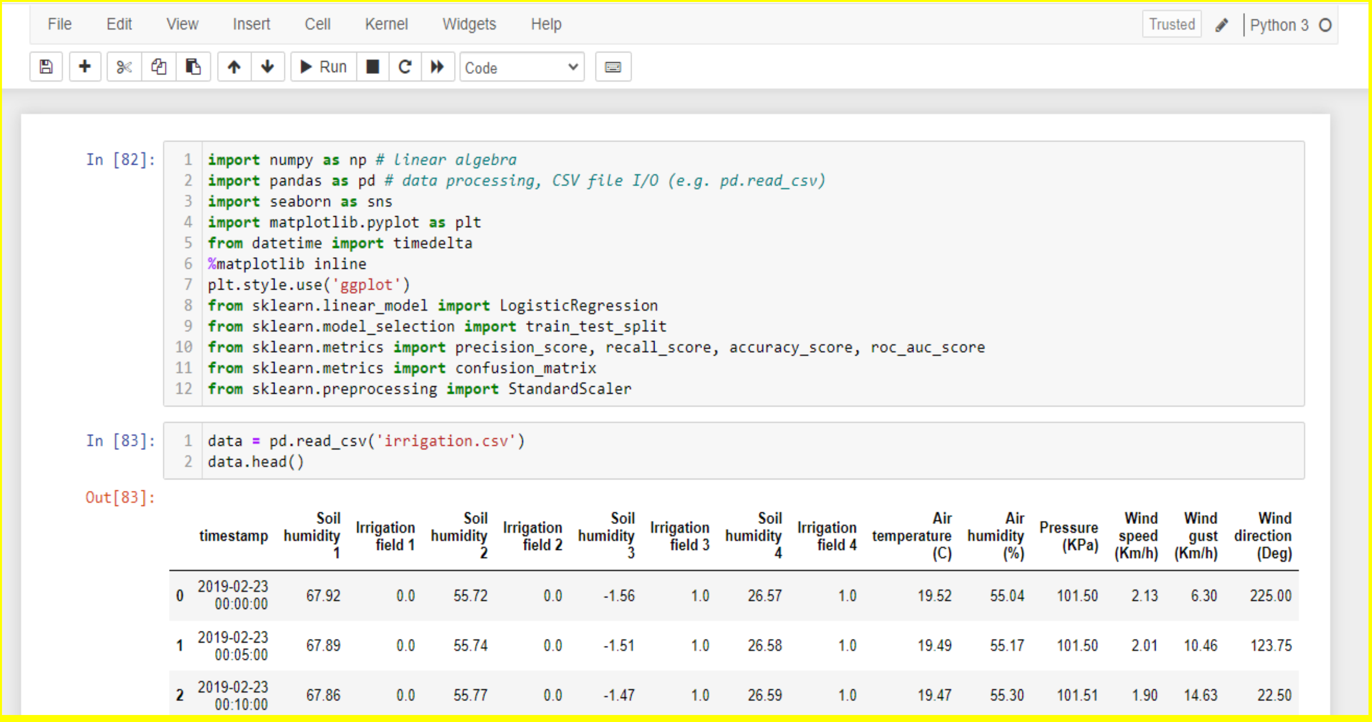
- **First of all, I have read my dataset:**



In [82]:

```python
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt
from datetime import timedelta
%matplotlib inline
plt.style.use('ggplot')
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_score, recall_score, accuracy_score, roc_auc_score
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler
```

