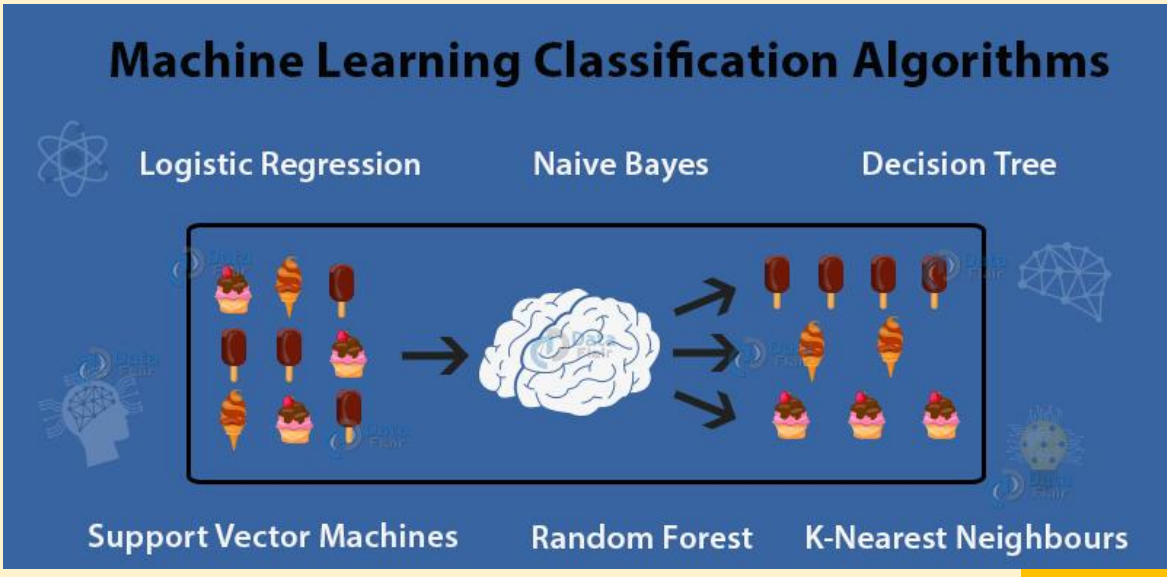
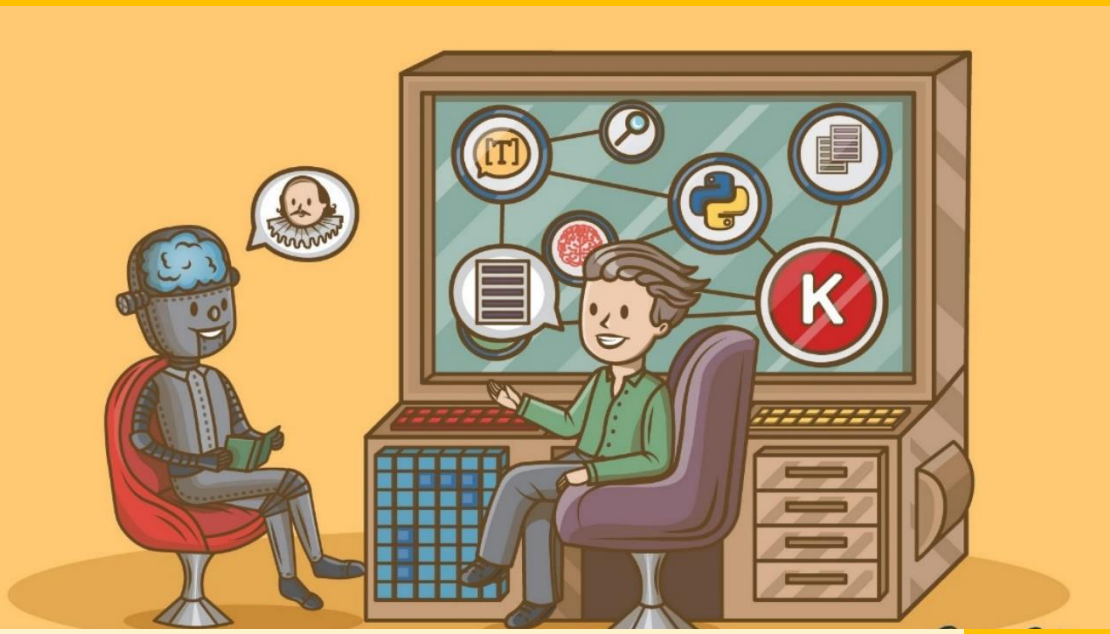
Pandas   Python

