Εξόρυξη Δεδομένων και Αλγόριθμοι Μάθησης

Εργαστηριακή Άσκηση
Εαρινό Εξάμηνο 2020-21

Στοιχεία:
Ονοματεπώνυμο: Βασιλική Στάμου
ΑΜ: 1059543 Έτος: 4ο

Ερώτημα 1

Υποερώτημα Α.

Ανάλυση του dataset

Το αρχείο healthcare-dataset-stroke-data.csv περιέχει πληροφορίες ασθενών οι οποίες προσανατολίζονται στην ύπαρξη ή μη εγκεφαλικού επεισοδίου. Αναλυτικότερα, υπάρχουν 11 κατηγορίες δεδομένων για 5110 ανώνυμους ασθενείς που προσδιορίζονται από ένα μοναδικό αριθμό id. Οι κατηγορίες δεδομένων είναι gender, age, hypertension, heart_disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, smoking_status και stroke.

Τύποι χαρακτηριστικών :

Αριθμητικοί (συνεχείς): avg_glucose_level , bmi

Διατακτικοί (ακέραιοι): age , hypertension , heart_disease , stroke

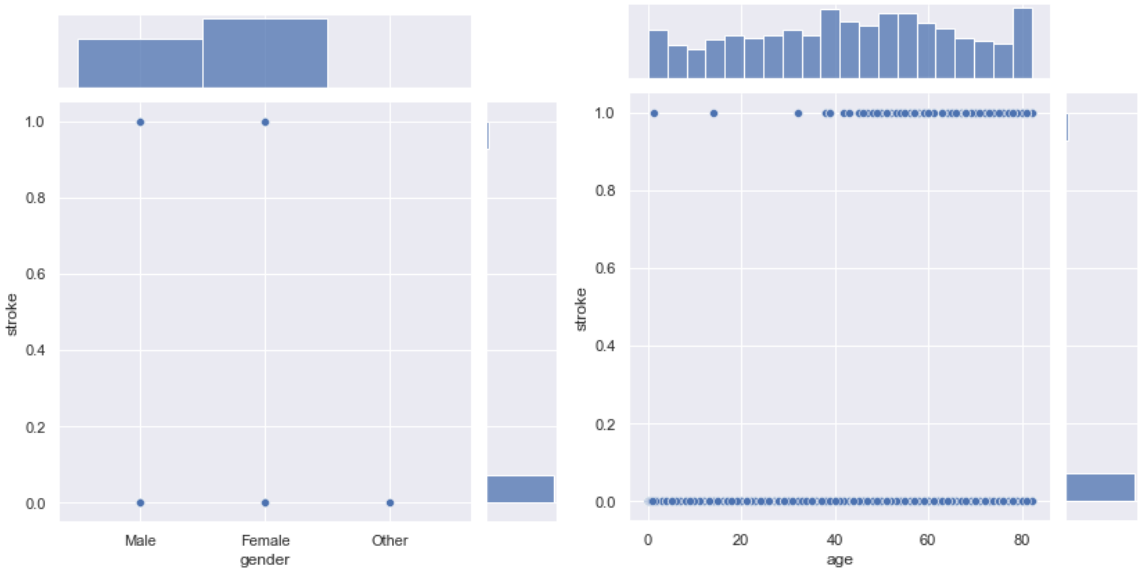
Ονομαστικοί (κατηγορικοί): gender , ever_married , work_type , Residence_level , smoking_status

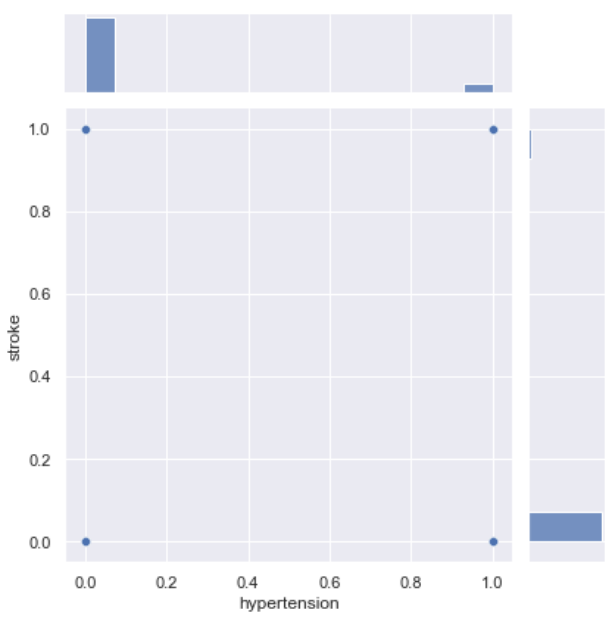
Σημείωση: Σαν missing values θεωρούμε τα N/A που υπάρχουν στο bmi και τα Unknown που υπάρχουν στο smoking_status .

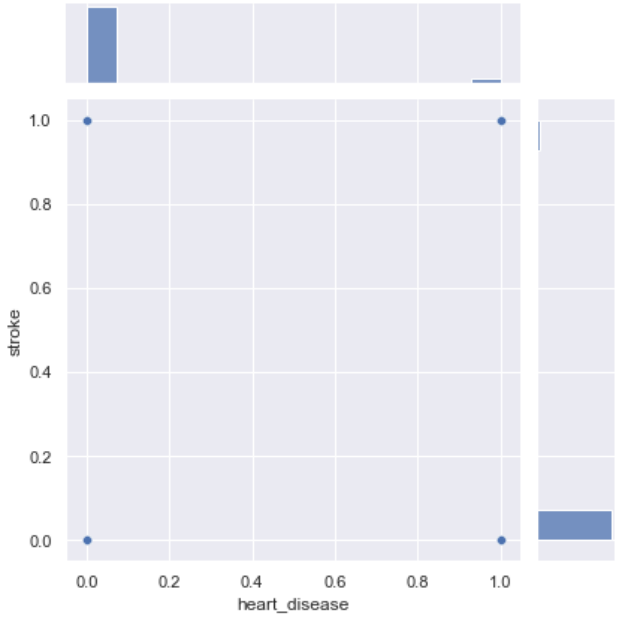
	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1
...	...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	NaN	never smoked	0
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.0	never smoked	0
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.6	never smoked	0
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.6	formerly smoked	0
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.2	Unknown	0