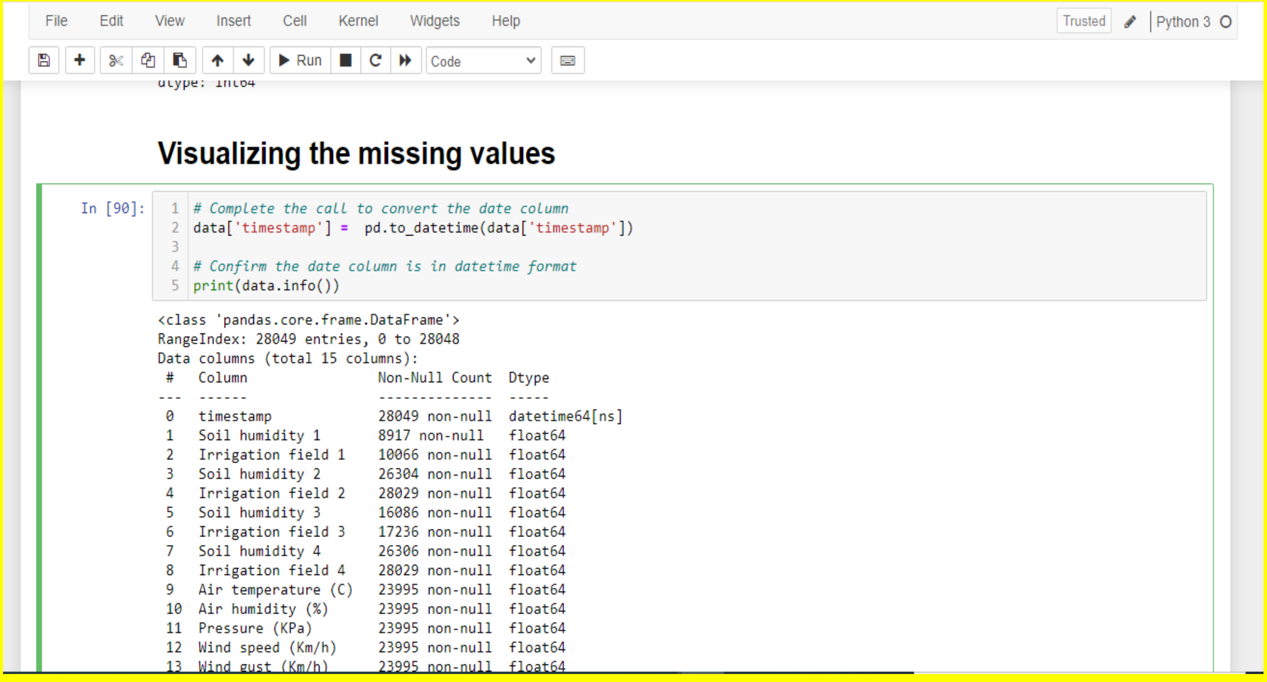
In [83]:

```python
data = pd.read_csv('irrigation.csv')
data.head()
```

Out[83]:

	timestamp	Soil humidity 1	Irrigation field 1	Soil humidity 2	Irrigation field 2	Soil humidity 3	Irrigation field 3	Soil humidity 4	Irrigation field 4	Air temperature (C)	Air humidity (%)	Pressure (KPa)	Wind speed (Km/h)	Wind gust (Km/h)	Wind direction (Deg)
0	2019-02-23 00:00:00	67.92	0.0	55.72	0.0	-1.56	1.0	26.57	1.0	19.52	55.04	101.50	2.13	6.30	225.00
1	2019-02-23 00:05:00	67.89	0.0	55.74	0.0	-1.51	1.0	26.58	1.0	19.49	55.17	101.50	2.01	10.46	123.75
2	2019-02-23 00:10:00	67.86	0.0	55.77	0.0	-1.47	1.0	26.59	1.0	19.47	55.30	101.51	1.90	14.63	22.50

