Machine Learning Practice

July 10, 2025

0.1 The Machine Learning Process is as follows:

- Loading Data
- Preprocessing Data
- Training a model
- Evaluating the model
- Predicting the future

[5]:
```
#Importing Necessary Libraries
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
```

[4]:
```
# Loading Data
url = "https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.
↪csv"
irisdata=pd.read_csv(url)
irisdata.head()
```

[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

[6]:
```
#Spliting Data into features and labels
X=irisdata.drop(columns=['species'])
y=irisdata['species']
```

[7]:
```
#splitting data into training and test sets
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.
↪2,random_state=42)
```

[8]:
```
#standardizing the features
stand=StandardScaler()
X_train_scaled=stand.fit_transform(X_train)
X_test_scaled=stand.transform(X_test)
```

[9]:
```
#creating model
model=LogisticRegression()
```

[16]:
```
#train the model
model.fit(X_train_scaled,y_train)
```

[16]:
```
LogisticRegression()
```

[15]:
```
#prediction stage
prediction=model.predict([[2,4.5,1.5,3.2]])
```

[17]:
```
#print the prediction
y_pre=model.predict(X_test_scaled)
acc=accuracy_score(y_test,y_pre)
print("Accuracy=",acc)
```

```
Accuracy= 1.0
```

[23]:
```
# Sample new data for prediction
new_data= np.array([
    [5.1, 3.5, 1.4, 0.2],
    [6.3, 2.9, 5.6, 1.8],
    [4.9, 3.0, 1.4, 0.2]
])
```

[24]:
```
#standardize new data
new=stand.transform(new_data)
```

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/sklearn/base.py:464: UserWarning: X does not have valid feature names,
but StandardScaler was fitted with feature names
  warnings.warn(
```

[25]:
```
#predict
prediction=model.predict(new)
print("Predictions",prediction)
```

```
Predictions ['setosa' 'virginica' 'setosa']
```

[ ]:
```

```