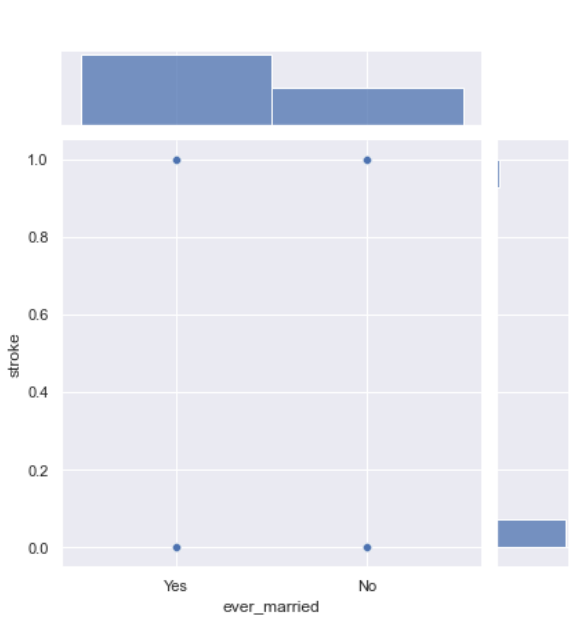
5110 rows × 12 columns

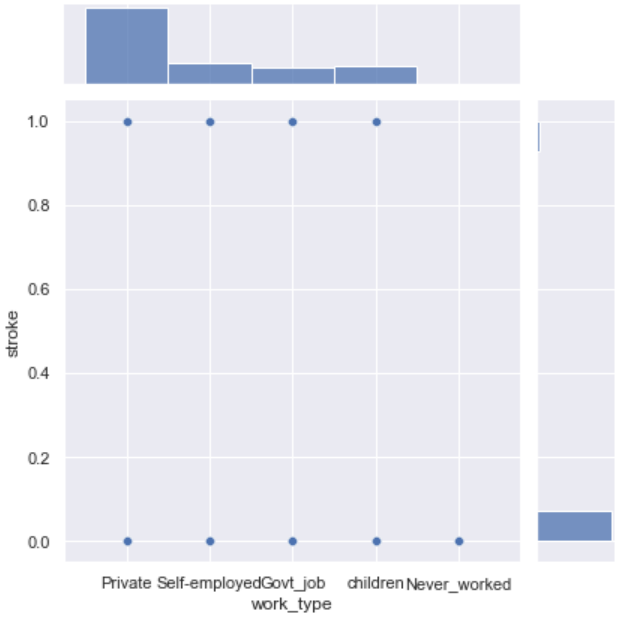
Γραφική αναπαράσταση

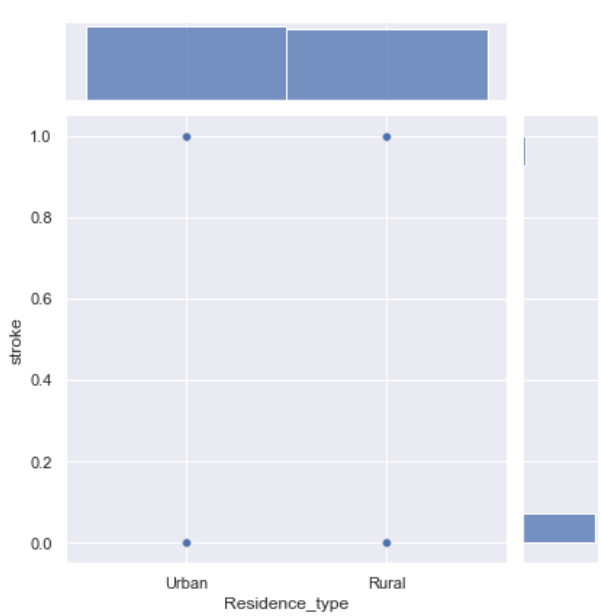


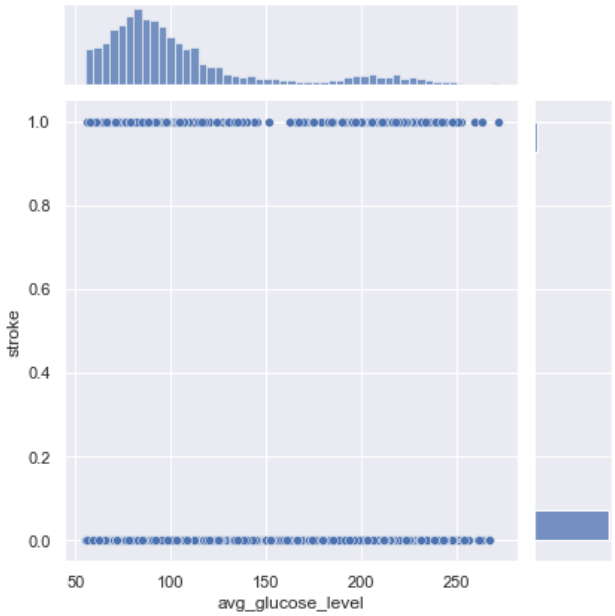

stroke
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
hypertension

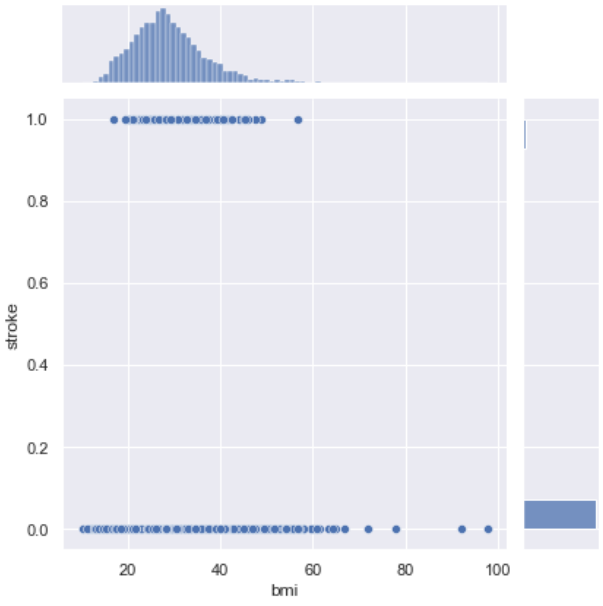

stroke
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
heart_disease

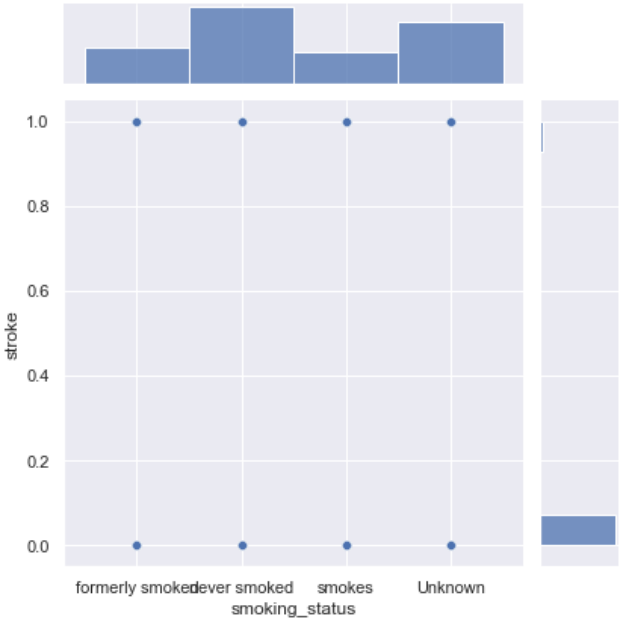

stroke
1.0
0.8
0.6
0.4
0.2
0.0
Yes
No
ever_married


stroke
1.0
0.8
0.6
0.4
0.2
0.0
Private
Self-employed
Govt_job
children
Never_worked
work_type


1.0
0.8
0.6
0.4
0.2
0.0
stroke
Urban
Rural
Residence_type


1.0
0.8
0.6
0.4
0.2
0.0
stroke
50
100
150
200
250
avg_glucose_level


1.0
0.8
0.6
0.4
0.2
0.0
stroke
20
40
60
80
100
bmi


1.0
0.8
0.6
0.4
0.2
0.0
stroke
formerly smoked
never smoked
smokes
Unknown
smoking_status

Αναλυτική καταγραφή του περιβάλλοντος υλοποίησης

Περιβάλλον υλοποίησης για την γραφική αναπαράσταση του dataset χρησιμοποιήθηκε το Jupyter Notebook και συγκεκριμένα οι βιβλιοθήκες “pandas” , “seaborn” .

Σχετικά με την εγκατάστασή του παρέχεται από το λογισμό Anaconda το οποίο είναι δωρεάν για ατομική χρήση και μπορεί εύκολα κανείς να το κατεβάσει από το διαδίκτυο.

Σύντομη περιγραφή της διαδικασίας υλοποίησης

Διάβασμα αρχείου csv, χρήση βιβλιοθήκης για γραφική αναπαράσταση δεδομένων.

Υποερώτημα Β.

Εντοπισμός πλήθους ελλιπών τιμών στο dataset

```
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                201
smoking_status    1544
stroke               0
dtype: int64
```

1. Αφαίρεση στήλης

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	1
...	...	...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	0
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	0
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	0
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	0
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	0

5110 rows × 10 columns

2. Συμπλήρωση ελλιπών τιμών με το μέσο όρο των στοιχείων στήλης

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.600000	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	28.893237	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.500000	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.400000	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.000000	1
...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	28.893237	0
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.000000	0
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.600000	0
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.600000	0
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.200000	0

5110 rows × 11 columns

3. Συμπλήρωση ελλιπών τιμών χρησιμοποιώντας Linear Regression

