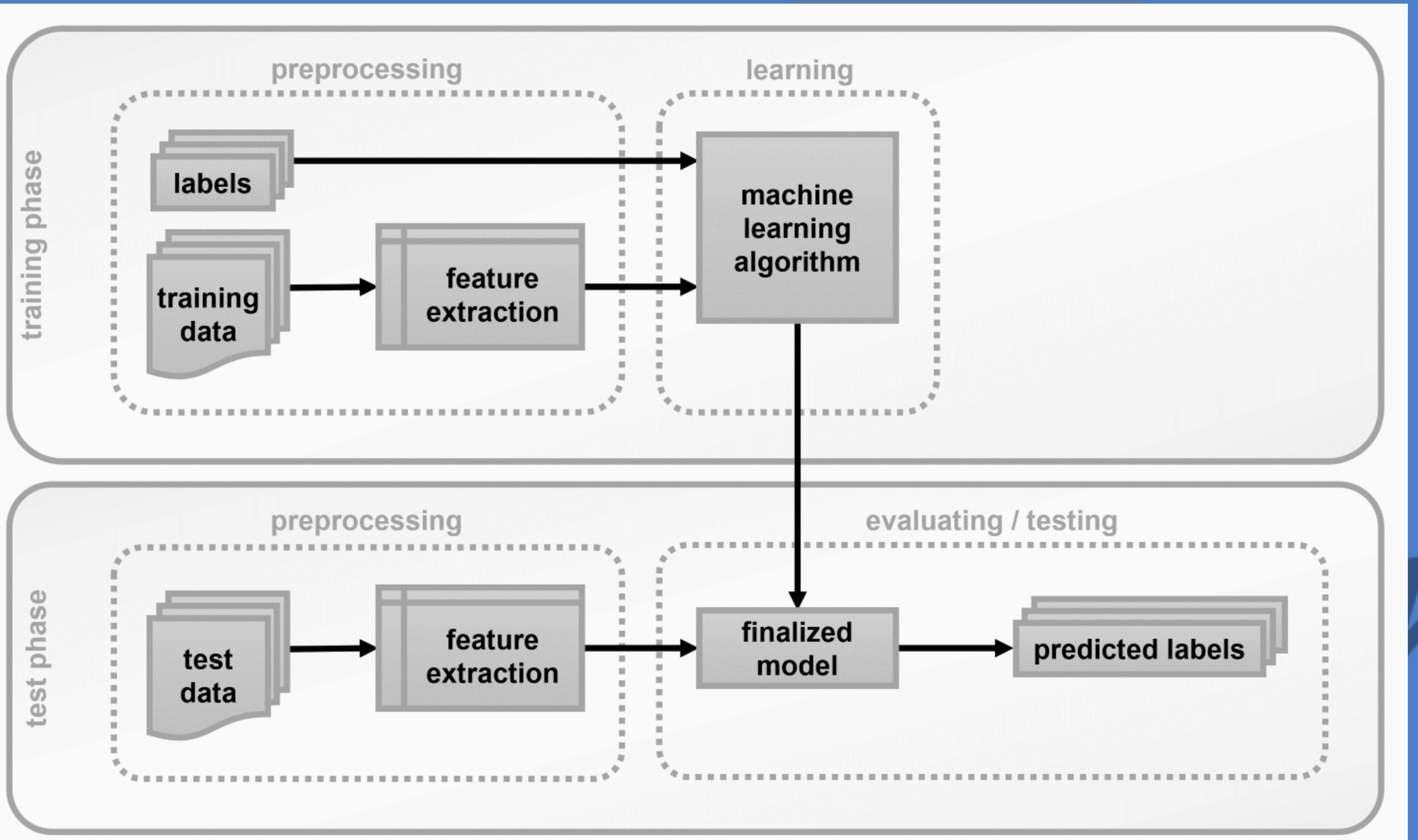
GATE WAY TO ML

MACHINE LEARNING WORKFLOW

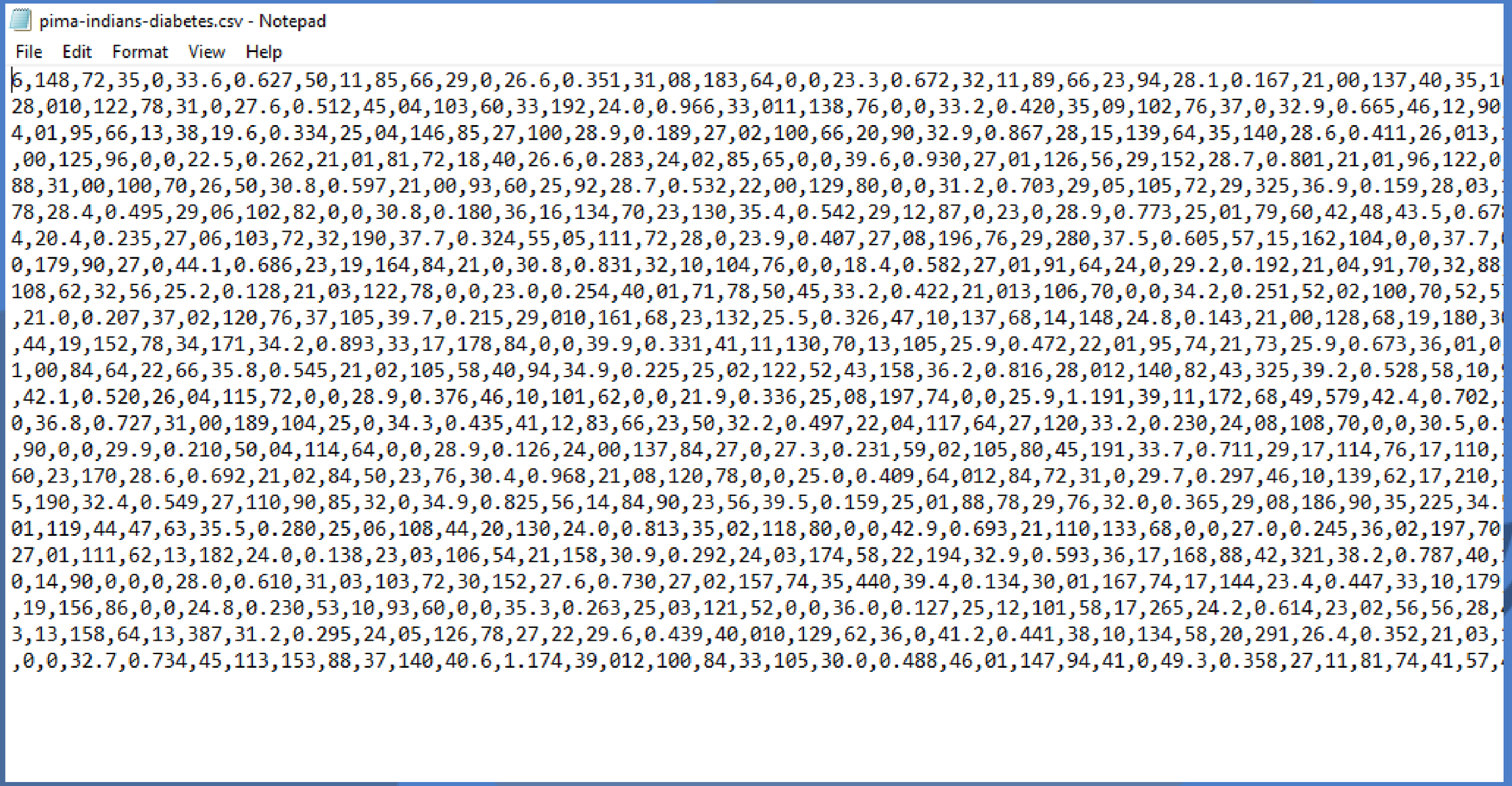

training phase
preprocessing
learning
labels
training data
feature extraction
machine learning algorithm
test phase
preprocessing
evaluating / testing
test data
feature extraction
finalized model
predicted labels

Let's Learn by doing

Raw Data

Pima Indians Dataset

This dataset describes the medical records for Pima Indians and whether or not each patient will have an onset of diabetes within five years. All of the inputattributes are numeric and the output variable to be predicted is binary (0 or 1). The data isfreely available from the UCI Machine Learning Repository.


