Prepare data for machine learning

ANALYZING IOT DATA IN PYTHON

Matthias Voppichler
IT Developer

Machine Learning Refresher

- Supervised learning
  - Classification
  - Regression
- Unsupervised learning
  - Cluster analysis
- Deep learning
  - Neural networks

Machine Learning Refresher

- Supervised learning
  - **Classification**
  - Regression
- Unsupervised learning
  - Cluster analysis
- Deep learning
  - Neural networks

Labels

```
print(environment_labeled.head())
```

```
                     humidity  temperature  pressure   label
timestamp
2018-10-01 00:00:00      81.0         11.8    1013.4       1
2018-10-01 00:15:00      79.7         11.9    1013.1       1
2018-10-01 00:30:00      81.0         12.1    1013.0       1
2018-10-01 00:45:00      79.7         11.7    1012.7       1
2018-10-01 01:00:00      84.3         11.2    1012.6       1
```

Train / Test split

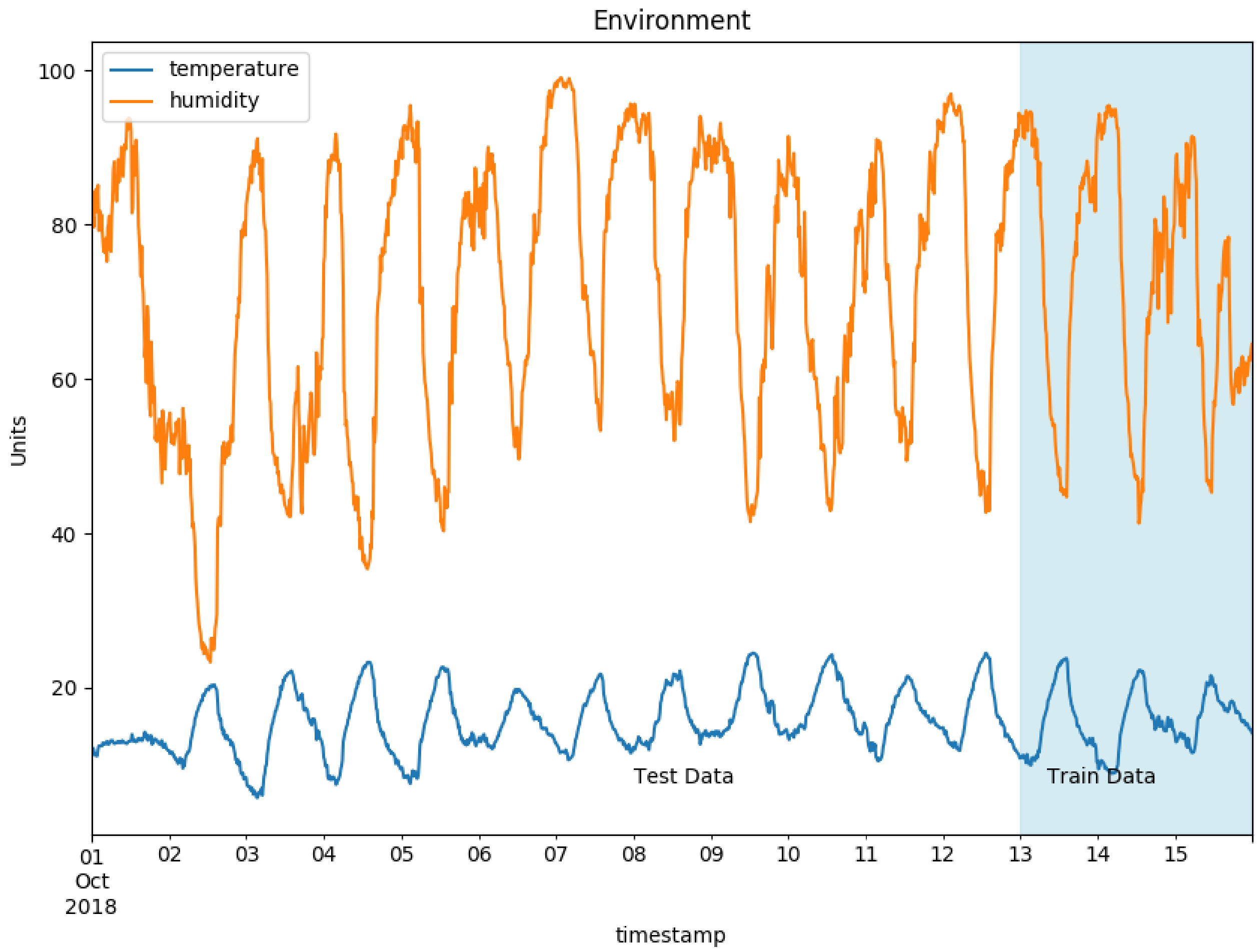
Splitting time series data

- Model should not see test-data during training
- Cannot use random split
- Model should not be allowed to look into the future

Train / test split

```
split_day = "2018-10-13"
train = environment[:split_day]
test = environment[split_day:]
print(train.iloc[0].name)
print(train.iloc[-1].name)
print(test.iloc[0].name)
print(test.iloc[-1].name)
```

```
2018-10-01 00:00:00
2018-10-13 23:45:00
2018-10-13 00:00:00
2018-10-15 23:45:00
```



Features and Labels

```
X_train = train.drop("target", axis=1)
y_train = train["target"]
X_test = test.drop("target", axis=1)
y_test = test["target"]
print(X_train.shape)
print(y_train.shape)
```

```
(1248, 3)
(1248,)
```

Logistic Regression

```python
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
print(logreg.predict(X_test))
```

```
[0 0 1 1 1 1 1 0 0]
```

Let's practice!