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.600000	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	30.017749	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.500000	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.400000	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.000000	1
...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	33.519375	0
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.000000	0
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.600000	0
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.600000	0
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.200000	0

5110 rows × 11 columns

4. Εφαρμογή k-NN για συμπλήρωση ελλιπών τιμών (k=10)

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	smoking_status	stroke
0	9046.0	Male	67.0	0.0	1.0	Yes	Private	Urban	228.69	formerly smoked	1.0
1	51676.0	Female	61.0	0.0	0.0	Yes	Self_employed	Rural	202.21	never smoked	1.0
2	31112.0	Male	80.0	0.0	1.0	Yes	Private	Rural	105.92	never smoked	1.0
3	60182.0	Female	49.0	0.0	0.0	Yes	Private	Urban	171.23	smokes	1.0
4	1665.0	Female	79.0	1.0	0.0	Yes	Self_employed	Rural	174.12	never smoked	1.0
...	...	...	...	...	...	...	...	...	...	...	...
5105	18234.0	Female	80.0	1.0	0.0	Yes	Private	Urban	83.75	never smoked	0.0
5106	44873.0	Female	81.0	0.0	0.0	Yes	Self_employed	Urban	125.20	never smoked	0.0
5107	19723.0	Female	35.0	0.0	0.0	Yes	Self_employed	Rural	82.99	never smoked	0.0
5108	37544.0	Male	51.0	0.0	0.0	Yes	Private	Rural	166.29	formerly smoked	0.0
5109	44679.0	Female	44.0	0.0	0.0	Yes	Govnt_job	Urban	85.28	never smoked	0.0

5110 rows × 11 columns

5. Συμπλήρωση ελλιπών τιμών αριθμητικών και κατηγορικών χρησιμοποιώντας Linear Regression και k-NN αντίστοιχα.

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.600000	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	30.017749	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.500000	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.400000	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.000000	never smoked	1
...	...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	33.519375	never smoked	0
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.000000	never smoked	0
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.600000	never smoked	0
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.600000	formerly smoked	0
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.200000	never smoked	0

5110 rows × 12 columns

Αναλυτική καταγραφή του περιβάλλοντος υλοποίησης

Περιβάλλον υλοποίησης για τον χειρισμό των ελλιπών τιμών του dataset χρησιμοποιήθηκε το Jupyter Notebook και συγκεκριμένα οι βιβλιοθήκες “pandas” , “numpy” και “ sklearn “ .

