exp4

March 13, 2022

[ ]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

[ ]:
```
df  = pd.read_csv("survey.csv",index_col=0, na_values=["NaN",])
df.dropna(axis=0,inplace=True)
df.head()
```

[ ]:

Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	no_employees	remote_work	…	leave	mental_health_consequence	phys_health_consequence	coworkers	supervisor
2014-08-27 11:36:48	33	male	United States	CA	No	Yes	Yes	Rarely	26-100	No	…	Don't know	No	No	Yes	Yes
2014-08-27 11:37:08	35	male	United States	TN	No	Yes	Yes	Sometimes	More than 1000	No	…	Very easy	Yes	No	Some of them	Yes
2014-08-27 11:39:36	42	Male	United States	WA	No	Yes	Yes	Sometimes	26-100	Yes	…	Very easy	Maybe	No	Some of them	Some of them
2014-08-27 11:43:36	38	Female	United States	TX	No	Yes	Yes	Sometimes	26-100	No	…	Somewhat easy	No	No	Some of them	Yes
2014-08-27 11:44:43	30	male	United States	IL	No	Yes	Yes	Rarely	26-100	No	…	Don't know	Maybe	No	Some of them	Some of them

```
2014-08-27 11:43:36                    No  Some of them          Yes
2014-08-27 11:44:43                    No  Some of them          Yes

                    mental_health_interview phys_health_interview  \
Timestamp
2014-08-27 11:36:48                      No                   Yes
2014-08-27 11:37:08                      No                   Yes
2014-08-27 11:39:36                   Maybe                   Yes
2014-08-27 11:43:36                      No                    No
2014-08-27 11:44:43                      No                    No

                    mental_vs_physical obs_consequence  \
Timestamp
2014-08-27 11:36:48         Don't know              No
2014-08-27 11:37:08                 No              No
2014-08-27 11:39:36         Don't know              No
2014-08-27 11:43:36                Yes              No
2014-08-27 11:44:43         Don't know              No

                                                                comments
Timestamp
2014-08-27 11:36:48                Relatively new job. Ask again later
2014-08-27 11:37:08  Sometimes I think  about using drugs for my me…
2014-08-27 11:39:36  I selected my current employer based on its po…
2014-08-27 11:43:36  Our health plan has covered my psychotherapy a…
2014-08-27 11:44:43  I just started a new job last week hence a lot…

[5 rows x 26 columns]
```

[ ]:
```
working_data = df[["Age","Gender","family_history","remote_work","treatment"]]
working_data.is_copy = False
working_data["Gender"] = working_data["Gender"].map({
    "male":0,
    "Male":0,
    "M":0,
    "female":1,
    "Female":1,
    "F":1,
})
working_data["Gender"] = working_data["Gender"]
working_data["family_history"] = working_data["family_history"].map({
    "Yes":1,
    "No":0,
})
working_data["remote_work"] = working_data["remote_work"].map({
    "Yes":1,
    "No":0,
```

```
})
working_data["treatment"] = working_data["treatment"].map({
    "Yes":1,
    "No":0,
})
working_data.dropna(axis=0,inplace=True)
working_data.reset_index()
working_data.head()
```

```
C:\Users\rinko\AppData\Local\Temp/ipykernel_3064/507328212.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  working_data["Gender"] = working_data["Gender"].map({
C:\Users\rinko\AppData\Local\Temp/ipykernel_3064/507328212.py:11:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  working_data["Gender"] = working_data["Gender"]
C:\Users\rinko\AppData\Local\Temp/ipykernel_3064/507328212.py:12:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  working_data["family_history"] = working_data["family_history"].map({
C:\Users\rinko\AppData\Local\Temp/ipykernel_3064/507328212.py:16:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  working_data["remote_work"] = working_data["remote_work"].map({
C:\Users\rinko\AppData\Local\Temp/ipykernel_3064/507328212.py:20:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
  working_data["treatment"] = working_data["treatment"].map({
C:\Users\rinko\AppData\Local\Temp/ipykernel_3064/507328212.py:24:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  working_data.dropna(axis=0,inplace=True)
```

[ ]:

Timestamp	Age	Gender	family_history	remote_work	treatment
2014-08-27 11:36:48	33	0.0	1	0	1
2014-08-27 11:37:08	35	0.0	1	0	1
2014-08-27 11:39:36	42	0.0	1	1	1
2014-08-27 11:43:36	38	1.0	1	0	1
2014-08-27 11:44:43	30	0.0	1	0	1

[ ]:
```
X = np.array(working_data[["Age","Gender","family_history","remote_work"]])
y = np.array(working_data["treatment"]).reshape(-1, 1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)

# print(X_test,y_test)
```

[ ]:
```
model = LogisticRegression()
model.fit(X_train,y_train)
```

```
C:\Users\rinko\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

[ ]:
```
LogisticRegression()
```

[ ]:
```
model.score(X_test,y_test)
```

[ ]:
```
0.7368421052631579
```