- **Cleaned dataset:**



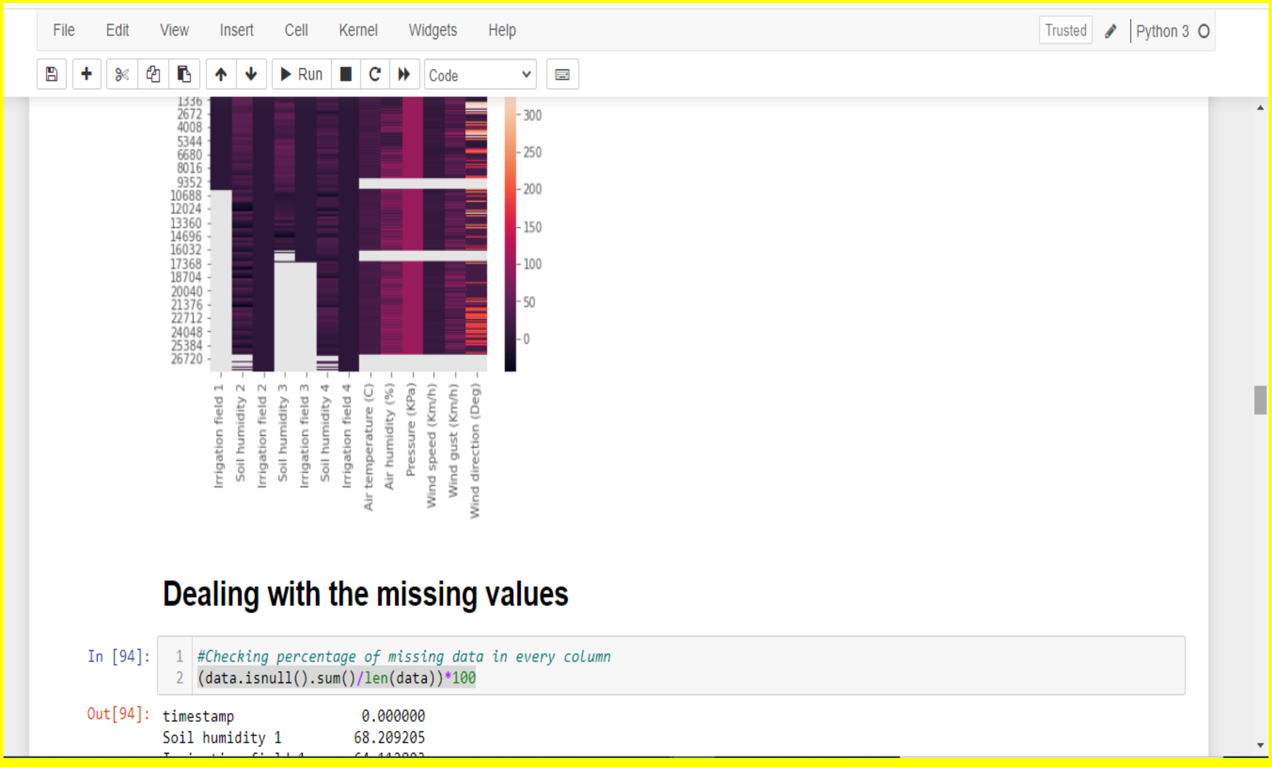
```
dtype: int64
```

Visualizing the missing values

In [90]:

```python
# Complete the call to convert the date column
data['timestamp'] =  pd.to_datetime(data['timestamp'])

# Confirm the date column is in datetime format
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28049 entries, 0 to 28048
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   timestamp            28049 non-null  datetime64[ns]
 1   Soil humidity 1      8917 non-null   float64
 2   Irrigation field 1   10066 non-null  float64
 3   Soil humidity 2      26304 non-null  float64
 4   Irrigation field 2   28029 non-null  float64
 5   Soil humidity 3      16086 non-null  float64
 6   Irrigation field 3   17236 non-null  float64
 7   Soil humidity 4      26306 non-null  float64
 8   Irrigation field 4   28029 non-null  float64
 9   Air temperature (C)  23995 non-null  float64
 10  Air humidity (%)     23995 non-null  float64
 11  Pressure (KPa)       23995 non-null  float64
 12  Wind speed (Km/h)    23995 non-null  float64
 13  Wind gust (Km/h)     23995 non-null  float64
```