Σχετικά με την εγκατάστασή του παρέχεται από το λογισμό Anaconda το οποίο είναι δωρεάν για ατομική χρήση και μπορεί εύκολα κανείς να το κατεβάσει από το διαδίκτυο.

Σύντομη περιγραφή της διαδικασίας υλοποίησης

B.1:

Διάβασμα αρχείου csv , εντοπισμός ελλιπών τιμών (NaN και Unknown) και αφαίρεση στηλών “bmi” και “smoking_status” όπου αυτές βρίσκονται.

B.2:

Διάβασμα αρχείου csv, υπολογισμός μέσης τιμής των μη ελλιπών τιμών της στήλης “bmi” και τοποθέτηση της τιμής αυτής στην θέση των ελλιπών τιμών , αφαίρεση στήλης smoking_status.

B.3:

Διάβασμα αρχείου csv, αφαίρεση στήλης smoking_status και μετατροπή στοιχείων κατηγοριών που είναι string σε integers (gender, ever_married, work_type, Residence_type) , έχουμε λοιπόν το μητρώο stroke 5110x11.

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	stroke
0	9046	1	67.0	0	1	1	2	1	228.69	36.6	1
1	51676	0	61.0	0	0	1	3	0	202.21	NaN	1
2	31112	1	80.0	0	1	1	2	0	105.92	32.5	1
3	60182	0	49.0	0	0	1	2	1	171.23	34.4	1
4	1665	0	79.0	1	0	1	3	0	174.12	24.0	1
...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	0	80.0	1	0	1	2	1	83.75	NaN	0
5106	44873	0	81.0	0	0	1	3	1	125.20	40.0	0
5107	19723	0	35.0	0	0	1	3	0	82.99	30.6	0
5108	37544	1	51.0	0	0	1	2	0	166.29	25.6	0
5109	44679	0	44.0	0	0	1	0	1	85.28	26.2	0

5110 rows × 11 columns

Δημιουργία μητρώου test_data 201x11 (το θεωρούμε test data) το οποίο περιέχει τις γραμμές του stroke στις οποίες βρίσκονται οι αριθμητικές ελλιπείς τιμές.

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	stroke
1	51676	0	61.0	0	0	1	3	0	202.21	NaN	1
8	27419	0	59.0	0	0	1	2	0	76.15	NaN	1
13	8213	1	78.0	0	1	1	2	1	219.84	NaN	1
19	25226	1	57.0	0	1	0	0	1	217.08	NaN	1
27	61843	1	58.0	0	0	1	2	0	189.84	NaN	1
...	...	...	...	...	...	...	...	...	...	...	...
5039	42007	1	41.0	0	0	0	2	0	70.15	NaN	0
5048	28788	1	40.0	0	0	1	2	1	191.15	NaN	0
5093	32235	0	45.0	1	0	1	0	0	95.02	NaN	0
5099	7293	1	40.0	0	0	1	2	0	83.94	NaN	0
5105	18234	0	80.0	1	0	1	2	1	83.75	NaN	0

201 rows × 11 columns

Αφαίρεση από το stroke τα ορίσματα που περιέχονται στο test_data, οπότε τώρα έχουμε stroke 4909x11 (το θεωρούμε train data).

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	stroke
0	9046	1	67.0	0	1	1	2	1	228.69	36.6	1
2	31112	1	80.0	0	1	1	2	0	105.92	32.5	1
3	60182	0	49.0	0	0	1	2	1	171.23	34.4	1
4	1665	0	79.0	1	0	1	3	0	174.12	24.0	1
5	56669	1	81.0	0	0	1	2	1	186.21	29.0	1
...	...	...	...	...	...	...	...	...	...	...	...
5104	14180	0	13.0	0	0	0	4	0	103.08	18.6	0
5106	44873	0	81.0	0	0	1	3	1	125.20	40.0	0
5107	19723	0	35.0	0	0	1	3	0	82.99	30.6	0
5108	37544	1	51.0	0	0	1	2	0	166.29	25.6	0
5109	44679	0	44.0	0	0	1	0	1	85.28	26.2	0