ANALYZING IOT DATA IN PYTHON

Scaling data for machine learning

ANALYZING IOT DATA IN PYTHON

Matthias Voppichler
IT Developer

Evaluate the model

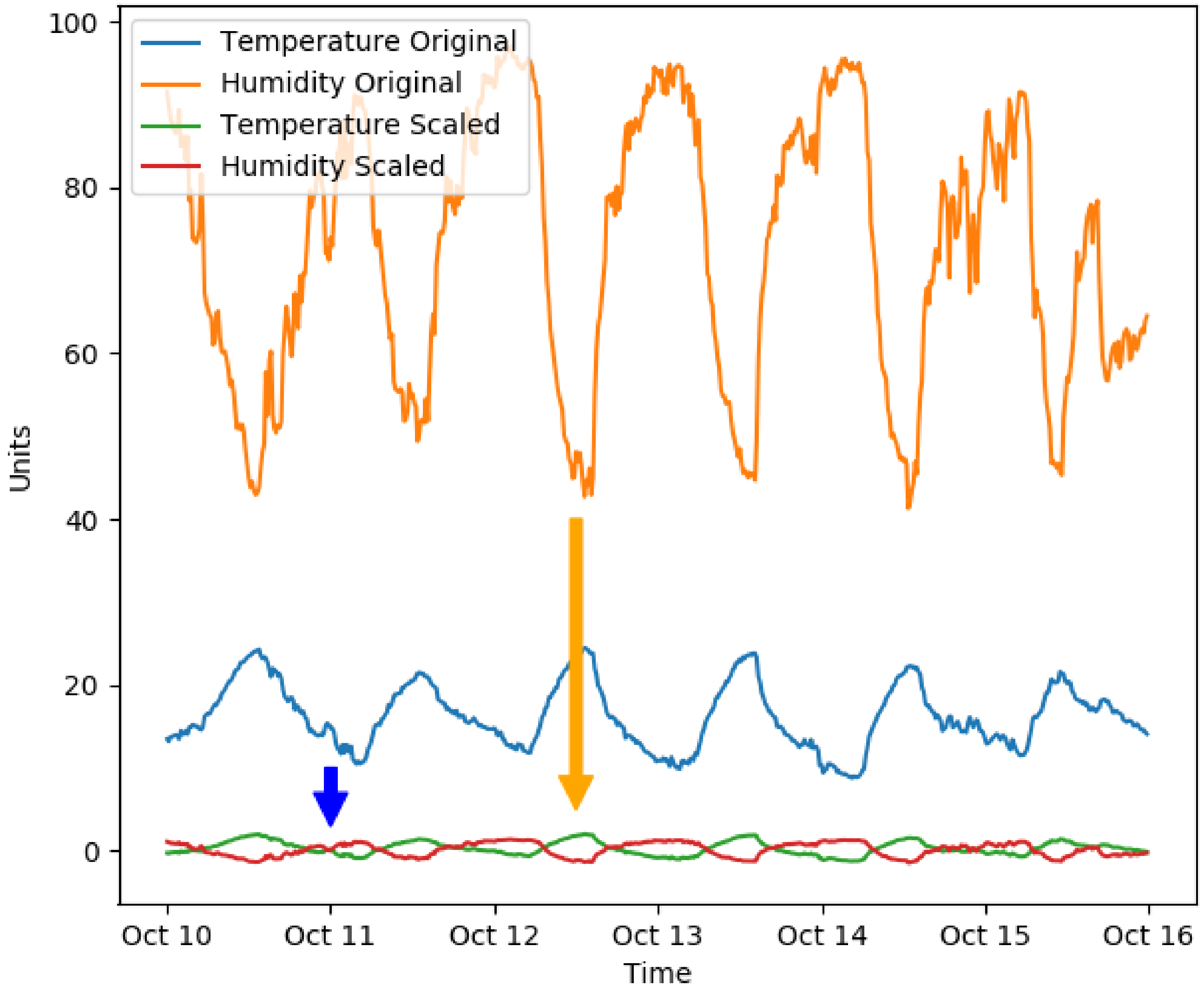
```python
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
print(logreg.score(X_test, y_test))
```

```
0.78145113
```