pima-indians-diabetes.csv - Notepad
File Edit Format View Help
6,148,72,35,0,33.6,0.627,50,11,85,66,29,0,26.6,0.351,31,08,183,64,0,0,23.3,0.672,32,11,89,66,23,94,28.1,0.167,21,00,137,40,35,1
28,010,122,78,31,0,27.6,0.512,45,04,103,60,33,192,24.0,0.966,33,011,138,76,0,0,33.2,0.420,35,09,102,76,37,0,32.9,0.665,46,12,90
4,01,95,66,13,38,19.6,0.334,25,04,146,85,27,100,28.9,0.189,27,02,100,66,20,90,32.9,0.867,28,15,139,64,35,140,28.6,0.411,26,013,
,00,125,96,0,0,22.5,0.262,21,01,81,72,18,40,26.6,0.283,24,02,85,65,0,0,39.6,0.930,27,01,126,56,29,152,28.7,0.801,21,01,96,122,0
88,31,00,100,70,26,50,30.8,0.597,21,00,93,60,25,92,28.7,0.532,22,00,129,80,0,0,31.2,0.703,29,05,105,72,29,325,36.9,0.159,28,03,
78,28.4,0.495,29,06,102,82,0,0,30.8,0.180,36,16,134,70,23,130,35.4,0.542,29,12,87,0,23,0,28.9,0.773,25,01,79,60,42,48,43.5,0.67
4,20.4,0.235,27,06,103,72,32,190,37.7,0.324,55,05,111,72,28,0,23.9,0.407,27,08,196,76,29,280,37.5,0.605,57,15,162,104,0,0,37.7,
0,179,90,27,0,44.1,0.686,23,19,164,84,21,0,30.8,0.831,32,10,104,76,0,0,18.4,0.582,27,01,91,64,24,0,29.2,0.192,21,04,91,70,32,88
108,62,32,56,25.2,0.128,21,03,122,78,0,0,23.0,0.254,40,01,71,78,50,45,33.2,0.422,21,013,106,70,0,0,34.2,0.251,52,02,100,70,52,5
,21.0,0.207,37,02,120,76,37,105,39.7,0.215,29,010,161,68,23,132,25.5,0.326,47,10,137,68,14,148,24.8,0.143,21,00,128,68,19,180,3
,44,19,152,78,34,171,34.2,0.893,33,17,178,84,0,0,39.9,0.331,41,11,130,70,13,105,25.9,0.472,22,01,95,74,21,73,25.9,0.673,36,01,0
1,00,84,64,22,66,35.8,0.545,21,02,105,58,40,94,34.9,0.225,25,02,122,52,43,158,36.2,0.816,28,012,140,82,43,325,39.2,0.528,58,10,
,42.1,0.520,26,04,115,72,0,0,28.9,0.376,46,10,101,62,0,0,21.9,0.336,25,08,197,74,0,0,25.9,1.191,39,11,172,68,49,579,42.4,0.702,
0,36.8,0.727,31,00,189,104,25,0,34.3,0.435,41,12,83,66,23,50,32.2,0.497,22,04,117,64,27,120,33.2,0.230,24,08,108,70,0,0,30.5,0.
,90,0,0,29.9,0.210,50,04,114,64,0,0,28.9,0.126,24,00,137,84,27,0,27.3,0.231,59,02,105,80,45,191,33.7,0.711,29,17,114,76,17,110,
60,23,170,28.6,0.692,21,02,84,50,23,76,30.4,0.968,21,08,120,78,0,0,25.0,0.409,64,012,84,72,31,0,29.7,0.297,46,10,139,62,17,210,
5,190,32.4,0.549,27,110,90,85,32,0,34.9,0.825,56,14,84,90,23,56,39.5,0.159,25,01,88,78,29,76,32.0,0.365,29,08,186,90,35,225,34.
01,119,44,47,63,35.5,0.280,25,06,108,44,20,130,24.0,0.813,35,02,118,80,0,0,42.9,0.693,21,110,133,68,0,0,27.0,0.245,36,02,197,70
27,01,111,62,13,182,24.0,0.138,23,03,106,54,21,158,30.9,0.292,24,03,174,58,22,194,32.9,0.593,36,17,168,88,42,321,38.2,0.787,40,
0,14,90,0,0,0,28.0,0.610,31,03,103,72,30,152,27.6,0.730,27,02,157,74,35,440,39.4,0.134,30,01,167,74,17,144,23.4,0.447,33,10,179
,19,156,86,0,0,24.8,0.230,53,10,93,60,0,0,35.3,0.263,25,03,121,52,0,0,36.0,0.127,25,12,101,58,17,265,24.2,0.614,23,02,56,56,28,
3,13,158,64,13,387,31.2,0.295,24,05,126,78,27,22,29.6,0.439,40,010,129,62,36,0,41.2,0.441,38,10,134,58,20,291,26.4,0.352,21,03,
,0,0,32.7,0.734,45,113,153,88,37,140,40.6,1.174,39,012,100,84,33,105,30.0,0.488,46,01,147,94,41,0,49.3,0.358,27,11,81,74,41,57,