4909 rows × 11 columns

Δημιουργούμε τα x_train και y_train από το stroke . Το y_train 4909x1 περιέχει την στήλη “bmi” ενώ το x_train 4909x10 όλες τις στήλες του stroke εκτός της “bmi”.

```
0       36.6
2       32.5
3       34.4
4       24.0
5       29.0
        ...
5104    18.6
5106    40.0
5107    30.6
5108    25.6
5109    26.2
Name: bmi, Length: 4909, dtype: float64
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	stroke
0	9046	1	67.0	0	1	1	2	1	228.69	1
2	31112	1	80.0	0	1	1	2	0	105.92	1
3	60182	0	49.0	0	0	1	2	1	171.23	1
4	1665	0	79.0	1	0	1	3	0	174.12	1
5	56669	1	81.0	0	0	1	2	1	186.21	1
...	...	...	...	...	...	...	...	...	...	...
5104	14180	0	13.0	0	0	0	4	0	103.08	0
5106	44873	0	81.0	0	0	1	3	1	125.20	0
5107	19723	0	35.0	0	0	1	3	0	82.99	0
5108	37544	1	51.0	0	0	1	2	0	166.29	0
5109	44679	0	44.0	0	0	1	0	1	85.28	0

4909 rows × 10 columns

Εκπαιδεύουμε το LinearRegression μοντέλο στα x_train και y_train.

Δημιουργούμε το X_test 201x10 από το test_data κρατώντας όλες τις στήλες εκτός από την “bmi”.

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	stroke
1	51676	0	61.0	0	0	1	3	0	202.21	1
8	27419	0	59.0	0	0	1	2	0	76.15	1
13	8213	1	78.0	0	1	1	2	1	219.84	1
19	25226	1	57.0	0	1	0	0	1	217.08	1
27	61843	1	58.0	0	0	1	2	0	189.84	1
...	...	...	...	...	...	...	...	...	...	...
5039	42007	1	41.0	0	0	0	2	0	70.15	0
5048	28788	1	40.0	0	0	1	2	1	191.15	0
5093	32235	0	45.0	1	0	1	0	0	95.02	0
5099	7293	1	40.0	0	0	1	2	0	83.94	0
5105	18234	0	80.0	1	0	1	2	1	83.75	0

201 rows × 10 columns

Εφαρμόζουμε το LinearRegression εκπαιδευμένο μοντέλο στο X_test και βρίσκουμε την πρόβλεψη y_pred 201x1 (των ελλιπών τιμών).

0	30.017749
1	28.988004
2	31.065805
3	30.099370
4	30.878113
...	...
196	26.327459
197	31.442187
198	35.182890
199	29.483602
200	33.519375

201 rows × 1 columns

Αντικαθιστούμε τις ελλιπείς τιμές (βρίσκονται στο test_data στην στήλη “bmi”) με τις προβλεπόμενες που βρίσκονται στο y_pred.

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	stroke
1	51676	0	61.0	0	0	1	3	0	202.21	30.017749	1
8	27419	0	59.0	0	0	1	2	0	76.15	28.988004	1
13	8213	1	78.0	0	1	1	2	1	219.84	31.065805	1
19	25226	1	57.0	0	1	0	0	1	217.08	30.099370	1
27	61843	1	58.0	0	0	1	2	0	189.84	30.878113	1
...	...	...	...	...	...	...	...	...	...	...	...
5039	42007	1	41.0	0	0	0	2	0	70.15	26.327459	0
5048	28788	1	40.0	0	0	1	2	1	191.15	31.442187	0
5093	32235	0	45.0	1	0	1	0	0	95.02	35.182890	0
5099	7293	1	40.0	0	0	1	2	0	83.94	29.483602	0
5105	18234	0	80.0	1	0	1	2	1	83.75	33.519375	0

201 rows × 11 columns

Βάζουμε τις προβλεπόμενες τιμές στο αρχικό stroke 5110x11

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	stroke
0	9046	1	67.0	0	1	1	2	1	228.69	36.600000	1
1	51676	0	61.0	0	0	1	3	0	202.21	30.017749	1
2	31112	1	80.0	0	1	1	2	0	105.92	32.500000	1
3	60182	0	49.0	0	0	1	2	1	171.23	34.400000	1
4	1665	0	79.0	1	0	1	3	0	174.12	24.000000	1
...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	0	80.0	1	0	1	2	1	83.75	33.519375	0
5106	44873	0	81.0	0	0	1	3	1	125.20	40.000000	0
5107	19723	0	35.0	0	0	1	3	0	82.99	30.600000	0
5108	37544	1	51.0	0	0	1	2	0	166.29	25.600000	0
5109	44679	0	44.0	0	0	1	0	1	85.28	26.200000	0

5110 rows × 11 columns

B.4:

Διάβασμα αρχείου csv, αφαίρεση στήλης bmi, μετατροπή των “Unknown” του smoking_status σε “NaN” και μετατροπή στοιχείων κατηγοριών που είναι string σε intigers (gender, ever_married, work_type, Residence_type, smoking_status)) , έχουμε λοιπόν το μητρώο stroke 5110x11.

Δημιουργία μητρώου test_data 1544x11 (το θεωρούμε test data) το οποίο περιέχει τις γραμμές του stroke στις οποίες βρίσκονται οι ελλιπείς τιμές.

Αφαίρεση από το stroke τα ορίσματα που περιέχονται στο test_data, οπότε τώρα έχουμε stroke 3566x11 (το θεωρούμε train data).

Δημιουργούμε τα x_train και y_train από το stroke . Το y_train 3566x1 περιέχει την στήλη “smoking_status” ενώ το x_train 3566x10 όλες τις στήλες του stroke εκτός της “smoking_status”.

Δημιουργούμε το KNN Classifier(k=10) μοντέλο και το εκπαιδεύουμε χρησιμοποιώντας τα x_train και y_train.

Δημιουργούμε το X_test 1544x10 από το test_data κρατώντας όλες τις στήλες εκτός από την “smoking_status”.

Εφαρμόζουμε το εκπαιδευμένο μοντέλο στο X_test και βρίσκουμε την πρόβλεψη y_pred 1544x1 (των ελλιπών τιμών).

Αντικαθιστούμε τις ελλιπείς τιμές (βρίσκονται στο test_data στην στήλη “smoking_status”) με τις προβλεπόμενες που βρίσκονται στο y_pred.

Μετατρέπουμε τις αριθμητικές τιμές του smoking_status στις αντίστοιχες αρχικές (formerly smoked, never smoked, smokes).

Βάζουμε τις προβλεπόμενες τιμές στο αρχικό stroke 5110x11.

Σημείωση: Τα βήματα που εκτελέστηκαν στο Β.4 είναι παρόμοια με αυτά του Β.3, για αυτό το λόγο η εισαγωγή φωτογραφιών παραλήφθηκε.

Β.5:

Εκτελούμε τα βήματα του Β.3 , αποθηκεύουμε την στήλη bmi , εκτελούμε τα βήματα του Β.4 , αποθηκεύουμε την στήλη smoking_status , φορτώνουμε το csv αρχείο , αντικαθιστούμε τις στήλες bmi ,smoking_status με αυτές που έχουμε κρατήσει από τα Β.3 και Β.4 αντίστοιχα.

Υποερώτημα Γ

Για κάθε μητρώο του ερωτήματος Β εφαρμόστηκε πρόβλεψη της κατηγορίας stroke και παρατίθεται το classification report , το feature importance , και η συσχέτιση με την κατηγορία stroke.

1.

	precision	recall	f1-score	support
0	0.95	0.99	0.97	1211
1	0.23	0.04	0.07	67
accuracy			0.94	1278
macro avg	0.59	0.52	0.52	1278
weighted avg	0.91	0.94	0.92	1278

```
avg_glucose_level    0.505386
age                  0.324111
work_type            0.052772
Residence_type       0.027627
hypertension         0.025956
gender               0.024988
heart_disease        0.021064
ever_married         0.018096
dtype: float64
```

```
stroke               1.000000
age                  0.245257
heart_disease        0.134914
avg_glucose_level    0.131945
hypertension         0.127904
ever_married         0.108340
Residence_type       0.015458
gender               0.008929
id                   0.006388
work_type           -0.032316
Name: stroke, dtype: float64
```

2.

	precision	recall	f1-score	support
0	0.95	0.99	0.97	1211
1	0.12	0.01	0.03	67
accuracy			0.94	1278
macro avg	0.54	0.50	0.50	1278
weighted avg	0.90	0.94	0.92	1278

