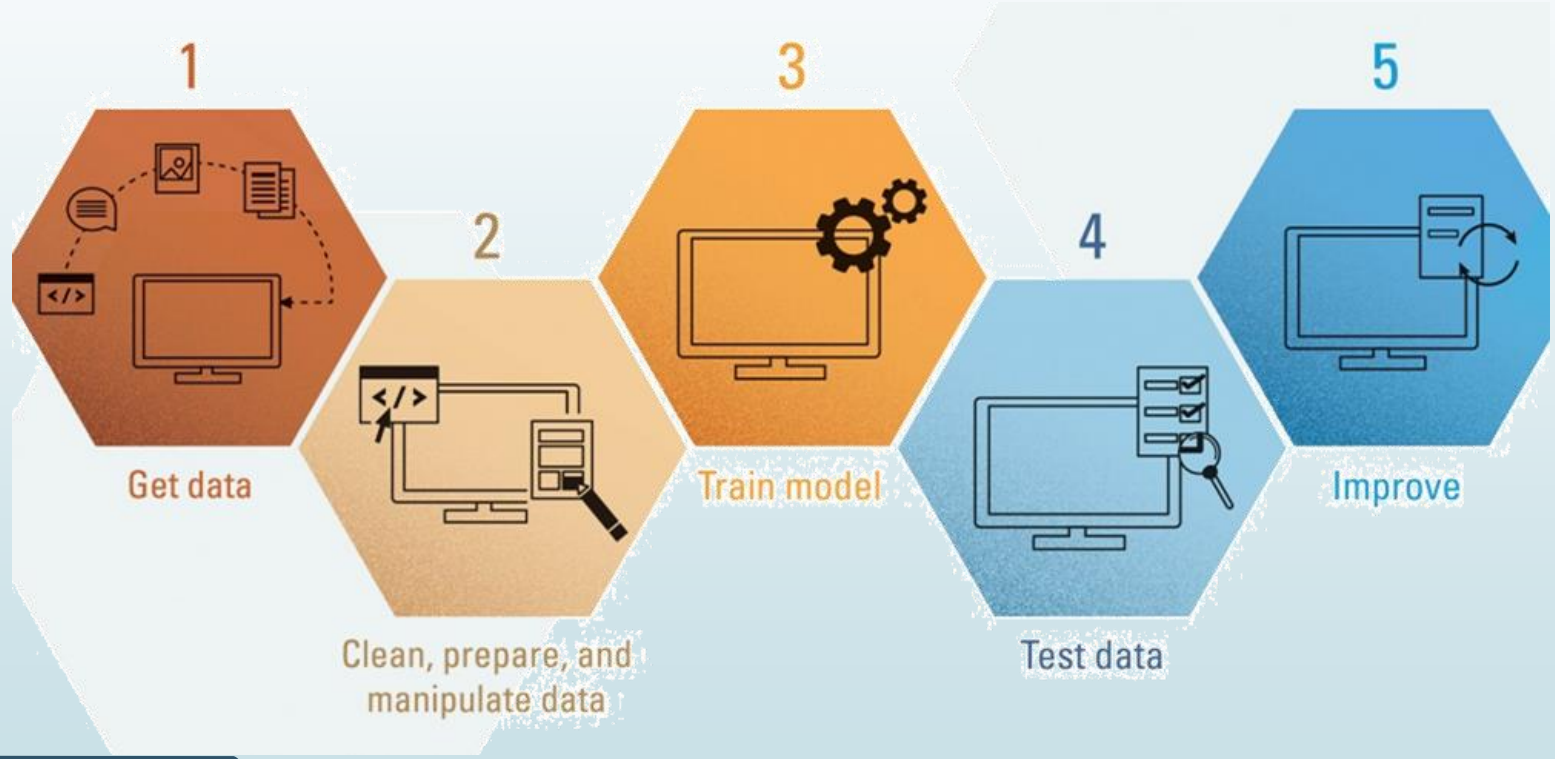
آماده‌سازی داده‌ها
(پیش‌پردازش و انتخاب ویژگی)