LOAD CSV FILE

The Python API provides the module CSV and the function reader()that can be used to load CSV files. Once loaded, you can convert the CSV data to a NumPy array and use it for machine learning.

```
# Load CSV Using Python Standard Library
import csvimport numpy
filename ='pima-indians-diabetes.data.csv'
raw_data = open(filename,'rb')
reader = csv.reader(raw_data, delimiter=',',
quoting=csv.QUOTE_NONE)
x = list(reader)
data = numpy.array(x).astype('float')
print(data.shape)
```

- Take a peek at your raw data.
- Review the dimensions of your dataset.
- Review the data types of attributes in your data.
- Summarize the distribution of instances across classes in your dataset.
- Summarize your data using descriptive statistics
- Understand the relationships in your data using correlations
- Review the skew of the distributions of each attribute.

UNDERSTAND YOUR DATA

DESCRIPTIVE STATISTICS

ML WORKFLOW

PEEK AT YOUR DATA

```
# View first 20 rowsfrom pandas import
read_csvfilename = "pima-indians-
diabetes.data.csv"
names =
['preg','plas','pres','skin','test','mass','pedi',
'age','class']
data = read_csv(filename, names=names)
peek = data.head(20)
print (peek)
```

DATA TYPE FOR EACH ATTRIBUTE

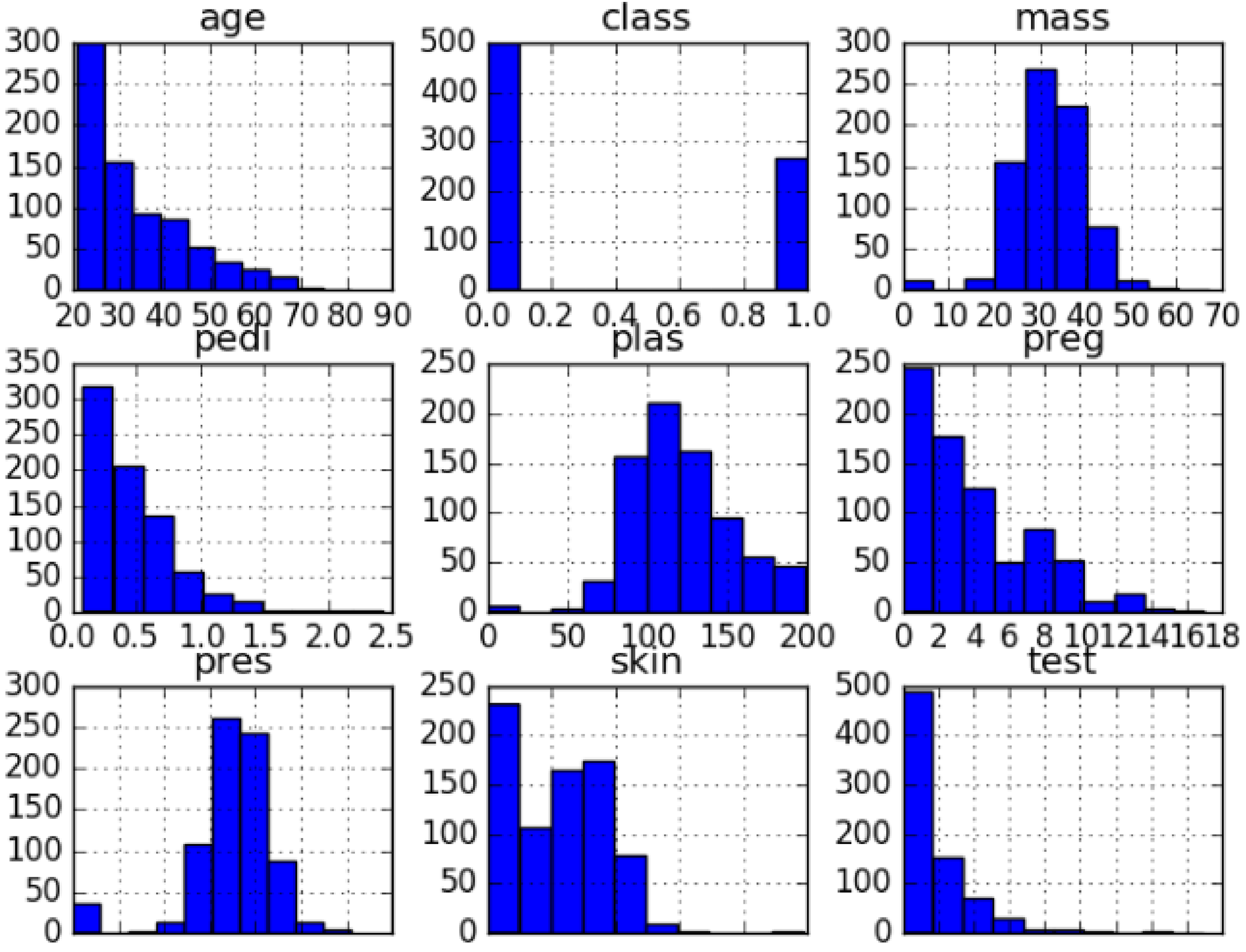
```
# Data Types for Each Attribute
from pandas import read_csv
filename = "pima-indians-diabetes.data.csv"
names =
['preg','plas','pres','skin','test','mass','pedi',
'age','class']
data = read_csv(filename, names=names)
types = data.dtypes
print(types)
```