Scaling

scikit-learn's `StandardScaler`

- remove mean
- scale data to variance



Unscaled data

```
print(data.head())
```

```
                     humidity  temperature  pressure
timestamp
2018-10-01 00:00:00      81.0         11.8    1013.4
2018-10-01 00:15:00      79.7         11.9    1013.1
2018-10-01 00:30:00      81.0         12.1    1013.0
2018-10-01 00:45:00      79.7         11.7    1012.7
2018-10-01 01:00:00      84.3         11.2    1012.6
```

Standardscaler

```python
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(data)
print(sc.mean_)
print(sc.var_)
```

```
[  71.8826716    14.17002019 1018.17042396]
[372.78261022  20.37926608  53.67519188]
```

```python
data_scaled = sc.transform(data)
```

Standardscaler

```
df_scaled = pd.DataFrame(data_scaled,
                         columns=data.columns,
                         index=data.index)
print(data_scaled.head())
```

```
                     humidity  temperature  pressure
timestamp
2018-10-01 00:00:00  0.472215    -0.524998 -0.651134
2018-10-01 00:15:00  0.404884    -0.502847 -0.692082
2018-10-01 00:30:00  0.472215    -0.458543 -0.705731
2018-10-01 00:45:00  0.404884    -0.547150 -0.746679
2018-10-01 01:00:00  0.643132    -0.657908 -0.760329
```

Evaluate the model

```
logreg = LogisticRegression()
logreg.fit(X_train_scaled, y_train_scaled)

print(logreg.score(X_test_scaled, y_test_scaled))
```

```
0.88145113
```

Let's practice!

ANALYZING IOT DATA IN PYTHON

Develop machine learning pipeline

ANALYZING IOT DATA IN PYTHON

Matthias Voppichler
IT Developer

Pipeline

- Transform
  - Conversation
  - Scaling
- Estimator
  - Model

Create a Pipeline

```python
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
# Initialize Objects
sc = StandardScaler()
logreg = LogisticRegression()
# Create pipeline
pl = Pipeline([
        ("scale", sc),
        ("logreg", logreg)
    ])
```

Inspect Pipeline

```
pl
```

```
Pipeline(memory=None,
         steps=[('scale', StandardScaler(copy=True, with_mean=True, with_std=True)),
                ('logreg', <class 'sklearn.linear_model.logistic.LogisticRegression'>)])
```

```
pl.fit(X_train, y_train)
print(pl.predict(X_test))
```

```
[0 0 1 1 0 1 1 0 0]
```

Save model

```
import pickle
with Path("pipeline_model.pkl").open("bw") as f:
    pickle.dump(pl, f)
```

Load Model

```python
import pickle
with Path("pipeline_model.pkl").open('br') as f:
    pl = pickle.load(f)

pl
```

```
Pipeline(memory=None,
     steps=[('scale', StandardScaler(copy=True, with_mean=True, with_std=True)),
             ('logreg', LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn', n_jobs=None, penalty='l2',
          random_state=None, solver='warn', tol=0.0001, verbose=0, warm_start=False))])
```

A word of caution

DO NOT unpickle untrusted files, this can lead to malicious code being executed.

Let's practice!

ANALYZING IOT DATA IN PYTHON

Apply a machine learning model

ANALYZING IOT DATA IN PYTHON

Matthias Voppichler
IT Developer

Model Recap

```
# Create Pipeline
pl = Pipeline([
        ("scale", StandardScaler()),
        ("logreg", LogisticRegression())
    ])
# Fit the pipeline
pl.fit(X_train, y_train)
print(pl.score(X_test, y_test))
```

```
0.8897932222860425
```

Predict

```python
predictions = pl.predict(X_test)
print(predictions)
print(f"Test length: {len(X_test)}")
print(f"Prediction length: {len(predictions)}")
```

```
[0 0 0 ... 1 1 1]
Test length: 500
Prediction length: 500
```

Record conversation

```
print(single_record)
```

```
{'timestamp': '2018-11-30 18:15:00',
  'humidity': 81.7,
  'pressure': 1019.8,
  'temperature': 1.5},
```

```
cols = X_train.columns
df = pd.DataFrame.from_records([single_record],
                               index="timestamp",
                               columns=cols)
```

Apply to datastream

```python
def on_message(client, userdata, message):
    data = json.loads(message.payload)
    df = pd.DataFrame.from_records([data],
                                   index="timestamp",
                                   columns=cols)
    category = pl.predict(df)
    maybe_alert(category[0])
subscribe.callback(on_message, topic, hostname=MQTT_HOST)
```

Let's practice!

ANALYZING IOT DATA IN PYTHON

Wrapping up

ANALYZING IOT DATA IN PYTHON

Matthias Voppichler
IT Developer

What you have learned

- Accessing IoT data
  - from a REST API
  - from a datastream
- Data Cleaning
- Correlations
- Time series decomposition
- Machine learning pipeline

Next steps

- Machine Learning
- Database
- Big data
- PySpark

Congratulations!

ANALYZING IOT DATA IN PYTHON