linear_algorithm_notebook

November 8, 2022

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
from sklearn.metrics import r2_score
```

[2]:
```
df= pd.read_excel('Linear Regression Algorithm/test-data.xlsx')
df=df.iloc[0:,:6]
x=(df.head(39))
y=[2,2,2,2,1]
print(df)
```

```
    SoTA  GA  Saves  Save%  CS  PSxG
0      3   0      3  100.0   1   0.3
1      7   4      3   42.9   0   3.4
2      3   1      3  100.0   0   1.1
3      8   1      7   87.5   0   1.3
4      3   2      2   66.7   0   1.1
..   …    ..    …     …    ..   …
73     5   0      5  100.0   1   1.1
74     4   1      3   75.0   0   1.5
75     3   1      2   66.7   0   1.4
76     3   1      2   66.7   0   0.6
77     2   1      1   50.0   0   0.8

[78 rows x 6 columns]
```

[3]:
```
columns = df.columns.tolist()
target = "GA"
# Generate the training set.  Set random_state to be able to replicate results.
train = df.loc[6:38]
```

```python
# Select anything not in the training set and put it in the testing set.
test = df.loc[2:5]
# Print the shapes of both sets.
print("Training set shape:", train.shape)
print("Testing set shape:", test.shape)
```

```
Training set shape: (33, 6)
Testing set shape: (4, 6)
```

[4]:

```python
lin_model = LinearRegression()
# Fit the model to the training data.
lin_model.fit(train[columns], train[target])
# Generate our predictions for the test set.
lin_predictions = lin_model.predict(test[columns])
print("Predictions:", lin_predictions)
# Compute error between our test predictions and the actual values.
lin_mse = mean_squared_error(lin_predictions, test[target])
print("Computed error:", lin_mse)
```

```
Predictions: [1. 1. 2. 2.]
Computed error: 1.6220952363607055e-29
```

[ ]:

```python

```