Mini Project - Building Classifiers



-by SHIVAM KUMAR GIRI

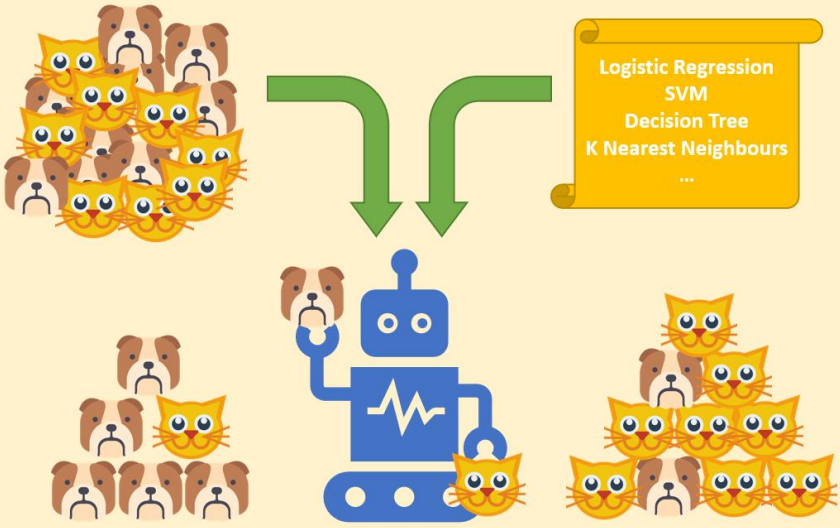


PROBLEM STATEMENTS

A dataset collected in a **cosmetics shop** showing details of customers and whether or not they responded to a special offer to buy a new lip-stick is shown in table below. You are to write a report covering following points: -

- Build classifiers with logistic regression and at least one other classification method like SVM or Decision tree.
- Find goodness of fits for each classifier.
- Compare performance among classifiers.

ABSTRACT



A classifier is a hypothesis or discrete-valued function that is used to assign (categorical) class labels to particular data points. In the email classification example, this classifier could be a hypothesis for labeling emails as spam or non-spam. Here we are going to classify whether or not customer will respond to a special offer to buy a new lip-stick. Here we used 6 classifiers:

- Logistic Regression
- Decision Tree
- Naïve Bayes
- SVM
- K Neighbour Classification
- Random Forest

METHODOLOGY

Step 1: Import the libraries

Libraries such as pandas, matplotlib, numpy, tabulate, sklearn, Matplotlib and tabulate is used for data visualization, pandas for creating data frames and data exploration, numpy for numerical computations and sklearn is used for importing classifiers and various parameters regarding Linear Regression.

Step 2: Import and the data

Here we have salary_data.csv file to get the data showing a cosmetics shop showing details of customers and whether or not they responded to a special offer to buy a new lip-stick.

The dataset consists of following 5 variables: -

- **age**: denotes age category 0 for age less than 21, 1 for age in between 21 and 35 and 2 for age greater than 35.
- **income**: 2 for high income, 1 for medium income group and 0 for low income group.
- **gender**: 0 for male, 1 for female.
- **marital status**: 1 for married and 0 for single.
- **buys**: 1 if user bought, 0 otherwise.

Step 3: Explore and Analyze data

```
data.head()
```

	age	income	gender	marital status	buys
0	0	2	0	0	0
1	0	2	0	1	0
2	1	2	0	0	1
3	2	1	0	0	1
4	2	0	1	0	1

```
data.shape
```

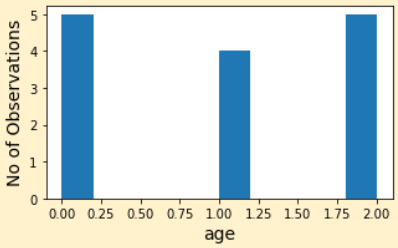
(14, 5)

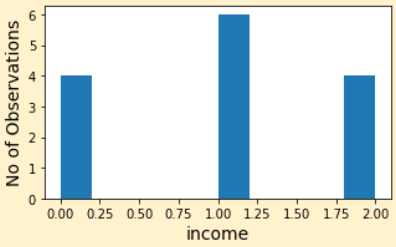
```
data.describe()
```

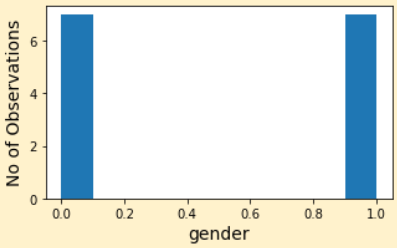
	age	income	gender	marital status	buys
count	14.000000	14.000000	14.000000	14.000000	14.000000
mean	1.000000	1.000000	0.500000	0.500000	0.642857
std	0.877058	0.784465	0.518875	0.518875	0.497245
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.250000	0.000000	0.000000	0.000000
50%	1.000000	1.000000	0.500000	0.500000	1.000000
75%	2.000000	1.750000	1.000000	1.000000	1.000000
max	2.000000	2.000000	1.000000	1.000000	1.000000