DESCRIPTIVE STATISTICS

```
# Statistical Summary
from pandas import read_csv
from pandas import set_option
filename = "pima-indians-diabetes.data.csv"
names =
['preg','plas','pres','skin','test','mass','pedi','
age','class']
data = read_csv(filename, names=names)
set_option('display.width', 100)
set_option('precision', 3)
description = data.describe()
print(description)
```

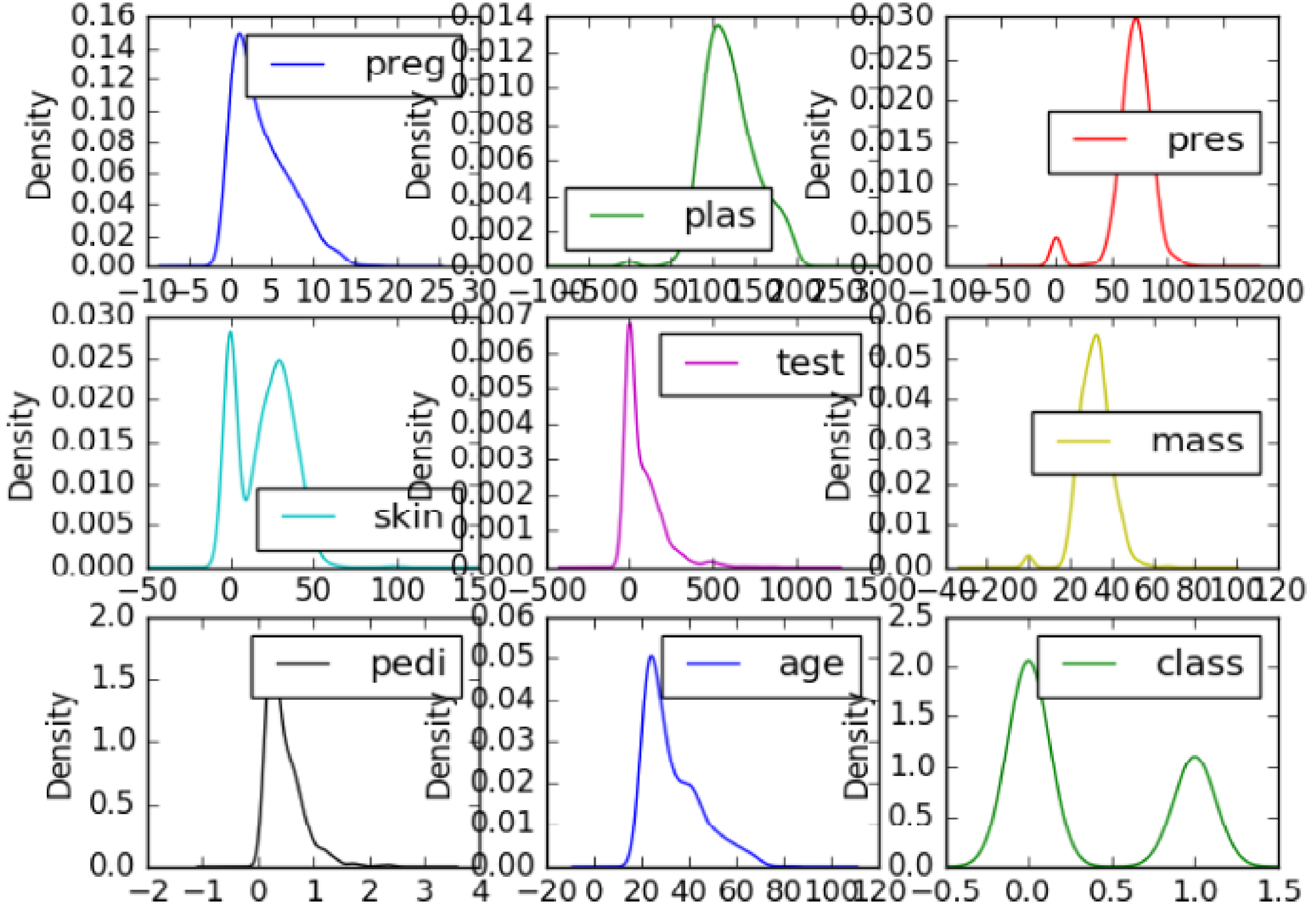
UNDERSTAND YOUR DATA WITH VISUALIZATION

```
# Univariate Histograms
from matplotlib import pyplot
from pandas import read_csv
filename ='pima-indians-diabetes.data.csv'
names =
['preg','plas','pres','skin','test','mass','pedi','
age','class']
data = read_csv(filename, names=names)
data.hist()
pyplot.show()
```


