# Comparison of Classification Algorithms

Prof.Dr. Bahadır AKTUĞ

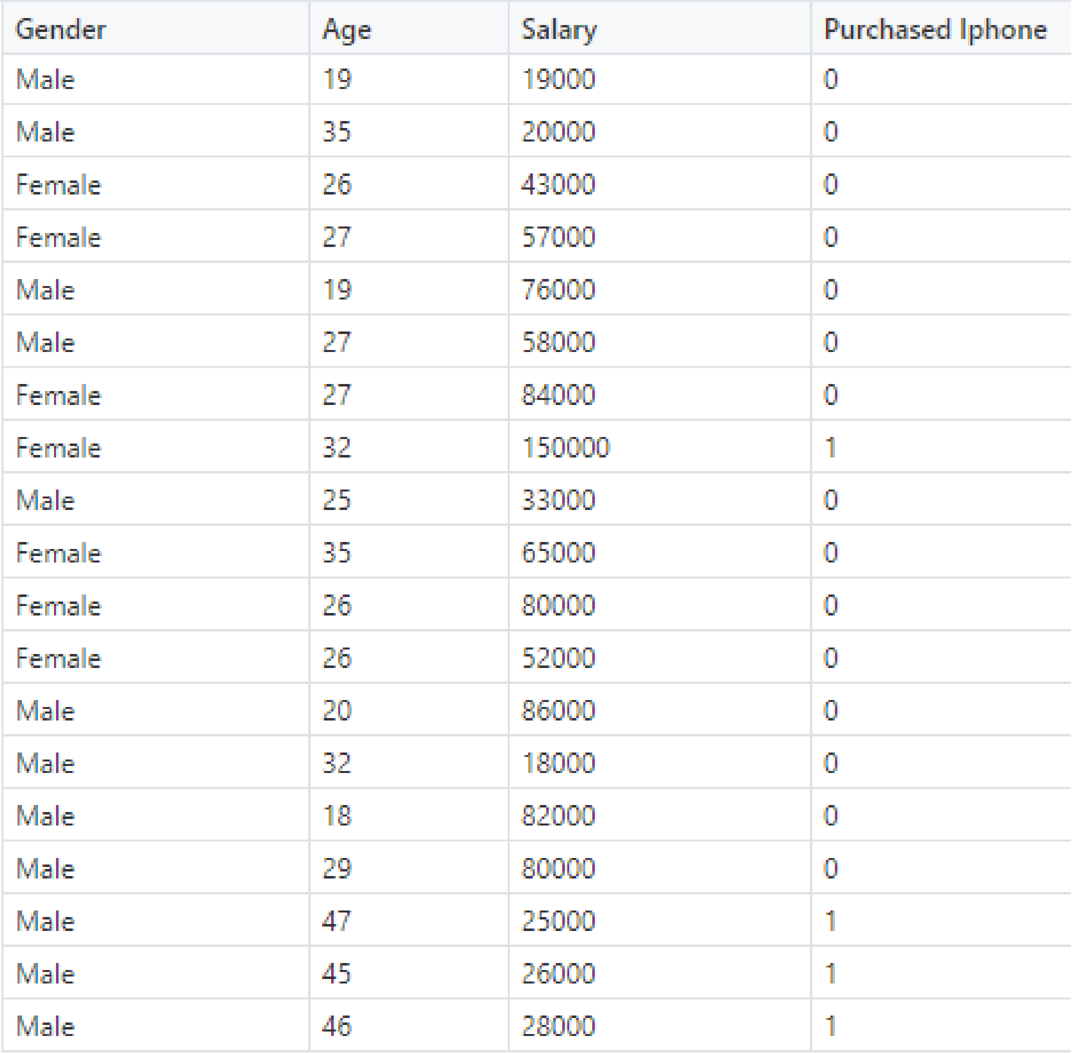
Machine Learning with Python

**Compiled from sources given in the references.*

# Classification Algorithms

- A simple comparison will be made for classification algorithms.
- The algorithms to be used:
  - Logistic Regression
  - KNN (K Nearest Neighbors)
  - Kernel SVM (Support Vector Machine)
  - Naive Bayes
  - Decision Tree
  - Random Forest
- It should be noted that this is just a demonstration and highly data-dependent, i.e. Different performance would be obtained with different data.
- It is only through trial and error and checking the performance metrics, we can narrow down and pick certain algorithms.