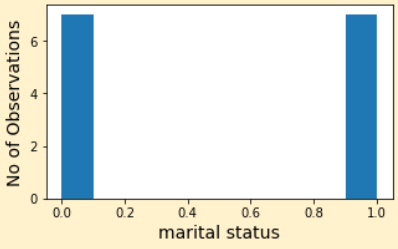
Hence, we see that there is no missing values in the dataset

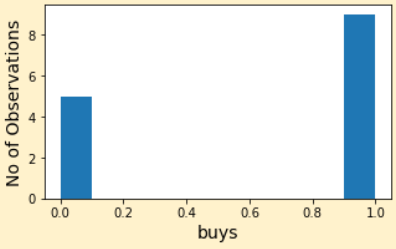
Step 4: Analyze the exploratory data



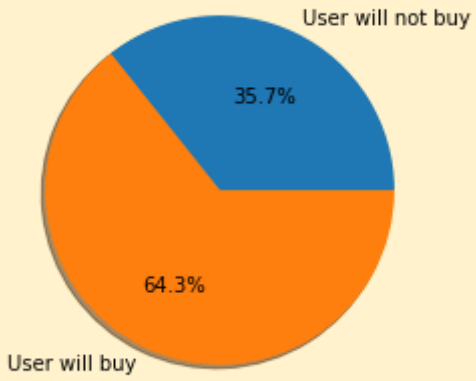








Step 5: plotting of target data



Step 6: Standardize the input data

Here I used StandardScaler() to get the standardized input data for the classifiers.

Classifier 1: Decision Tree

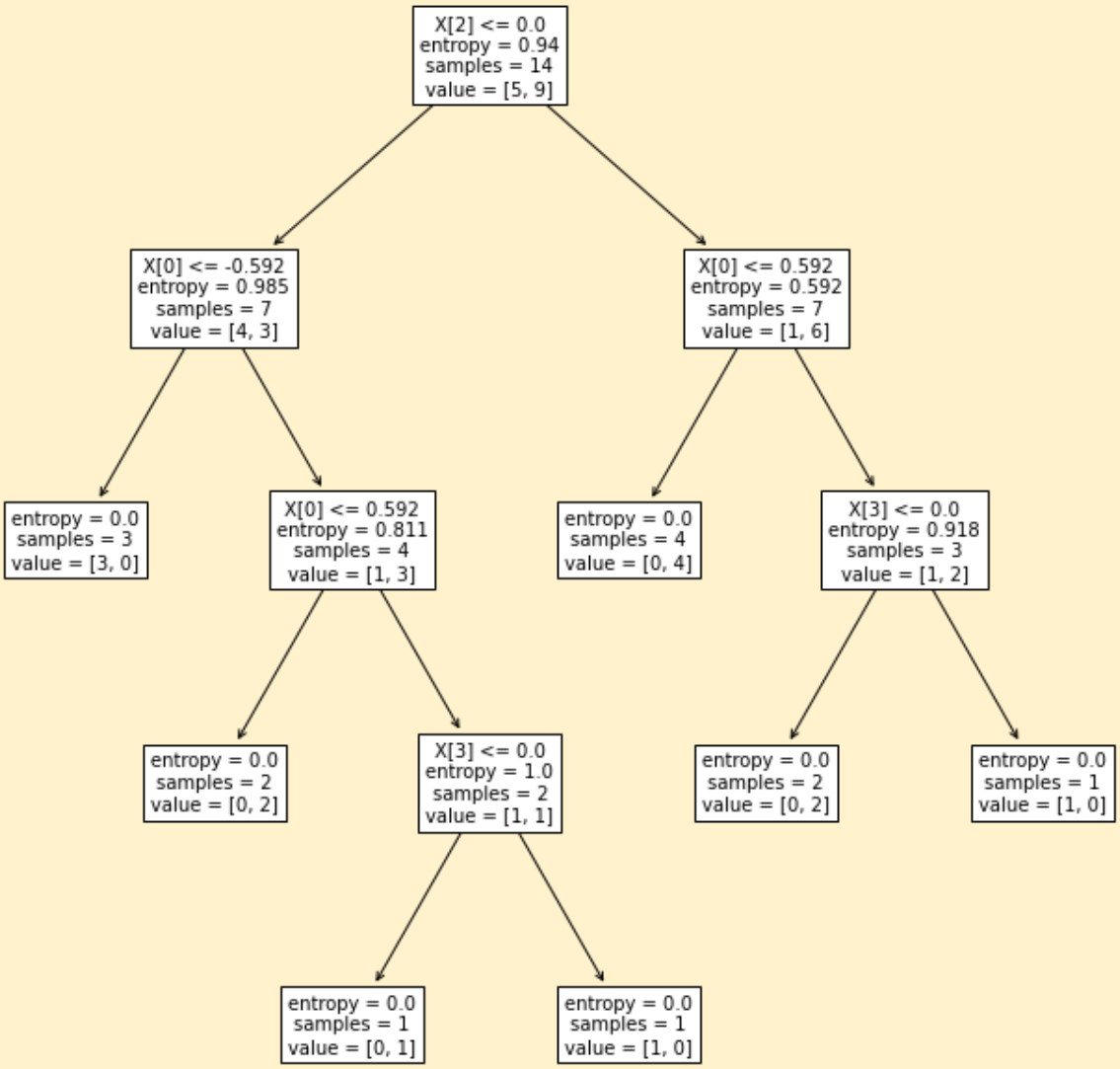
Decision tree is the most powerful and popular tool for classification and prediction. **A Decision tree is a flowchart like tree structure**, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Predicted Output: [0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0]