age
class
mass
pedi
plas
preg
pres
skin
test

UNDERSTAND YOUR DATA WITH VISUALIZATION

```
# Univariate Density Plotsfrom matplotlib
import pyplotfrom pandas
import read_csvfile
name ='pima-indians-diabetes.data.csv'
names =
['preg','plas','pres','skin','test','mass','pedi','
age','class']
data = read_csv(filename, names=names)
data.plot(kind='density', subplots=True, layout=
(3,3), sharex=False)
pyplot.show()
```


preg
plas
pres
skin
test
mass
pedi
age
class
Density

PREPARE YOUR DATA FOR MACHINE LEARNING

ML WORKFLOW

```
# 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.data.csv'
names =
['preg','plas','pres','skin','test','mass','pedi','age',
'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
# separate array into input and output components
X = array[:,0:8]
Y = array[:,8]
scaler = MinMaxScaler(feature_range=(0, 1))
rescaledX = scaler.fit_transform(X)
# summarize transformed
dataset_printoptions(precision=3)
```

PREPARE YOUR DATA FOR MACHINE LEARNING

ML WORKFLOW

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

PREPARE YOUR DATA FOR MACHINE LEARNING

ML WORKFLOW

```
# binarization
from sklearn.preprocessing import Binarizer
from pandas import read_csv
from numpy import set_printoptions
filename ='pima-indians-diabetes.data.csv
'names =
['preg','plas','pres','skin','test','mass','pedi','age',
'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
# separate array into input and output components
X = array[:,0:8]
Y = array[:,8]
binarizer = Binarizer(threshold=0.0).fit(X)
binaryX = binarizer.transform(X)
# summarize transformed
dataset_printoptions(precision=3)
print(binaryX[0:5,:])
```

FEATURE SELECTION FOR MACHINE LEARNING

ML WORKFLOW

```
# Feature Extraction with Univariate Statistical Tests (Chi-squared for
classification)
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.data.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]
test = SelectKBest(score_func=chi2, k=4)
fit = test.fit(X, Y)
# summarize
scoresset_printoptions(precision=3)print(fit.scores_)
features = fit.transform(X)
# summarize selected
 featuresprint(features[0:5,:])
```