```
avg_glucose_level    0.307469
bmi                  0.271912
age                  0.245156
work_type            0.054546
gender               0.029465
hypertension         0.025653
Residence_type       0.025444
heart_disease        0.021997
ever_married         0.018360
dtype: float64
```

```
stroke               1.000000
age                  0.245257
heart_disease        0.134914
avg_glucose_level    0.131945
hypertension         0.127904
ever_married         0.108340
bmi                  0.038947
Residence_type       0.015458
gender               0.008929
id                   0.006388
work_type           -0.032316
Name: stroke, dtype: float64
```

3.

	precision	recall	f1-score	support
0	0.95	1.00	0.97	1211
1	0.00	0.00	0.00	67
accuracy			0.95	1278
macro avg	0.47	0.50	0.49	1278
weighted avg	0.90	0.95	0.92	1278

```
avg_glucose_level    0.320522
bmi                  0.262490
age                  0.235369
work_type            0.060968
gender               0.032887
Residence_type       0.024845
hypertension         0.023724
heart_disease        0.021562
ever_married         0.017633
dtype: float64
```

```
stroke               1.000000
age                  0.245257
heart_disease        0.134914
avg_glucose_level    0.131945
hypertension         0.127904
ever_married         0.108340
bmi                  0.041102
Residence_type       0.015458
gender               0.008929
id                   0.006388
work_type           -0.032316
Name: stroke, dtype: float64
```

4.

	precision	recall	f1-score	support
0	0.95	1.00	0.97	1211
1	0.33	0.03	0.05	67
accuracy			0.95	1278
macro avg	0.64	0.51	0.51	1278
weighted avg	0.92	0.95	0.92	1278

```
avg_glucose_level    0.430804
age                  0.290480
work_type            0.066726
smoking_status       0.059523
gender               0.046662
hypertension         0.030288
Residence_type       0.029543
heart_disease        0.023445
ever_married         0.022527
dtype: float64
```

```
stroke               1.000000
age                  0.245257
heart_disease        0.134914
avg_glucose_level    0.131945
hypertension         0.127904
ever_married         0.108340
Residence_type       0.015458
gender               0.008929
id                   0.006388
smoking_status      -0.032110
work_type           -0.032316
Name: stroke, dtype: float64
```

5.

	precision	recall	f1-score	support
0	0.95	1.00	0.97	1211
1	1.00	0.01	0.03	67
accuracy			0.95	1278
macro avg	0.97	0.51	0.50	1278
weighted avg	0.95	0.95	0.92	1278

```
avg_glucose_level    0.307197
bmi                  0.238823
age                  0.215722
smoking_status       0.056841
work_type            0.051873
gender               0.036988
Residence_type       0.026035
heart_disease        0.023944
hypertension         0.023482
ever_married         0.019095
dtype: float64
```

```
stroke               1.000000
age                  0.245257
heart_disease        0.134914
avg_glucose_level    0.131945
hypertension         0.127904
ever_married         0.108340
bmi                  0.041102
Residence_type       0.015458
gender               0.008929
id                   0.006388
smoking_status      -0.032110
work_type           -0.032316
Name: stroke, dtype: float64
```

Αναλυτική καταγραφή του περιβάλλοντος υλοποίησης

Περιβάλλον υλοποίησης για τον χειρισμό των ελλιπών τιμών του dataset χρησιμοποιήθηκε το Jupyter Notebook και συγκεκριμένα οι βιβλιοθήκες “pandas” , “numpy” και “ sklearn “ .

Σχετικά με την εγκατάστασή του παρέχεται από το λογισμό Anaconda το οποίο είναι δωρεάν για ατομική χρήση και μπορεί εύκολα κανείς να το κατεβάσει από το διαδίκτυο.

Σύντομη περιγραφή της διαδικασίας υλοποίησης

<u>Πριν την προσπάθεια βελτιστοποίησης</u>:

Ορίζουμε το εξαρτημένο μητρώο Y το οποίο περιέχει τις τιμές της κατηγορίας stroke. Y (5110x1)

Ορίζουμε το ανεξάρτητο μητρώο X το οποίο περιέχει το dataset stroke εκτός της εξαρτημένης κατηγορίας stroke και της ασυσχέτιστης κατηγορίας id. X (5110x8)

Χωρίζουμε το dataset σε train και test με αναλογία 75%-25% αντίστοιχα.

X_train (3832x8) X_test (1278x8) Y_train(3832x1) Y_test (1278x1)

Δημιουργούμε το μοντέλο με Random Forest Classifier (n_estimators=10) και το εκπαιδεύουμε με τα X_train και Y_train.

Χρησιμοποιούμε το εκπαιδευμένο μοντέλο πάνω στο X_test και παίρνουμε το μητρώο prediction_test. prediction_test (1278x1)

Η αποτελεσματικότητα της κατηγοριοποίησης φαίνεται μέσα από τη σύγκριση των prediction_test και Y_test. Κύριο μέσο σύγκρισης χρησιμοποιήθηκε το classification_report από την βιβλιοθήκη sklearn.metrics , επίσης , για διευκόλυνση της βελτίωσης του κατηγοριοποιητή που θα υπάρξει στην συνέχεια , μετά από την χρήση του Random Forest Classifier παραθέτουμε το feature_imp που αντιπροσωπεύει κατά πόσο έχουν συνεισφέρει οι κατηγορίες του dataset στην πρόβλεψη του stroke. Για κάθε ένα από τα 5 μητρώα του ερωτήματος Β υπάρχουν φωτογραφίες του classification_report.

<u>Για την προσπάθεια βελτιστοποίησης</u>:

Θα πρέπει να διαχειριστούμε τα outliers για την καλύτερη απόδοση του Random Forest Classifier. Συγκεκριμένα τα αντικαταστήσαμε με την μέση τιμή της στήλης.

Μπορούμε να βγάλουμε τις κατηγορίες που έχουν χαμηλή συσχέτιση με το stroke.

Κανονικοποίηση τιμών, που προέρχονται από αριθμητικά δεδομένα, στο διάστημα [0-1].

Ο KNN Classifier έγινε με k=20.

Αποτελέσματα:

1.

	precision	recall	f1-score	support
0	0.95	0.99	0.97	1211
1	0.29	0.09	0.14	67
accuracy			0.94	1278
macro avg	0.62	0.54	0.55	1278
weighted avg	0.92	0.94	0.93	1278