# Data (Iphone Purchase)



| Gender | Age | Salary | Purchased Iphone |
|---|---|---|---|
| Male | 19 | 19000 | 0 |
| Male | 35 | 20000 | 0 |
| Female | 26 | 43000 | 0 |
| Female | 27 | 57000 | 0 |
| Male | 19 | 76000 | 0 |
| Male | 27 | 58000 | 0 |
| Female | 27 | 84000 | 0 |
| Female | 32 | 150000 | 1 |
| Male | 25 | 33000 | 0 |
| Female | 35 | 65000 | 0 |
| Female | 26 | 80000 | 0 |
| Female | 26 | 52000 | 0 |
| Male | 20 | 86000 | 0 |
| Male | 32 | 18000 | 0 |
| Male | 18 | 82000 | 0 |
| Male | 29 | 80000 | 0 |
| Male | 47 | 25000 | 1 |
| Male | 45 | 26000 | 1 |
| Male | 46 | 28000 | 1 |

- Iphone purchasing records
- 400 records
- Independent Parameters:
  - Gender (categorical data)
  - Age
  - Salary
- Dependent parameter: Purchased Iphone or not

# Step-1: Load Data

- We need to assign the independent variables “Gender”, “Salary” and “Age” to X. The dependent variable “Purchased iphone” is dependent variable and should be assigned to y.

```
import pandas as pd
dataset = pd.read_csv("iphone_purchase_records.csv")
X = dataset.iloc[:,:-1].values
y = dataset.iloc[:, 3].values
```

# Step 2: Convert Gender to Number

- We have a categorical variable "Gender" that we have to convert to number. We will use the class LabelEncoder to convert Gender to number.

```
from sklearn.preprocessing import LabelEncoder
labelEncoder_gender = LabelEncoder()
X[:,0] = labelEncoder_gender.fit_transform(X[:,0])
```