نادیه آرمین
گروه مهندسی کامپیوتر
دانشگاه فردوسی
پاییز ۹۹

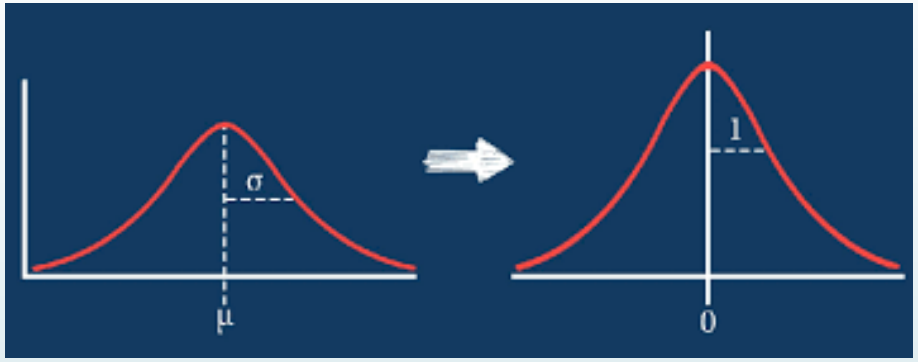
پروژه یادگیری ماشین

1. آماده‌سازی مسئله
2. شناخت داده‌ها
3. **آماده‌سازی داده‌ها**
4. ساخت و ارزیابی مدل
5. بهبود دقت
6. نهایی‌کردن مدل



سر فصل

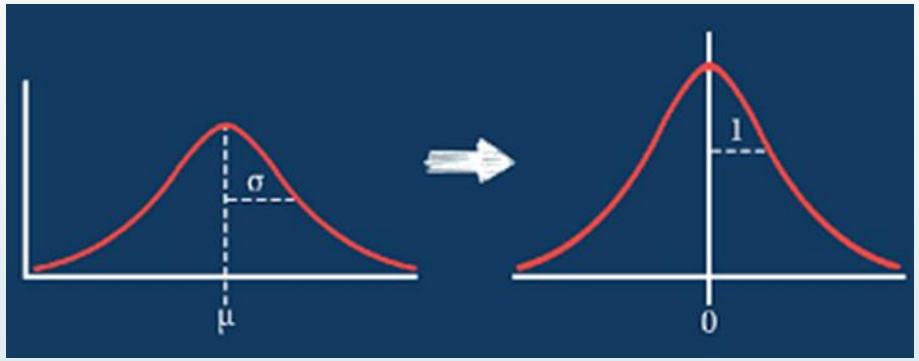
- آماده سازی داده‌ها
  - پیش پردازش داده‌ها:
    - تغییر مقیاس
    - استاندارد کردن
    - نرمال سازی
    - دودویی کردن
  - انتخاب ویژگی:
    - انتخاب یک متغییره
    - حذف ویژگی بازگشتی
    - تحلیل مولفه اصلی



آماده سازی داده‌ها

پیش پردازش داده‌ها:

- تغییر مقیاس
- استاندارد کردن
- نرمال سازی
- دودویی کردن



$$\frac{x - x_{min}}{x_{max} - x_{min}}$$

$$\vec{u} = \frac{\vec{v}}{\|\vec{v}\|}$$

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean
σ = Standard Deviation

پیش پردازش داده‌ها: تغییر مقیاس

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

کاربرد :

- در الگوریتم‌های بهینه‌سازی (گردایان نزولی)
- در الگوریتم‌های با ورودی وزن‌دار (رگرسیون، شبکه عصبی،..)
- در الگوریتم‌های با سنجه های فاصله‌ای (KNN،..)

```
#Rescale data (between 0 and 1)
from  pandas import read_csv
from numpy import set_printoptions
from sklearn.preprocessing import MinMaxScaler
filename = 'pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array=dataframe.values
#seprate array into input and output component
X=array[:,0:8]
Y=array[:,8]
scaler=MinMaxScaler(feature_range=(0,1)) # Transformed data into 0,1
rescaledX=scaler.fit_transform(X)
#summarize transformed data
set_printoptions(precision=3)
print(rescaledX[0:5,:])
```

پیش پردازش داده‌ها: استاندارد کردن

تبدیل صفات با توزیع گوسی و یک توزیع با میانگین ۰ و انحراف معیار ۱

کاربرد :

- در الگوریتم‌های با ورودی توزیع گوسی (رگرسیون خطی و رگرسیون لجستیک و LDA)

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean

σ = Standard Deviation

```
#Standardize data (0 mean, 1 stdev)
from  pandas import read_csv
from numpy import set_printoptions
from sklearn.preprocessing import StandardScaler
filename = 'pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array=dataframe.values
#seprate array into input and output component
X=array[:,0:8]
Y=array[:,8]
scaler=StandardScaler().fit(X)
rescaledX=scaler.fit_transform(X)
#summarize transformed data
set_printoptions(precision=3)
print(rescaledX[0:5,:])
```