Accuracy: 100.00
Confusion Matrix: [[5 0]
[0 9]]

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	1.00	1.00	1.00	9
Accuracy			1.00	14
macro avg	1.00	1.00	1.00	14
weighted avg	1.00	1.00	1.00	14



Classifier 2: Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis**, logistic regression (or logit regression) is estimating the parameters of a logistic model** (a form of binary regression). Using Linear Regression, **all predictions >= 0.5 can be considered as 1 and rest all < 0.5 can be considered as 0**.

Predicted Output: [1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

Accuracy: 71.43
Confusion Matrix: [[2 3]
[1 8]]

Classification Report:

	Precision	recall	f1-score	support
0	0.67	0.40	0.50	5
1	0.73	0.89	0.80	9
accuracy			0.71	14
macro avg	0.70	0.64	0.65	14
weighted avg	0.71	0.71	0.69	14

Classifier 3: SMV

In machine learning, support vector machines (SVMs, also support vector networks) **are supervised learning models** with associated learning algorithms that analyze data used for classification and regression analysis.

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), **the algorithm outputs an optimal hyperplane which categorizes** new examples.

Predicted Output: [1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

Accuracy: 71.43
Confusion Matrix: [[2 3]
[1 8]]

Classification Report:

	precision	recall	f1-score	support
0	0.67	0.40	0.50	5
1	0.73	0.89	0.80	9
accuracy			0.71	14
macro avg	0.70	0.64	0.65	14
weighted avg	0.71	0.71	0.69	14

Classifier 4: Random Forest

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly **known as bagging**. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called **Bootstrap**.

Predicted Output: [0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0]

Accuracy: 100.00

Confusion Matrix: [[5 0]
[0 9]]

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	1.00	1.00	1.00	9
accuracy			1.00	14
macro avg	1.00	1.00	1.00	14
weighted avg	1.00	1.00	1.00	14

Classifier 5: KNeighborsClassifier

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).

Predicted Output: [0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0]

Accuracy: 78.57
Confusion Matrix: [[4 1]
[2 7]]

Classification Report:

	precision	recall	f1-score	support
0	0.67	0.80	0.73	5
1	0.88	0.78	0.82	9
accuracy			0.79	14
macro avg	0.77	0.79	0.78	14
weighted avg	0.80	0.79	0.79	14

Classifier 6: Naive Bayes

Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Predicted Output: [0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0]

Accuracy: 78.57
Confusion Matrix: [[4 1]
[2 7]]

Classification Report:

	precision	recall	f1-score	support
0	0.67	0.80	0.73	5
1	0.88	0.78	0.82	9
accuracy			0.79	14
macro avg	0.77	0.79	0.78	14
weighted avg	0.80	0.79	0.79	14

Tabulate the data

Accuracy by various Models of Classifiers

Penalty Methods		Accuracy	Confusion Matrix
Logistic Regression	:	71.43	[[2 3] [1 8]]
Decison tree	:	100	[[5 0] [0 9]]
SMV	:	71.43	[[2 3] [1 8]]
Random Forest	:	100	[[5 0] [0 9]]
K-Neighbour Classifier	:	78.57	[[4 1] [2 7]]
Naive Bayes	:	78.57	[[4 1] [2 7]]

CONCLUSION

Hence, we see that the **Decision tree and Random Forest Classifier Perform the best with 100% Accuracy,** while **SMV and Logistic Regression Perform the least by 71.43% Accuracy**. **K-Neighbour Classifier with k=4 and Naïve Bayes have Accuracy of 78.57%.**

THE END