# Step 3: Feature Scaling

- Except for the classifiers the Decision Tree and Random Forest, all other classifiers require normalized data. The easiest way to do it to use StandardScaler function of sci-kit learn module. StandardScaler.
- StandardScaler standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X = sc.fit_transform(X)
```

# Step 4: Comparing

- We will use 10-fold cross validation to evaluate each algorithm and we will find the mean accuracy and the standard deviation accuracy.
- First, we will create a list and add objects of the different classifiers we want to evaluate. Then we loop through the list and use the cross_val_score method to get the accuracies.
- Each time model is assigned a different algorithm (which can easilty be done using a list)

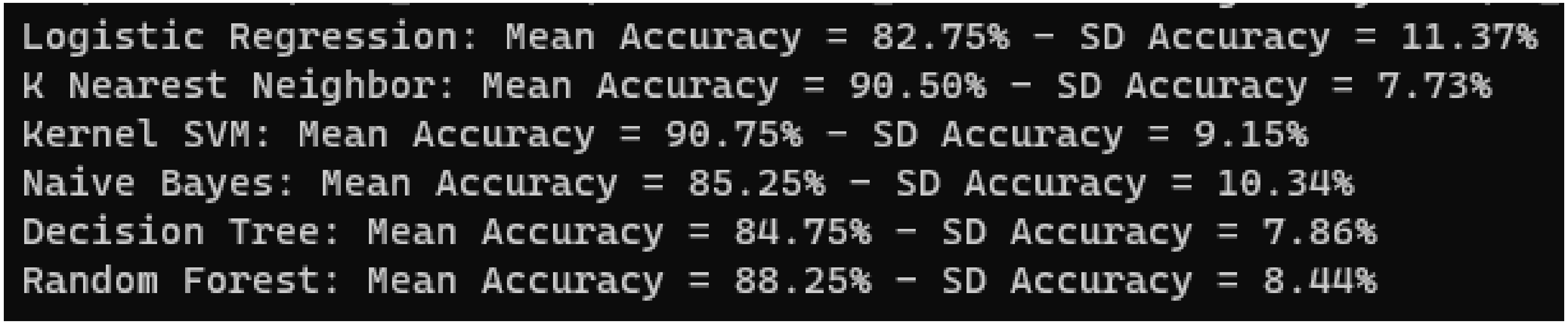
```
kfold = KFold(n_splits=10, random_state=7)
result = cross_val_score(model, X, y, cv=kfold, scoring='accuracy')
print("%s: Mean Accuracy = %.2f%% - SD Accuracy = %.2f%%" %
(name, result.mean()*100, result.std()*100))
```

# Step 5: Parameters & Results

- The input parameters of the models are as follows:

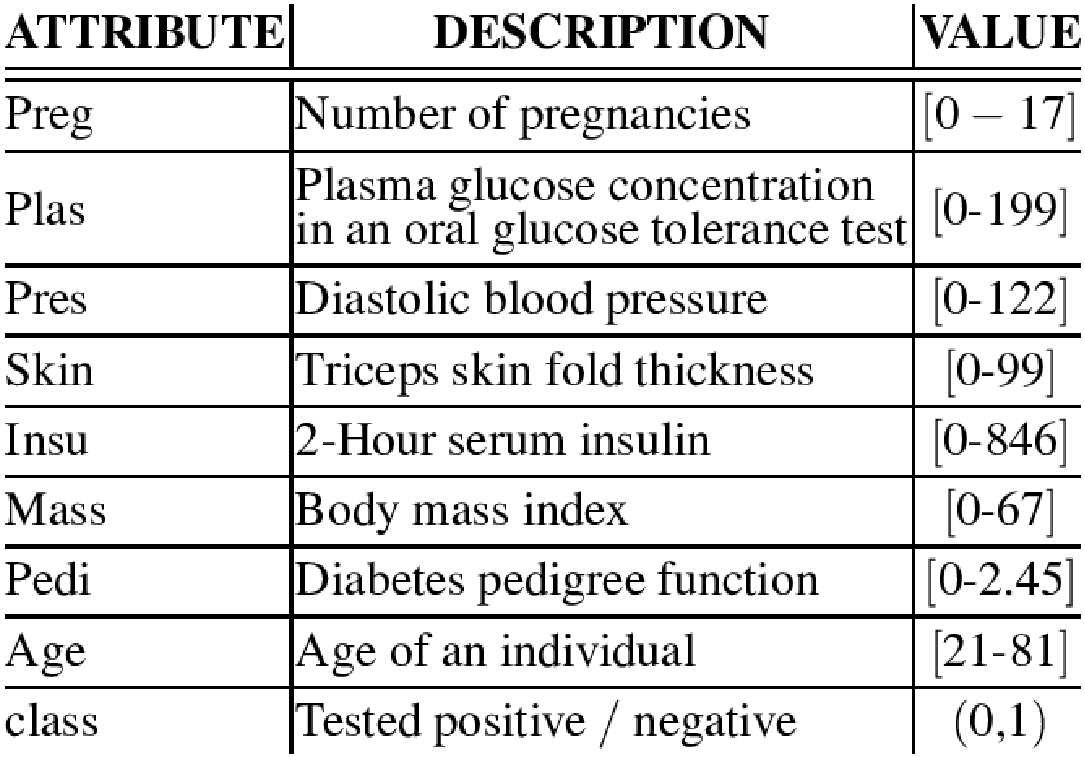
```
LogisticRegression(solver="liblinear")
KNeighborsClassifier(n_neighbors=5, metric="minkowski",p=2)
SVC(kernel = 'rbf',gamma='scale')
GaussianNB()
DecisionTreeClassifier(criterion = "entropy")
RandomForestClassifier(n_estimators=100, criterion="entropy")
```

- The performance results are as follows:



```
Logistic Regression: Mean Accuracy = 82.75% - SD Accuracy = 11.37%
K Nearest Neighbor: Mean Accuracy = 90.50% - SD Accuracy = 7.73%
Kernel SVM: Mean Accuracy = 90.75% - SD Accuracy = 9.15%
Naive Bayes: Mean Accuracy = 85.25% - SD Accuracy = 10.34%
Decision Tree: Mean Accuracy = 84.75% - SD Accuracy = 7.86%
Random Forest: Mean Accuracy = 88.25% - SD Accuracy = 8.44%
```

# Data (Pima Indians Diabetes Database )



| ATTRIBUTE | DESCRIPTION | VALUE |
|---|---|---|
| Preg | Number of pregnancies | [0 – 17] |
| Plas | Plasma glucose concentration in an oral glucose tolerance test | [0-199] |
| Pres | Diastolic blood pressure | [0-122] |
| Skin | Triceps skin fold thickness | [0-99] |
| Insu | 2-Hour serum insulin | [0-846] |
| Mass | Body mass index | [0-67] |
| Pedi | Diabetes pedigree function | [0-2.45] |
| Age | Age of an individual | [21-81] |
| class | Tested positive / negative | (0,1) |

- Diabetes attributes of Pima Indians
- 768 records
- 8 Independent Parameters:
- Dependent parameter: Class of whether positive or negative

# Tested Classification Algorithms

- Another simple comparison will be made for classification algorithms.
- The algorithms to be used:
  - Logistic Regression
  - Linear Discriminant Analysis
  - K-Nearest Neighbors
  - Classification and Regression Trees
  - Naive Bayes
  - Support Vector Machines
- Again, it should be noted that this is just a demonstration and highly data-dependent, i.e. Different performance would be obtained with different data and it is only through trial and error and checking the performance metrics, we can narrow down and pick certain algorithms.

# Step-1: Load Data

- We need to assign the independent variables “Gender”, “Salary” and “Age” to X. The dependent variable “Purchased iphone” is dependent variable and should be assigned to y.

