

Logistic regression TP

1/ Train a binary classifier

Use a logistic regression from scikit-learn with LogisticRegression

```
#Loading Libraries
from sklearn.linear_model import LogisticRegression
from sklearn import datasets
from sklearn.preprocessing import StandardScaler

#Loading data with 2 classes
iris = datasets.load_iris()
features = iris.data[:100,:]
target = iris.target[:100]

#Standardisation of characteristics
scaler = StandardScaler()
features_standarized = scaler.fit_transform(features)

#Create an object of logistic regression
logistic_regression = LogisticRegression(random_state=0)

#Train a model
model = logistic_regression.fit(features_standarized, target)
```

We can create a new observation and ask for prediction and probability of prediction

```
#Create an observation
new_observation = [[.5, .5, .5, .5]]

#Predict a class
model.predict(new_observation)

#Print prediction probabilities
model.predict_proba(new_observation)
```

2/ Train a multiclass classifier:

```
# Loading Libraries
from sklearn.linear_model import LogisticRegression
from sklearn import datasets
from sklearn.preprocessing import StandardScaler

# Loading data
iris = datasets.load_iris()
features = iris.data
target = iris.target

# Standardisation of characteristics
scaler = StandardScaler()
features_standardized = scaler.fit_transform(features)

# Create an object of logistic regression
logistic_regression = LogisticRegression(random_state=0, multi_class='ovr') # ovr one versus rest

# Train a model
model = logistic_regression.fit(features_standardized, target)
```

3/ Variance reduction with regularization

```
# Loading Libraries
from sklearn.linear_model import LogisticRegressionCV
from sklearn import datasets
from sklearn.preprocessing import StandardScaler

# Loading data
iris = datasets.load_iris()
features = iris.data
target = iris.target

# Standardisation of characteristics
scaler = StandardScaler()
features_standardized = scaler.fit_transform(features)

# Create an object of logistic regression
logistic_regression = LogisticRegressionCV(
    penalty='l2', Cs=10, random_state=0, n_jobs=-1)

# Train a model
model = logistic_regression.fit(features_standardized, target)
```

4/ Train a classifier on big data

```
#Loading Libraries
from sklearn.linear_model import LogisticRegression
from sklearn import datasets
from sklearn.preprocessing import StandardScaler

#Loading data
iris = datasets.load_iris()
features = iris.data
target = iris.target

#Standardisation of characteristics
scaler = StandardScaler()
features_standardized = scaler.fit_transform(features)

#Create an object of logistic regression
logistic_regression = LogisticRegression(random_state=0, solver='sag') # Stochastic descent gradient

#Train a model
model = logistic_regression.fit(features_standardized, target)
```

Apply

1/ Apply all the above examples, split the data and see classification report or accuracy or mean squared error, visualize the data.

2/ apply a logistic regression on

2.1/ Dataset: Breast Cancer Classification

We'll use the Breast Cancer dataset from scikit-learn, where:

- Features: Characteristics of cell nuclei (e.g., radius, texture, area).
- Target: 1 (Malignant) or 0 (Benign).

2.2/ Titanic dataset (Seaborn: `sns.load_dataset('titanic')`)

2.3/ Heart disease dataset (Kaggle: Heart Disease UCI)

Remark: Prepare the data in the examples.