پیش پردازش داده‌ها: نرمال سازی

تغییر مقیاس هر ردیف برای داشتن طول ۱(نرم واحد)

$$\vec{u} = \frac{\vec{v}}{\|\vec{v}\|}$$

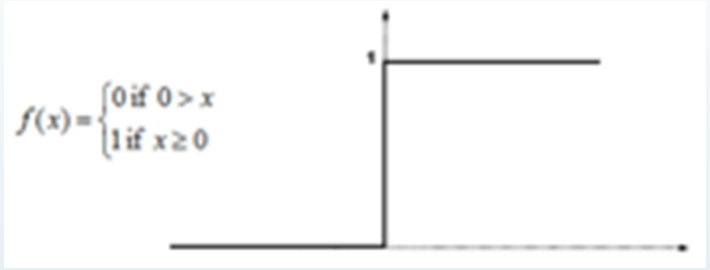
کاربرد:

- مجموعه داده تنک با صفاتی با مقیاسهای متفاوت
- در الگوریتم‌های دارای ورودی‌هایی وزن دار (رگرسیون، شبکه عصبی،..)
- در الگوریتم‌های با سنجه های فاصله‌ای (KNN،..)

```
#Normalize data (length 1)
from  pandas import read_csv
from numpy import set_printoptions
from sklearn.preprocessing import Normalizer
filename = 'pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array=dataframe.values
#seprate array into input and output component
X=array[:,0:8]
Y=array[:,8]
scaler=Normalizer().fit(X)
rescaledX=scaler.fit_transform(X)
#summarize transformed data
set_printoptions(precision=3)
print(rescaledX[0:5,:])
```

پیش پردازش داده‌ها: دودویی کردن

دودویی کردن با استفاده از یک آستانه



کاربرد:

- مقادیر قطعی از مقادیر احتمالاتی
- انتخاب ویژگی (سیاه و سفید کردن عکس)

```
#Binarization data (length 1)
from  pandas import read_csv
from numpy import set_printoptions
from sklearn.preprocessing import Binarizer
filename = 'pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array=dataframe.values
#seprate array into input and output component
X=array[:,0:8]
Y=array[:,8]
scaler=Binarizer(threshold=0.0).fit(X)
rescaledX=scaler.fit_transform(X)
#summarize transformed data
set_printoptions(precision=3)
print(rescaledX[0:5,:])
```

آماده سازی داده‌ها

انتخاب ویژگی:

- انتخاب یک متغییره
- حذف ویژگی بازگشتی
- تحلیل مولفه اصلی

آماده سازی داده‌ها

انتخاب ویژگی :

انتخاب ویژگی‌هایی از داده‌ها که بیشترین نقش را در پیش بینی خروجی دارند

چرا انتخاب ویژگی:

- کاهش بیش برازش(Overfitting)
- بهبود دقت مدل
- کاهش زمان آموزش

انتخاب ویژگی: انتخاب یک متغییره

استفاده از آزمون های آماری

- ویژگی هایی که قویترین رابطه را با خروجی دارند

```
from pandas import read_csv
from numpy import set_printoptions
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
filename = 'pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:, 0:8]
Y = array[:, 8]
# feature selection
test = SelectKBest(score_func=chi2, k=4)
fit = test.fit(X, Y)
# summarize scores
set_printoptions(precision=3)
print (fit.scores_)
features = fit.transform(X)
print (features[0:5, :])
```

انتخاب ویژگی:حذف ویژگی بازگشتی(REF)

استفاده از دقت مدل برای شناسایی صفات با بیشترین نقش در پیش بینی

```
from pandas import read_csv
from numpy import set_printoptions
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
filename = 'pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:, 0:8]
Y = array[:, 8]
```

```
# Feature extraction with RFE
model = LogisticRegression()
rfe = RFE(model, 3)
fit = rfe.fit(X, Y)
print ("Num Features:")
print (fit.n_features_)
print ("Selected Features:")
print (fit.support_)
print ("Num Features: ")
print(fit.ranking_)
```

انتخاب ویژگی:تحلیل مولفه اصلی(PCA)

استفاده از جبرخطی برای تبدیل مجموعه داده به شکلی فشرده

```
from pandas import read_csv
from sklearn.decomposition import PCA
filename = 'pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:, 0:8]
Y = array[:, 8]
pca = PCA(n_components=3)
fit = pca.fit(X)
# summarize componenta
print ("Explanied Variance: %s" % fit.explained_variance_ratio_)
print (fit.components_)
```