

THE GOAL IS GET 100 % FOR Diabetes Expectation Class

This dataset contains the following:

	#	age	#	bs_fast	#	bs_pp	#	plasma_r	#	plasma_f	#	hba1c	type	class
1		50		6.8		8.8		11.2		7.2		62	Type1	true
2		31		5.2		6.8		10.9		4.2		33	Normal	false
3		32		6.8		8.8		11.2		7.2		62	Type1	true
4		21		5.7		5.8		10.7		4.8		49	Normal	false
5		33		6.8		8.8		11.2		7.2		62	Type1	true
6		30		5.2		7.4		8.7		5.6		41	Normal	false
7		26		5.8		4.2		11.4		8.4		53	Type2	true

In this project we chose

- low volume data and its concept.
- We prepared it properly
- We chose a suitable algorithm< DecisionTreeClassifier>

Thus, we were able to obtain a 100% class expectation. And because this type of decision does not tolerate any error, you need a result of 0 or 1 diabetes or non-diabetes.

We also used machine learning as an essential element in data science to make all data numeric

Libraries AND methods we used: Pandas, NumPy, Matplotlib, Seaborn
Warnings

sklearn.metrics sklearn.model_selection

sklearn.preprocessing sklearn.tree

This dataset can be found at [data. World] (Diabetes type dataset - dataset by abelvikas data.World).

- We reviewed the data on jupyter and we do EDA.
- We used the following presentation tools:
 - 1- Class count plot
 - 2-Histogram of each feature
 - 3- pairplot
 - 4-heatmap(data[top_corr_features])
- We separated the data to train and test.
- We succeeded in building the model and apply it.
- DecisionTreeClassifier(criterion='entropy', random_state=0)
- We tested the model by comparing between y_pred and y_test

```
In [31]: y_pred = Classifier.predict(x_test)

In [32]: y_pred

Out[32]: array([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 0, 0, 2, 2, 0, 0,
2, 2, 1, 2, 0, 0, 0, 0, 1, 0, 1, 0, 2, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 2, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 1, 1, 2, 0, 2, 2, 0, 0,
1, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 2, 0, 1, 0, 0,
1, 0, 0, 2, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 2, 1,
1, 2], dtype=int64)

In [33]: y_test

Out[33]: array([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 0, 0, 2, 2, 0, 0,
2, 2, 1, 2, 0, 0, 0, 0, 1, 0, 1, 0, 2, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 2, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 1, 1, 2, 0, 2, 2, 0, 0,
1, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 2, 0, 1, 0, 0,
1, 0, 0, 2, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 2, 1,
1, 2], dtype=int64)
```

- Then we tested the model's results using two methods

1- confusion_matrix 2- accuracy_score and got a 100% perfect prediction.

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

cm

array([[110,  0,  0],
       [ 0, 20,  0],
       [ 0,  0, 26]], dtype=int64)

from sklearn.metrics import accuracy_score
print('Accuracy score is : ', accuracy_score(y_test, y_pred))

Accuracy score is : 1.0
```

SADAIA TUWAIQ BOOTCAMP T5 4 WEEKS

Prepare by: Atif Eid Alnefaie

Supervisor: Dr-Mariam – Dr-Shikh
Data scientist: Abdulrahman – Rasha