            714 non-null float64  
SibSp          891 non-null int64  
Parch          891 non-null int64  
Ticket         891 non-null object  
Fare           891 non-null float64  
Cabin          204 non-null object  
Embarked       889 non-null object  
dtypes: float64(2), int64(5), object(5)  
memory usage: 83.6+ KB
```

df2.info()

Other Dataframe

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 10 entries, 0 to 9  
Data columns (total 4 columns):  
Name           10 non-null object  
Surname        10 non-null object  
Height         10 non-null int64  
Weight         10 non-null int64  
dtypes: int64(2), object(2)  
memory usage: 400.0+ bytes
```


First Step READING THE DATA AND FIRST INSIGHTS

`df.describe()`

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

First Step READING THE DATA AND FIRST INSIGHTS

```
include_list = ['object', 'float64', 'int64']
df.describe(include = include_list)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000000	891	891.000000	204	889
unique	NaN	NaN	NaN	891	2	NaN	NaN	NaN	681	NaN	147	3
top	NaN	NaN	NaN	Moubarek, Master. Gerios	male	NaN	NaN	NaN	1601	NaN	B96 B98	S
freq	NaN	NaN	NaN	1	577	NaN	NaN	NaN	7	NaN	4	644
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381594	NaN	32.204208	NaN	NaN
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806057	NaN	49.693429	NaN	NaN
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000000	NaN	0.000000	NaN	NaN
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000000	NaN	7.910400	NaN	NaN
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000000	NaN	14.454200	NaN	NaN
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000000	NaN	31.000000	NaN	NaN
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000000	NaN	512.329200	NaN	NaN

Second Step

Accessing specific records



Second Step

ACCESSING SPECIFIC RECORDS

```
df[0:10]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

Second Step

ACCESSING SPECIFIC RECORDS

```
df.loc[1]
```

PassengerId	2
Survived	1
Pclass	1
Name	Cumings, Mrs. John Bradley (Florence Briggs Th...
Sex	female
Age	38
SibSp	1
Parch	0
Ticket	PC 17599
Fare	71.2833
Cabin	C85
Embarked	C

Name: 1, dtype: object

ACCESSING SPECIFIC RECORDS

We can also use the **iloc** method for accessing data with a certain index position

```
df.iloc[1]
```

PassengerId	2
Survived	1
Pclass	1
Name	Cumings, Mrs. John Bradley (Florence Briggs Th...
Sex	female
Age	38
SibSp	1
Parch	0
Ticket	PC 17599
Fare	71.2833
Cabin	C85
Embarked	C

Name: 1, dtype: object

What's the difference then?

Name	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
Braund, Mr. Owen Harris	1	0	3	male	22.0	1	0	A/5 21171	7.2500	NaN	S
Cumings, Mrs. John Bradley (Florence Briggs Thayer)	2	1	1	female	38.0	1	0	PC 17599	71.2833	C85	C
Heikkinen, Miss. Laina	3	1	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
Futrelle, Mrs. Jacques Heath (Lily May Peel)	4	1	1	female	35.0	1	0	113803	53.1000	C123	S
Allen, Mr. William Henry	5	0	3	male	35.0	0	0	373450	8.0500	NaN	S

```
df.loc['Cumings, Mrs. John Bradley (Florence Briggs Thayer)']
```

```

PassengerId      2
Survived          1
Pclass           1
Sex              female
Age             38
SibSp            1
Parch            0
Ticket           PC 17599
Fare             71.2833
Cabin            C85
Embarked         C
Name: Cumings, Mrs. John Bradley (Florence Briggs Thayer), dtype: object

```

Second Step

ACCESSING SPECIFIC RECORDS

df['Sex']

```
df['Sex']
```

Name	
Braund, Mr. Owen Harris	male
Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female
Heikkinen, Miss. Laina	female
Futrelle, Mrs. Jacques Heath (Lily May Peel)	female
Allen, Mr. William Henry	male
Moran, Mr. James	male
McCarthy, Mr. Timothy J	male
Palsson, Master. Gosta Leonard	male
Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female
Nasser, Mrs. Nicholas (Adele Achem)	female
Sandstrom, Miss. Marguerite Rut	female

Third Step

Indexes, Columns and Drop



Third Step INDEXES, COLUMNS AND DROP

using the drop method, for which we must declare the **axis as 1** to specify that it is a column we want to remove.

```
df = df.drop(['Parch', 'SibSp'], axis = 1)
```

#Setting the indexes of our dataframe to be the names of the passengers:

```
df = df.set_index("Name")
```

Fourth Step

unique and value_counts



Fourth Step UNIQUE AND VALUE_COUNTS

`df['Cabin'].unique()`

```
array([nan, 'C85', 'C123', 'E46', 'G6', 'C103', 'D56', 'A6',
       'C23 C25 C27', 'B78', 'D33', 'B30', 'C52', 'B28', 'C83', 'F33',
       'F G73', 'E31', 'A5', 'D10 D12', 'D26', 'C110', 'B58 B60', 'E101',
       'F E69', 'D47', 'B86', 'F2', 'C2', 'E33', 'B19', 'A7', 'C49', 'F4',
       'A32', 'B4', 'B80', 'A31', 'D36', 'D15', 'C93', 'C78', 'D35',
       'C87', 'B77', 'E67', 'B94', 'C125', 'C99', 'C118', 'D7', 'A19',
       'B49', 'D', 'C22 C26', 'C106', 'C65', 'E36', 'C54',
       'B57 B59 B63 B66', 'C7', 'E34', 'C32', 'B18', 'C124', 'C91', 'E40',
       'T', 'C128', 'D37', 'B35', 'E50', 'C82', 'B96 B98', 'E10', 'E44',
       'A34', 'C104', 'C111', 'C92', 'E38', 'D21', 'E12', 'E63', 'A14',
       'B37', 'C30', 'D20', 'B79', 'E25', 'D46', 'B73', 'C95', 'B38',
       'B39', 'B22', 'C86', 'C70', 'A16', 'C101', 'C68', 'A10', 'E68',
       'B41', 'A20', 'D19', 'D50', 'D9', 'A23', 'B50', 'A26', 'D48',
       'E58', 'C126', 'B71', 'B51 B53 B55', 'D49', 'B5', 'B20', 'F G63',
       'C62 C64', 'E24', 'C90', 'C45', 'E8', 'B101', 'D45', 'C46', 'D30',
       'E121', 'D11', 'E77', 'F38', 'B3', 'D6', 'B82 B84', 'D17', 'A36',
       'B102', 'B69', 'E49', 'C47', 'D28', 'E17', 'A24', 'C50', 'B42',
       'C148'], dtype=object)
```

Fourth Step

UNIQUE AND VALUE_COUNTS

Seeing how many passengers are male and female

```
df['Sex'].value_counts()
```

```
male          577  
female        314  
Name: Sex, dtype: int64
```

Let's Code!



Quiz Time!

Any Questions?



Thanks