```
avg_glucose_level    0.661542
age                  0.276234
heart_disease        0.022992
ever_married         0.020023
hypertension         0.019209
dtype: float64
```

```
stroke               1.000000
age                  0.194215
heart_disease        0.134914
hypertension         0.127904
ever_married         0.108340
avg_glucose_level    0.102588
Residence_type       0.015458
gender               0.008929
id                   0.006388
work_type           -0.032316
Name: stroke, dtype: float64
```

2.

	precision	recall	f1-score	support
0	0.95	1.00	0.97	1211
1	0.44	0.06	0.11	67
accuracy			0.95	1278
macro avg	0.70	0.53	0.54	1278
weighted avg	0.92	0.95	0.93	1278

avg_glucose_level	0.366561
bmi	0.298022
age	0.234726
Residence_type	0.038074
heart_disease	0.025676
hypertension	0.019575
ever_married	0.017366

dtype: float64

stroke	1.000000
age	0.194215
heart_disease	0.134914
hypertension	0.127904
ever_married	0.108340
avg_glucose_level	0.102588
bmi	0.038695
Residence_type	0.015458
gender	0.008929
id	0.006388
work_type	-0.032316

Name: stroke, dtype: float64

3.

	precision	recall	f1-score	support
0	0.95	1.00	0.97	1211
1	0.50	0.04	0.08	67
accuracy			0.95	1278
macro avg	0.72	0.52	0.53	1278
weighted avg	0.93	0.95	0.93	1278

avg_glucose_level	0.369078
bmi	0.348595
age	0.225242
hypertension	0.022094
ever_married	0.018980
heart_disease	0.016011

dtype: float64

```
stroke               1.000000
age                  0.193246
heart_disease        0.134914
hypertension         0.127904
ever_married         0.108340
avg_glucose_level    0.091827
bmi                  0.044685
Residence_type       0.015458
gender               0.008929
id                   0.006388
work_type           -0.032316
Name: stroke, dtype: float64
```

4.

	precision	recall	f1-score	support
0	0.95	0.99	0.97	1211
1	0.22	0.07	0.11	67
accuracy			0.94	1278
macro avg	0.58	0.53	0.54	1278
weighted avg	0.91	0.94	0.92	1278

```
avg_glucose_level    0.539981
age                  0.294782
smoking_status       0.054641
hypertension         0.031737
Residence_type       0.030670
heart_disease        0.028233
ever_married         0.019956
dtype: float64
```

```
stroke               1.000000
age                  0.193246
heart_disease        0.134914
hypertension         0.127904
ever_married         0.108340
avg_glucose_level    0.091827
Residence_type       0.015458
gender               0.008929
id                   0.006388
smoking_status      -0.029264
work_type           -0.032316
Name: stroke, dtype: float64
```

5.

	precision	recall	f1-score	support
0	0.95	1.00	0.97	1211
1	0.43	0.04	0.08	67
accuracy			0.95	1278
macro avg	0.69	0.52	0.53	1278
weighted avg	0.92	0.95	0.93	1278

```
bmi                 0.328757
avg_glucose_level   0.315274
age                 0.225297
smoking_status      0.052137
hypertension        0.028922
heart_disease       0.026740
ever_married        0.022873
dtype: float64
```