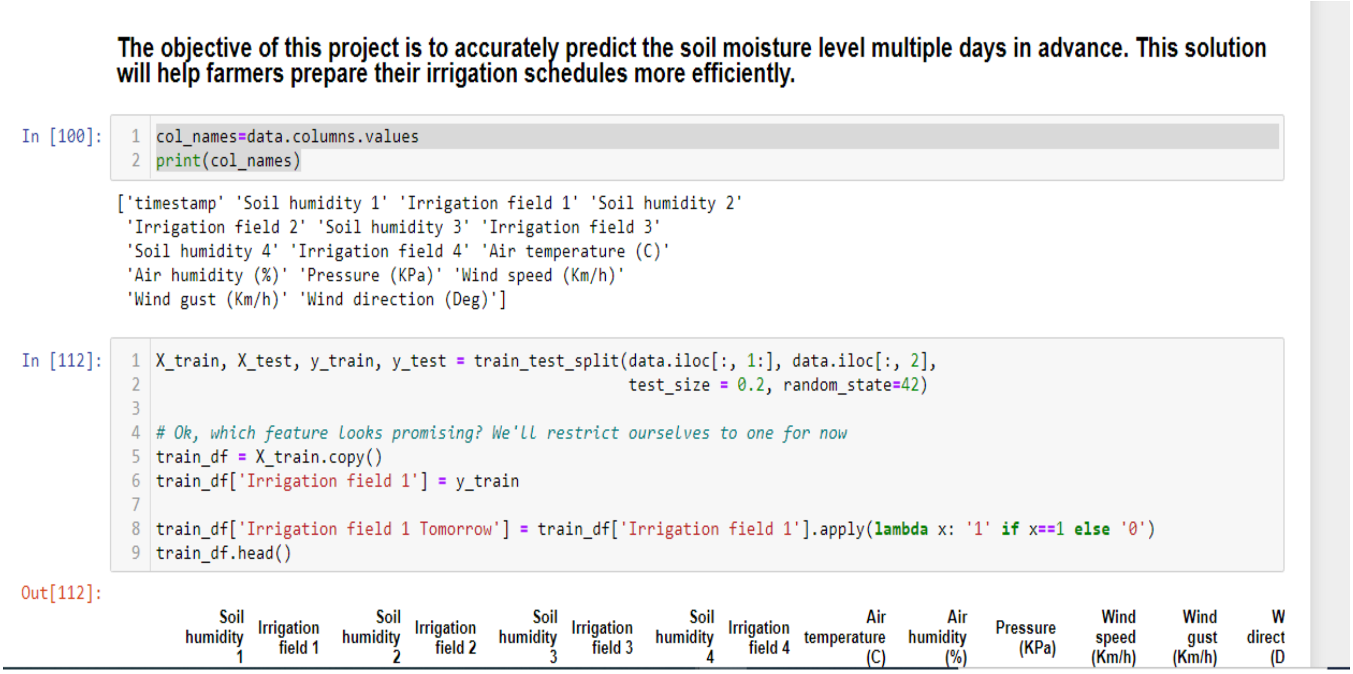


Dealing with the missing values

In [94]:
```
#Checking percentage of missing data in every column
(data.isnull().sum()/len(data))*100
```

Out[94]:
```
timestamp              0.000000
Soil humidity 1       68.209205
```

- **Then i have train my data :**



The objective of this project is to accurately predict the soil moisture level multiple days in advance. This solution will help farmers prepare their irrigation schedules more efficiently.

In [100]:
```
col_names=data.columns.values
print(col_names)
```

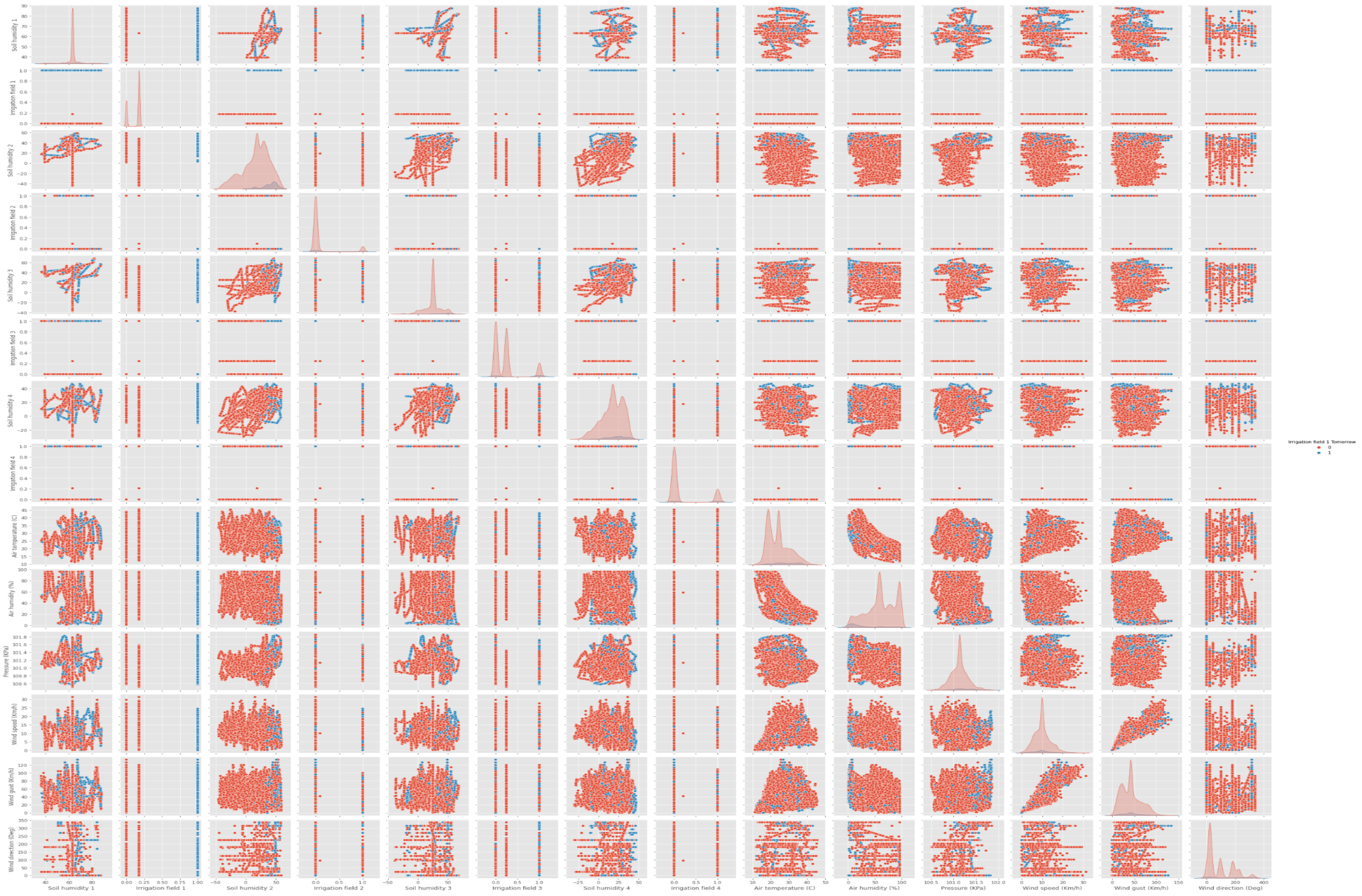
```
['timestamp' 'Soil humidity 1' 'Irrigation field 1' 'Soil humidity 2'
 'Irrigation field 2' 'Soil humidity 3' 'Irrigation field 3'
 'Soil humidity 4' 'Irrigation field 4' 'Air temperature (C)'
 'Air humidity (%)' 'Pressure (KPa)' 'Wind speed (Km/h)'
 'Wind gust (Km/h)' 'Wind direction (Deg)']
```