FEATURE SELECTION FOR MACHINE LEARNING

ML WORKFLOW

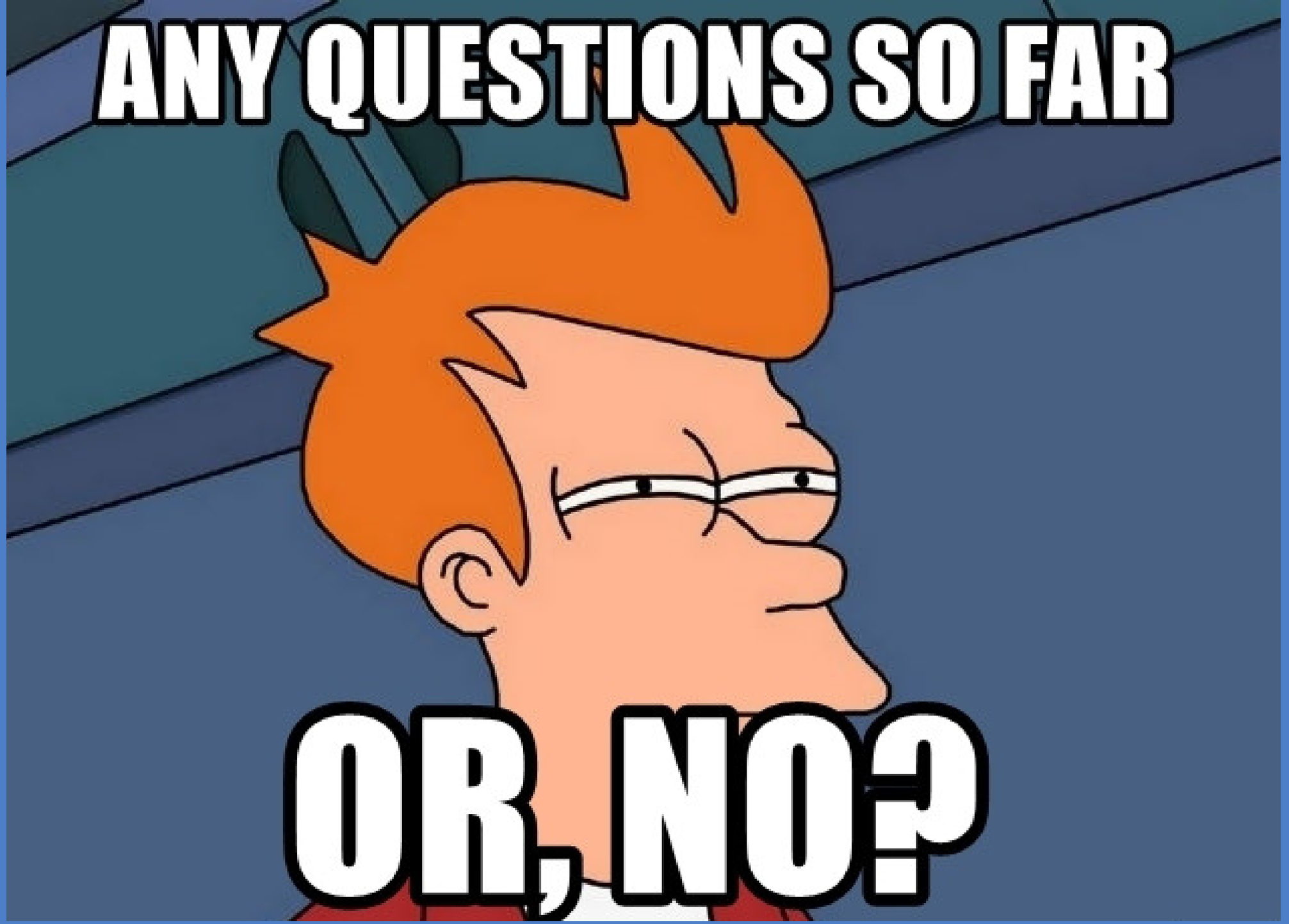
```
# Feature Importance with Extra Trees Classifier
from pandas import read_csv
from sklearn.ensemble import ExtraTreesClassifier
# load data
filename ='pima-indians-diabetes.data.csv'
names =
['preg','plas','pres','skin','test','mass','pedi','age','clas
s']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
# feature extraction
model = ExtraTreesClassifier()
model.fit(X, Y)
print(model.feature_importances_)
```

SPLIT INTO TRAIN AND TEST SETS

```
# Evaluate using a train and a test set
from pandas import read_csv
from sklearn.model_selection
import train_test_split
from sklearn.linear_model import LogisticRegression
filename ='pima-indians-diabetes.data.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]
test_size = 0.33
seed = 7
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=test_size,random_state=seed)
model = LogisticRegression()
model.fit(X_train, Y_train)
result = model.score(X_test, Y_test)
print("Accuracy: %.3f%%") % (result*100.0)
```

K-FOLD CROSS-VALIDATION

```
# Evaluate using Cross Validation
from pandas import read_csv
from sklearn.model_selection
import KFoldfrom sklearn.model_selection
import cross_val_score
from sklearn.linear_model
import LogisticRegression
filename ='pima-indians-diabetes.data.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]
num_folds = 10
seed = 7
kfold = KFold(n_splits=num_folds, random_state=seed)
model = LogisticRegression()
results = cross_val_score(model, X, Y, cv=kfold)
print("Accuracy: %.3f%% (%.3f%%)") % (results.mean()*100.0,
results.std()*100.0)
```


ANY QUESTIONS SO FAR
OR, NO?

SPOT-CHECK CLASSIFICATION ALGORITHMS

Logistic Regression.

Linear Discriminant Analysis.

k-Nearest Neighbors.

Naive Bayes.

Classification and Regression Trees

Support Vector Machines.

SPOT-CHECK REGRESSION ALGORITHMS

Linear Regression.
Ridge Regression.
LASSO Linear Regression.
Elastic Net Regression.
k-Nearest Neighbors.
Classification and Regression Trees
Support Vector Machines.

