Sprachverarbeitung: Übung

SoSe 24

Janis Pagel
Department for Digital Humanities, University of Cologne

2024-06-18

For this exercise, you need to both submit manual calculations as well as Python code. Please submit two files in Ilias, one a PDF with your calculations and one a file containing your Python code (either Jupyter Notebook or Python script). You can also combine both files into a zip-archive and submit only the archive. You can either solve the calculations by hand on a sheet of paper, scan it and submit as a PDF file or use the capabilities to write mathematical equations of tools like MS Word / LibreOffice / LaTeX, etc. to write down your calculations digitally.

Exercise 1.

For this part of the exercise, you need to submit manual calculations as a PDF file.
Given are the data points:

Text length	Number of literary characters
10	3
105	5
150	8
210	12
250	7
295	13

Given are two different linear functions that could potentially fit the data:

$$f1(x) = 0.45x + 1 \tag{1}$$

$$f2(x) = 0.68x - 5 \tag{2}$$

Calculate the mean squared error for both hypotheses given the data. Which function fits the data better?

Solution 1.

$$
\begin{aligned}
f1(10) &= 0.45 \times 10 + 1 = 5.5 \\
f1(105) &= 0.45 \times 105 + 1 = 48.25 \\
f1(150) &= 0.45 \times 150 + 1 = 68.5 \\
f1(210) &= 0.45 \times 210 + 1 = 95.5 \\
f1(250) &= 0.45 \times 250 + 1 = 113.5 \\
f1(295) &= 0.45 \times 295 + 1 = 133.75
\end{aligned}
$$

$$
J(f1) = \frac{1}{6} \times ((5.5-3)^2 + (48.25-5)^2 + (68.5-8)^2 + \\
(95.5-12)^2 + (113.5-7)^2 + (133.75-13)^2) \approx 6405.35
$$

$$
\begin{aligned}
f2(10) &= 0.68 \times 10 - 5 = 1.8 \\
f2(105) &= 0.68 \times 105 - 5 = 66.4 \\
f2(150) &= 0.68 \times 150 - 5 = 97 \\
f2(210) &= 0.68 \times 210 - 5 = 137.8 \\
f2(250) &= 0.68 \times 250 - 5 = 165 \\
f2(295) &= 0.68 \times 295 - 5 = 195.6
\end{aligned}
$$

$$
J(f2) = \frac{1}{6} \times ((1.8-3)^2 + (66.4-5)^2 + (97-8)^2 + \\
(137.8-12)^2 + (165-7)^2 + (195.6-13)^2) \approx 14304.13
$$

The MSE of function $f2$ is much higher than for function $f1$, this means that $f1$ fits the data better than $f2$. However, the MSE of $f1$ is also so high that it is likely that a better function can be found.

Exercise 2.

For this part of the exercise, you need to submit Python code.
Given is a dataset on cooking recipes, which includes, among others, the duration a recipe takes to prepare (“minutes” column), a description of the necessary steps (“steps” column) and the number of steps necessary (“n_steps” column). Load the data (`https://lehre.idh.uni-koeln.de/site/assets/files/5151/recipe.tsv`) with Pandas and create a column that contains the lengths of each step description (column name “steps”)[1]. Call this new column “steps_len”. Split the data into train and test data using the train_test_split function of sklearn (`https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html`), with the train data at 60% of the whole dataset and the test data at 40%. Write Python code that trains a linear regression model (`https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression`) on the training data and tests it on the test data with the features:

[1] see the Pandas function `https://pandas.pydata.org/docs/reference/api/pandas.Series.str.len.html`

1. “step_len” “minutes”
2. “step_len” and “n_steps”

For evaluating the models performance, use Mean Squared Error (`https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html`). Which combination performs better and why? Use seaborn’s regplot function (`https://seaborn.pydata.org/generated/seaborn.regplot.html`) to plot both feature combinations. Does the distribution of the data and the predicted linear function match with the evaluation of the models?

Next, extract all rows from the dataset which have the number of steps in the “n_steps” column set to 5 and to 10 and save it into a new dataframe. Convert the occurrences of 5s in “n_steps” column to 0 and the occurrences of 10s to 1. Use seaborn’s regplot function to plot the length of steps vs. the 0s and 1s of the new “n_steps” column. In the regplot function, set `logistic=True`. What do you observe? Does the predicted logistic function match the sampled data?

Solution 2.

```
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import pandas as pd
import seaborn as sns

df = pd.read_csv("recipes.tsv", sep="\t")
df["steps_len"] = df.steps.str.len()
train_df, test_df = train_test_split(df, train_size=0.6, random_state=42)

clf = LinearRegression()
clf.fit(train_df[["steps_len"]], train_df["minutes"])
preds = clf.predict(test_df[["steps_len"]])
print(mean_squared_error(test_df["minutes"], preds))

clf = LinearRegression()
clf.fit(train_df[["steps_len"]], train_df["n_steps"])
preds = clf.predict(test_df[["steps_len"]])
print(mean_squared_error(test_df["n_steps"], preds))

sns.regplot(x = "steps_len", y = "minutes", data = df)

sns.regplot(x = "steps_len", y = "n_steps", data = df)

train_df_5_10 = train_df[(train_df["n_steps"]==5) | (train_df["n_steps"]==10)
    ]
train_df_5_10["n_steps"] = train_df_5_10["n_steps"]==10
train_df_5_10["n_steps"].astype(int)
sns.regplot(x = "steps_len", y = "n_steps", data = train_df_5_10, logistic =
    True)
```