In [112]:
```
X_train, X_test, y_train, y_test = train_test_split(data.iloc[:, 1:], data.iloc[:, 2],
                                                     test_size = 0.2, random_state=42)

# Ok, which feature looks promising? We'll restrict ourselves to one for now
train_df = X_train.copy()
train_df['Irrigation field 1'] = y_train

train_df['Irrigation field 1 Tomorrow'] = train_df['Irrigation field 1'].apply(lambda x: '1' if x==1 else '0')
train_df.head()
```

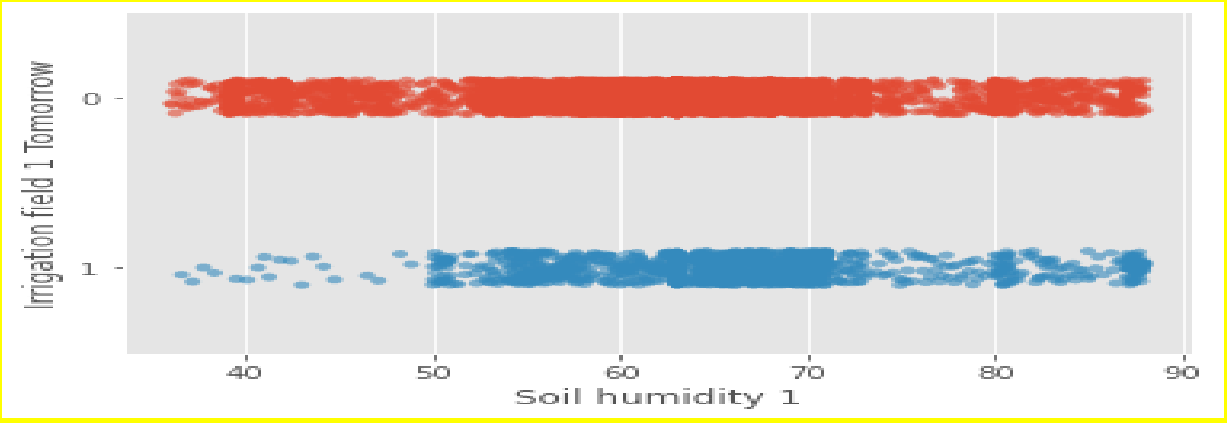
Out[112]:

Soil humidity 1	Irrigation field 1	Soil humidity 2	Irrigation field 2	Soil humidity 3	Irrigation field 3	Soil humidity 4	Irrigation field 4	Air temperature (C)	Air humidity (%)	Pressure (KPa)	Wind speed (Km/h)	Wind gust (Km/h)	W direct (D

- Apply pairplot to pick one feature to start that separates the two cases. :



- I find the good features is (Soil Humidity 1 , with the Irrigation field 1 Tomorrow “prediction” is the best one , after that i will apply logistic regression Model Random Forest ..etc :


Irrigation field 1 Tomorrow
Soil humidity 1