

TO START

```
# IMPORT DATA LIBRARIES
import pandas as pd
import numpy as np

# IMPORT VIS LIBRARIES
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

# IMPORT MODELLING LIBRARIES
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
```

PRELIMINARY OPERATIONS

<code>df=pd.read_csv('data.csv')</code>	read data
<code>df.head()</code>	check head df
<code>df.info()</code>	check info df
<code>df.describe()</code>	check stats df
<code>df.columns</code>	check col names

VISUALISE DATA

<code>sns.heatmap(df.isnull())*</code>	check null values
<code>sns.set_style('whitegrid')</code>	set different style
<code>sns.countplot('col',df)</code>	countplot
<code>sns.countplot('col',df,palette=)</code>	countplot
<code>sns.countplot('col',df,hue=,palette=)</code>	countplot
<code>sns.distplot(df['col'].dropna(),bins=30)</code>	distribution plot

sns.heatmap(): can take more useful parameters;
`yticklabels=False, cbar=False, cmap='viridis'`

DATA CLEANING

create a personalised function*	impute values
apply the personalised function*	apply function
<code>dummy_var = pd.get_dummies(df['col'],drop_first=True)*</code>	convert categorical features
<code>df.drop(['old.col1',...])</code>	drop old columns
<code>df= pd.concat([dummy_var],axis=1)</code>	add the new dummy var into the df

See imputing and apply section.

drop.first=True: without this command, we would have two specular columns, leading to issues of multicollinearity.

IMPUTING AND APPLY

```
# EXAMPLE OF A POSSIBLE FUNCTION TO IMPUTE MISSING VALUES
def impute_age(cols):
    Age = cols[0]
    Pclass = cols[1]

    if pd.isnull(Age):
        if Pclass == 1:
            return 37
        elif Pclass == 2:
            return 29
        else:
            return 24
    else:
        return Age

# EXAMPLE OF HOW TO APPLY THE FUNCTION
train['Age'] = train[['Age', 'Pclass']].apply(impute_age,axis=1)
```

You can impute using mean, median, etc. If you are interested in using Bayesian Estimation, you can see here:
<https://github.com/jeweinberg/Pandas-MICE> or
<https://pypi.python.org/pypi/fancyimpute>



By **DarioPittera** (aggialavura)

Not published yet.

Last updated 26th June, 2019.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

TRAIN and EVALUATE MODEL

□ CREATE X and y

`X = df[['col1','col2',etc.]]` create df features

`y = df['col']` create df var to predict

▢ SPLIT DATASET

`X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)` split df in train and test df

..|| FIT THE MODEL

`log = LogisticRegression()` instatiate model

`log.fit(X_train,y_train)` train/fit the model

◎ MAKE PREDICTIONS

`predictions = log.predict(X_test)` make predictions

✓ EVAUATE MODEL

`print(classification_report(y_test,predictions))` useful measures

`confusion_matrix(y_test, predictions)`



By **DarioPittera** (aggialavura)

Not published yet.

Last updated 26th June, 2019.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>