ALGORITHM EVALUATION METRICS

Classification Metrics

ML WORKFLOW

Classification Accuracy.

Logarithmic Loss.

Area Under ROC Curve.

Confusion Matrix.

Classification Report.

CLASSIFICATION METRICS

Classification Accuracy.

ML WORKFLOW

```
# Cross Validation Classification Accuracy
from pandas import read_csv
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
filename ='pima-indians-diabetes.data.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]
```

CLASSIFICATION METRICS

Classification Accuracy.

```
kfold = KFold(n_splits=10, random_state=7)
model = LogisticRegression()
scoring ='accuracy'
results = cross_val_score(model, X, Y, cv=kfold,
scoring=scoring)
print("Accuracy: %.3f (%.3f)") % (results.mean(),
results.std())
```

CLASSIFICATION METRICS

Confusion Matrix.

ML WORKFLOW

```
# Cross Validation Classification Confusion Matrix
from pandas import read_csv
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
filename ='pima-indians-diabetes.data.csv'
names =
['preg','plas','pres','skin','test','mass','pedi','ag
e','class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
test_size = 0.33
```

CLASSIFICATION METRICS

Confusion Matrix.

```
seed = 7
X_train, X_test, Y_train, Y_test =
train_test_split(X, Y,
test_size=test_size,random_state=seed)
model = LogisticRegression()
model.fit(X_train, Y_train)
predicted = model.predict(X_test)
matrix = confusion_matrix(Y_test, predicted)
print(matrix)
```

CLASSIFICATION METRICS

Classification Report

ML WORKFLOW

```
# Cross Validation Classification Confusion Matrix
from pandas import read_csv
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
filename ='pima-indians-diabetes.data.csv'
names =
['preg','plas','pres','skin','test','mass','pedi','a
ge','class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
test_size = 0.33
```

CLASSIFICATION METRICS

Classification Report

```
seed = 7
X_train, X_test, Y_train, Y_test =
train_test_split(X, Y,
test_size=test_size,random_state=seed)
model = LogisticRegression()
model.fit(X_train, Y_train)
predicted = model.predict(X_test)
report = classification_report(Y_test, predicted)
print(report)
```

ALGORITHM EVALUATION METRICS

Regression Accuracy.

Mean Absolute Error.

Mean Squared Error.

R2

REGRESSION ACCURACY.

Mean Absolute Error.

ML WORKFLOW

```
# Cross Validation Regression MAE
from pandas import read_csv
from sklearn.model_selection import KFold
from sklearn.model_selection import
cross_val_score
from sklearn.linear_model import LinearRegression
filename ='housing.csv'
names =
['CRIM','ZN','INDUS','CHAS','NOX','RM','AGE','DIS'
,'RAD','TAX','PTRATIO','B','LSTAT','MEDV']
dataframe = read_csv(filename,
delim_whitespace=True, names=names)
array = dataframe.values
X = array[:,0:13]
Y = array[:,13]
```

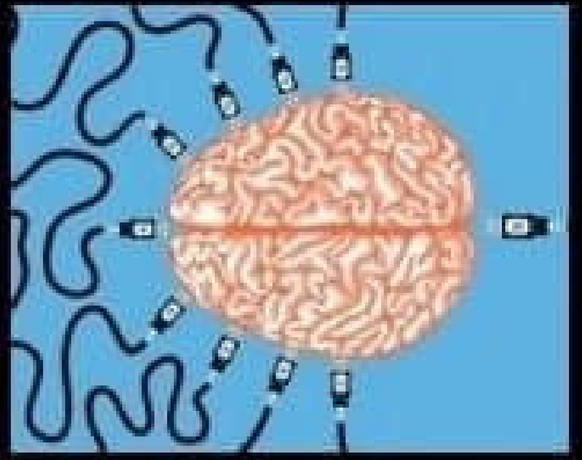
REGRESSION ACCURACY.

Mean Absolute Error.

```
kfold = KFold(n_splits=10, random_state=7)
model = LinearRegression()
scoring ='neg_mean_absolute_error'
results = cross_val_score(model, X, Y, cv=kfold,
scoring=scoring)
print("MAE: %.3f (%.3f)") % (results.mean(),
results.std())
```

Machine Learning

What society thinks I do.



What my friends thinks I do.

What computer scientists think I do.

What my boss thinks I do.

What I think I do.

What I really do.

Now you have the tools
let's sum up

next section on

ML PROBLEMS