```
fname = ‘pima-indians-diabetes.txt’
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = pandas.read_csv(url, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
```

# Step 2: Import the Algorithms

- All the algorithms will be used with the default parameters.
- ML Algorithms are located in different sub modules of sklearn.

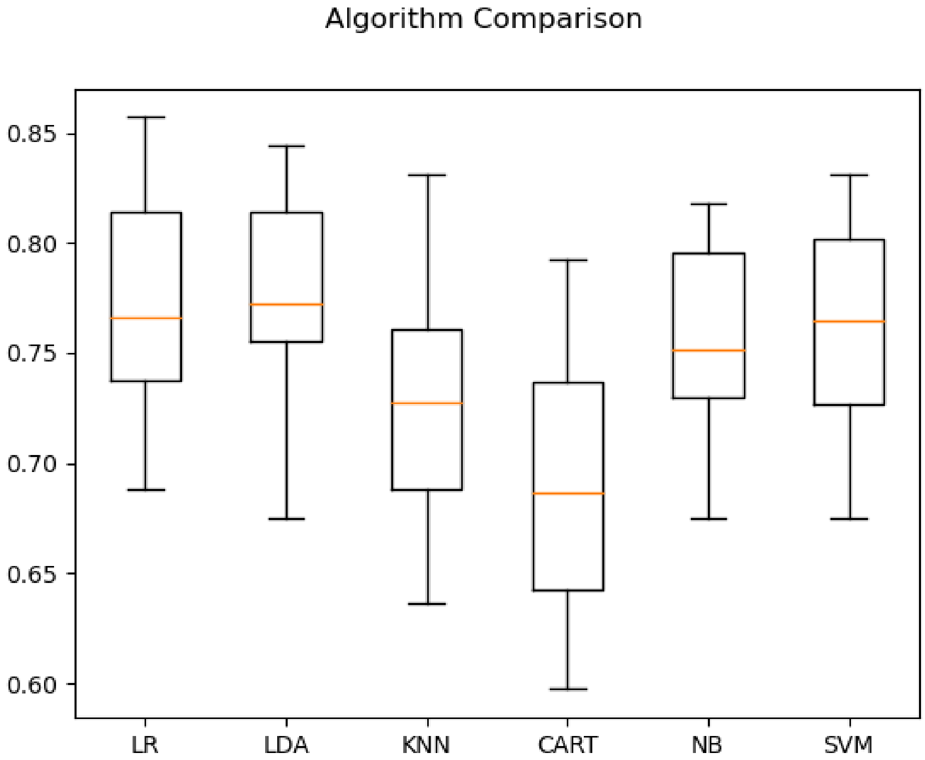
```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn import model_selection
```

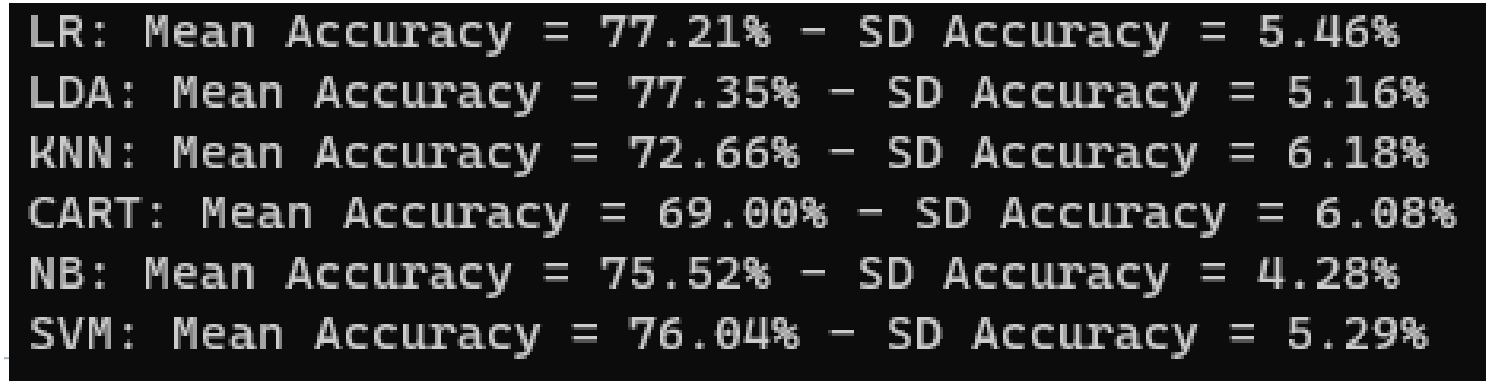
# Step 3: Comparing

- We will use 10-fold cross validation with random seed of 7 as in the previous example to evaluate each algorithm and we will find the mean accuracy and the standard deviation accuracy.
- For cross validation we will be using accuracy again.
- Each time model is assigned a different algorithm (which can easilty be done using a list)

```
kfold = KFold(n_splits=10, random_state=7)
result = cross_val_score(model, X,Y, cv=kfold, scoring='accuracy')
print("%s: Mean Accuracy = %.2f%% - SD Accuracy = %.2f%%" %
(name, result.mean()*100, result.std()*100))
```

# Step 4: Results





```
LR: Mean Accuracy = 77.21% - SD Accuracy = 5.46%
LDA: Mean Accuracy = 77.35% - SD Accuracy = 5.16%
KNN: Mean Accuracy = 72.66% - SD Accuracy = 6.18%
CART: Mean Accuracy = 69.00% - SD Accuracy = 6.08%
NB: Mean Accuracy = 75.52% - SD Accuracy = 4.28%
SVM: Mean Accuracy = 76.04% - SD Accuracy = 5.29%
```

# References

1 https://scikit-learn.org/

2 https://towardsdatascience.com/

3 McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython 2nd Edition.

4 Albon, C. (2018). Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning

5 Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 1st Edition

6 Müller, A. C., Guido, S. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists

7 Burkov, A. (2019). The Hundred-Page Machine Learning Book.

8 Burkov, A. (2020). Machine Learning Engineering.

9 Goodrich, M.T., Tamassia, R., Goldwasser, M.H. (2013). Data Structures and Algorithms in Python, Wiley.

10 https://towardsdatascience.com

11 https://towardsdatascience.com/machine-learning-project-17-compare-classification-algorithms-87cb50e1cb60

12 https://machinelearningmastery.com/compare-machine-learning-algorithms-python-scikit-learn/

13 https://developers.google.com/edu/python/

14 http://learnpythonthehardway.org/book/