



PREPROCESSING FOR MACHINE LEARNING IN PYTHON

Standardizing Data

Sarah Guido

Senior Data Scientist



What is standardization?

- Scikit-learn models assume normally distributed data
- Log normalization and feature scaling in this course
- Applied to continuous numerical data



When to standardize: models

- Model in linear space
- Dataset features have high variance
- Dataset features are continuous and on different scales
- Linearity assumptions



PREPROCESSING FOR MACHINE LEARNING IN PYTHON

Let's practice!



PREPROCESSING FOR MACHINE LEARNING IN PYTHON

Log normalization

Sarah Guido

Senior Data Scientist



What is log normalization?

- Applies log transformation
- Natural log using the constant e (2.718)
- Captures relative changes, the magnitude of change, and keeps everything in the positive space

Number	Log
30	3.4
300	5.7
3000	8

Log normalization in Python

```
In [1]: print(df)
```

	col1	col2
0	1.00	3.0
1	1.20	45.5
2	0.75	28.0
3	1.60	100.0

```
In [2]: print(df.var())
```

col1	0.128958
col2	1691.729167
dtype:	float64

```
In [3]: import numpy as np
```

```
In [4]: df["col2_log"] =  
        np.log(df["col2"])
```

```
In [5]: print(df)
```

	col1	col2	col2_log
0	1.00	3.0	1.098612
1	1.20	45.5	3.817712
2	0.75	28.0	3.332205
3	1.60	100.0	4.605170

```
In [6]: print(np.var(df[["col1",  
                        "col2_log"]]))
```

col1	0.096719
col2_log	1.697165
dtype:	float64



PREPROCESSING FOR MACHINE LEARNING IN PYTHON

Let's practice!



PREPROCESSING FOR MACHINE LEARNING IN PYTHON

Scaling data

Sarah Guido

Senior Data Scientist



What is feature scaling?

- Features on different scales
- Model with linear characteristics
- Center features around 0 and transform to unit variance
- Transforms to approximately normal distribution



How to scale data

```
In [1]: print(df)
```

	col1	col2	col3
0	1.00	48.0	100.0
1	1.20	45.5	101.3
2	0.75	46.2	103.5
3	1.60	50.0	104.0

```
In [2]: print(df.var())
```

```
col1    0.128958  
col2    4.055833  
col3    3.526667  
dtype: float64
```

How to scale data

```
In [3]: from sklearn.preprocessing import StandardScaler
```

```
In [4]: scaler = StandardScaler()
```

```
In [5]: df_scaled = pd.DataFrame(scaler.fit_transform(df),  
                                columns=df.columns)
```

```
In [6]: print(df_scaled)
```

	col1	col2	col3
0	-0.442127	0.329683	-1.352726
1	0.200967	-1.103723	-0.553388
2	-1.245995	-0.702369	0.799338
3	1.487156	1.476409	1.106776

```
In [7]: print(df_scaled.var())
```

col1	1.333333
col2	1.333333
col3	1.333333

dtype: float64



PREPROCESSING FOR MACHINE LEARNING IN PYTHON

Let's practice!



PREPROCESSING FOR MACHINE LEARNING IN PYTHON

Standardized data and modeling

Sarah Guido

Senior Data Scientist



K-nearest neighbors

```
In [1]: from sklearn.model_selection import train_test_split
In [2]: from sklearn.neighbors import KNeighborsClassifier

# Preprocessing first

In [3]: X_train, X_test, y_train, y_test = train_test_split(X, y)

In [4]: knn = KNeighborsClassifier()
In [5]: knn.fit(X_train, y_train)

In [6]: knn.score(X_test, y_test)
```



PREPROCESSING FOR MACHINE LEARNING IN PYTHON

Let's practice!