Chapter 8 : Part 5

**Deep Learning**

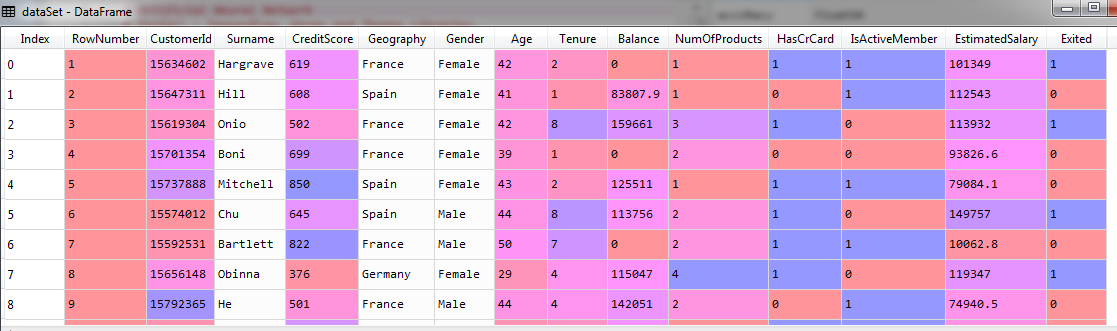
**ANN:** Predict new Data-point

* Here we already trained our ANN, using the given data (six-month observation of customers of a Bank).
* Now a new customer's data is arrived to us. We have to predict if the customer will leave or stay in the bank by using our ANN-model that we just built. The new data is given below:
* Use our *ANN* *model* to *predict* if the customer with the *following informations* will *leave* the bank:

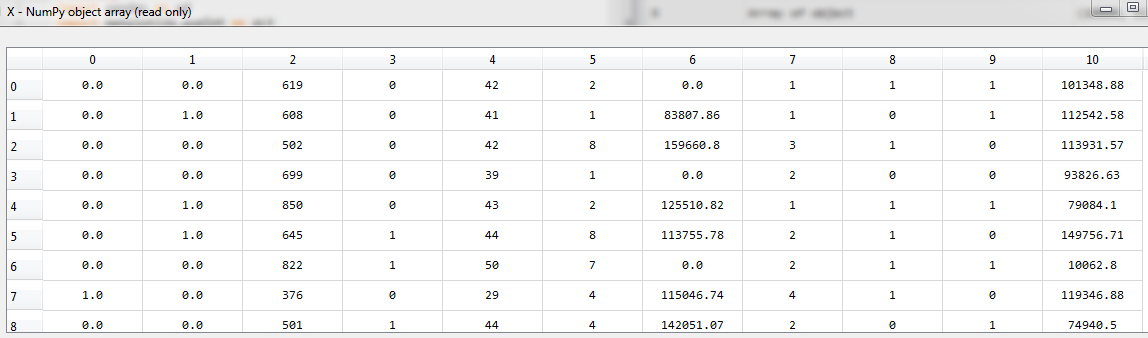
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Geography: | France | | Credit Score: | 600 | | Gender: | Male | | Age: | 40 years old | | Tenure: | 3 years | | Balance: | $60000 | | Number of Products: | 2 | | Does this customer have a credit card ? | Yes | | Is this customer an Active Member: | Yes | | Estimated Salary: | $50000 |  * So should we say goodbye to that customer ? | * What we need to do:  1. ***Arrange*** the new data in correct order (same order of our data-set), 2. Find the ***right*** ***dummy*** ***variable*** for categorical variables (Geography, Gender etc), 3. Transform the data into a ***NumPy*** ***array***, 4. ***Scale*** the data-point, 5. Make the ***prediction*** by converting the ***probability***. |

* Arrange the new data in correct order: Let's compare our original data-set to Encoded-data-set feature-matrix ***X***.
* i.e. Feature-matrix ***X*** after ***encoding-categorical data*** and before train-test split and feature-scaling, so that we can compare the variables to original data-set.

**Original-Data**



**Encoded Feature-matrix *X***



new\_dt\_pt = **np.array**([[0.0, 0.0, 600, 1, 40, 3, 60000.0, 2, 1, 1, 50000.0]])

* From, ***row*** no. ***0***, ***1*** and ***7*** we notice France = (0.0, 0.0), Spain = (0.0, 1.0), Germany = (1.0, 0.0) represented using the dummy variables.
* Since Dummy variables of Geography appears *first* in our *feature matrix*, we need not to *rearrange* the *columns* for our new data-point **new\_dt\_pt**.
* Also we not need to re-arrange other columns, they are in right order.
* From, ***row*** no. ***0*** to ***4*** and ***5***,***6*** we notice Male = ***1*** and female = ***0***
* Credit card : yes = ***1***, no =***0***
* Active-member : yes = ***1***, no =***0***
* Hence we represent our new data-point **new\_dt\_pt** as:

0.0, 0.0, 600, 1, 40, 3, 60000.0, 2, 1, 1, 50000.0

* Now if we use it as a list, it won't be a row, it will be a **vector/column**:

[0.0, 0.0, 600, 1, 40, 3, 60000.0, 2, 1, 1, 50000.0]

* To make it as row of the feature matrix we define it as a list-of-list i.e. **[[]],** as 1x11 matrix

[[0.0, 0.0, 600, 1, 40, 3, 60000.0, 2, 1, 1, 50000.0]]

* NumPy array: We also need to convert it to ***NumPy*** array, we use ***np.array*** (here numpy imported as **np**):

**np.array**([[0.0, 0.0, 600, 1, 40, 3, 60000.0, 2, 1, 1, 50000.0]])

We put it to the variable called **new\_dt\_pt**.

* Scale: Then we *scale* this new data-point **new\_dt\_pt**, using our *Standard Scaler* **st\_x**.
* Predict: Finally *predict* the new data-point **new\_dt\_pt**, using the classifier **ann\_classifier** and ***transform*** the returned *probability* into **True/False** using *threshold* value ***0.5***.

new\_dt\_pt = **np.array**([[0.0, 0.0, 600, 1, 40, 3, 60000.0, 2, 1, 1, 50000.0]])

new\_dt\_pt= **st\_x.transform**(new\_dt\_pt) # *scaling*

predict\_data\_pt = (**ann\_classifier.predict**(new\_dt\_pt) **>** 0.5)

**All code for new data-point prediction**

# *first create a 2D "NumPy array" in our X\_train's format.*

    # *it will be smilar to a single row of our X\_train*

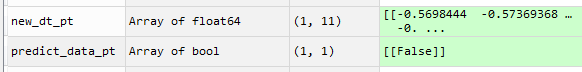
    # *2 "[" used to define a single row of a 2-D array*

new\_dt\_pt = **np.array**([[0.0, 0.0, 600, 1, 40, 3, 60000.0, 2, 1, 1, 50000.0]])

new\_dt\_pt= **st\_x.transform**(new\_dt\_pt) # *scaling*

predict\_data\_pt = (**ann\_classifier.predict**(new\_dt\_pt) **>** 0.5) # *Predict the data-point*

* Result: The prediction is "**False**". That is the customer *not going to leave the Bank*. Since leave = 1, stay = 0 in dependent variable, "**Exited**". Here ***True/False*** is represented by ***1*** or ***0***.



Hence we *don't say Goodbye* to that customer.