```
stroke              1.000000
age                 0.193246
heart_disease       0.134914
hypertension        0.127904
ever_married        0.108340
avg_glucose_level   0.091827
bmi                 0.044685
Residence_type      0.015458
gender              0.008929
id                  0.006388
smoking_status     -0.029264
work_type          -0.032316
Name: stroke, dtype: float64
```

Σχολιασμός των τελικών αποτελεσμάτων

Πριν την προσπάθεια βελτίωσης: Παρατηρούμε πως με την χρήση Random Forest τα καλύτερα αποτελέσματα έδωσε το μητρώο B5 στο οποίο χρησιμοποιήσαμε Linear Regression για τις ελλιπείς τιμές του bmi και K-NN για τις ελλιπείς τιμές του smoking status.

Μετά την προσπάθεια βελτίωσης: Παρατηρούμε πως τα καλύτερα αποτελέσματα έδωσε και πάλι το μητρώο B5 , τα αποτελέσματα από τα μητρώο B1-B3 βελτιώθηκαν , το μητρώο B4 δεν έδειξε βελτίωση ένα από τους λόγους μπορεί να είναι ότι αυξήθηκε το k από 10 σε 20 . Παραθέτουμε παρακάτω λίγα λόγια σχετικά με αυτό.

Curse of Dimensionality: Ο k-NN έχει καλύτερο performance με μικρό αριθμό από features παρά με μεγάλο. Η αύξηση της διάστασης του αλγορίθμου οδηγεί σε overfitting , για να αποφευχθεί αυτό, τα δεδομένα μας θα πρέπει να μεγαλώνουν εκθετικά καθώς αυξάνουμε τον αριθμό των διαστάσεων . Το πρόβλημα αυτό, των μεγαλύτερων διαστάσεων ,είναι γνωστό ως Curse of Dimensionality και για να το διαχειριστούμε πρέπει να εφαρμόσουμε PCA προτού εφαρμόσουμε οποιοδήποτε αλγόριθμο μάθησης ή μπορούμε να χρησιμοποιήσουμε κάποια feature selection προσέγγιση. Επιπρόσθετα , έρευνες έχουν

δείξει πως σε μεγάλες διαστάσεις η ευκλείδεια απόσταση σταματά να είναι χρήσιμη , έτσι μπορεί κανείς να προτιμήσει άλλες μητρικές όπως cosine similarity η οποία επηρεάζεται λιγότερο από τις μεγάλες διαστάσεις.

Εμπειρικά παρατηρήθηκε κατά την διάρκεια υλοποίησης του project πως ένας μικρός αριθμός από γείτονες (μικρό k) αποτελεί flexible fit και θα έχει μικρό bias και μεγάλο variance . Αντίστοιχα ένας μεγάλος αριθμός από γείτονες θα έχει ομαλό όριο απόφασης το οποίο συνεπάγει μικρό variance και μεγάλο bias.

Feature importance: Έχει υλοποιηθεί για τα ερωτήματα του project , βλέποντας τον πίνακα ήταν εύκολο να αποφασίσουμε ποια χαρακτηριστικά να μην χρησιμοποιήσουμε με βάση το πόσο συνεισφέρουν στην διαδικασία της πρόβλεψης . Αυτό κρίθηκε αναγκαίο διότι όσο περισσότερα features έχουμε τόσο πιο πιθανό είναι το μοντέλο μας εμφανίσει overfitting.

Ερώτημα 2

Αναλυτική καταγραφή του περιβάλλοντος υλοποίησης

Περιβάλλον χρησιμοποιήθηκε το Jupyter Notebook και συγκεκριμένα οι βιβλιοθήκες “pandas” , “numpy”, “matplotlib” ,“ sklearn “, ”tensorflow.keras” και ”warnings” .

Σύντομη περιγραφή της διαδικασίας υλοποίησης

Προεπεξεργασία δεδομένων: Υπάρχει μια ελλιπή τιμή την οποία την διαχειριζόμαστε με αφαίρεση γραμμής.

Word Embedding: Τα word embeddings υπολογίζονται χρησιμοποιώντας δύο τεχνικές, supervised learning και self-supervised learning (Word2Vec, GloVe). Στο project αυτό επιλέχθηκε η πρώτη τεχνική , supervised learning και τα word embeddings υπολογίζονται κατά την διάρκεια του fitting του νευρωνικού δικτύου .

Νευρωνικό δίκτυο: Επιλέχθηκε Long Short-Term Memory network (LSTM) και υλοποιήθηκε με keras.

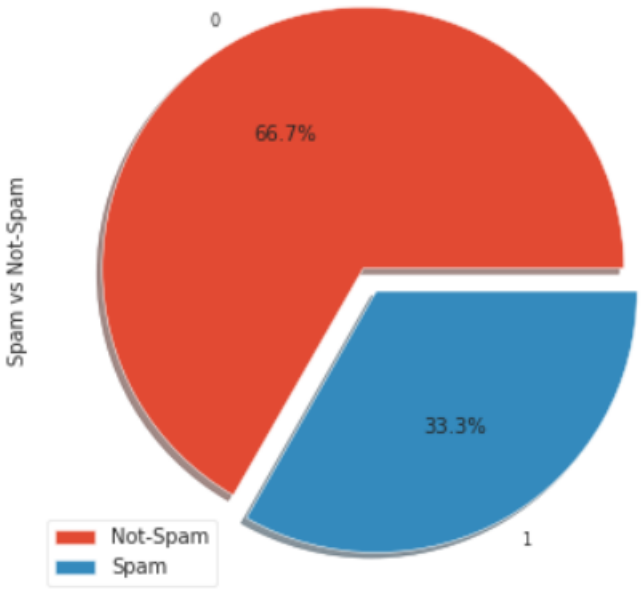
Διαδικασία Εκτέλεσης: Χωρίζουμε το dataset σε train και test με αναλογία 75%-25% ορίζοντας τις μεταβλητές X_train , y_train, X_test , y_test όπου με X συμβολίζουμε την στήλη email και με y την στήλη label .

Χρησιμοποιούμε την μέθοδο fit on texts του Tokenizer από keras preprocessing text και βρίσκουμε το λεξικό των training email . Έπειτα χρησιμοποιούμε την μέθοδο texts to sequences για τα email του train και test η οποία μετατρέπει κάθε email σε μια ακολουθία ακεραίων. Επειδή τα μήκοι είναι διαφορετικά αν email εφαρμόζουμε padding χρησιμοποιώντας τη συνάρτηση pad_sequences.

Φτιάχνουμε το μοντέλο LSTM το οποίο είναι ακολουθιακό (διαδοχικά layers) . Το πρώτο layer είναι το embedding που χρησιμοποιείται για να παρέχει πυκνή αναπαράσταση από λέξεις , τα επόμενα 2 layers είναι LSTM , το τελευταίο layer (Dense) αποτελείται από έναν νευρώνα με σιγμοειδή συνάρτηση ενεργοποίησης. Αφότου το κάνουμε compile , το εκπαιδεύουμε στα train data και το αξιολογούμε στα test data.

Σχολιασμός των τελικών αποτελεσμάτων :

Ποσοστά από το αρχικό dataset:



Αξιολόγηση μοντέλου χρησιμοποιώντας τις μητρικές f1-score,precision και recall:

```
Precision: 0.935714
Recall: 1.000000
F1 score: 0.966790
```

Precision = 0.93

Για ένα συγκεκριμένο email που θέλουμε το μοντέλο μας να κατηγοριοποιήσει αν το κατηγοριοποιήσει ως spam η πιθανότητα να είναι στην πραγματικότητα spam 93% .

Ένα ποσοστό 7% των email που έχουν κατηγορικοποιηθεί ως spam στην πραγματικότητα δεν είναι spam. Αυτό είναι ένα αρκετά μεγάλο ποσοστό σημαντικών μηνυμάτων που χάνονται.

Recall = 1

Από όλα τα spam μηνύματα του testing-dataset το μοντέλο τα κατηγοριοποίησε όλα ως spam.

F1-score = 0.97

Το f1-score μετράει το tradeoff μεταξύ precision και recall . Όσο πιο κοντά στο 1 βρίσκεται τόσο καλύτερα precision και recall έχει το μοντέλο μας.

Δύο παραδείγματα κατηγοριοποίησης :

```
#Test-Estimation

sms_test = ['Hi Paul, would you come around tonight']
sms_seq = tokenizer.texts_to_sequences(sms_test)

sms_pad = pad_sequences(sms_seq, maxlen=max_length, padding='post')
tokenizer.index_word
sms_pad
lstm_model.predict_classes(sms_pad)
#classified the text as no spam. Correct!
```

```
array([[0]])
```

```
#Test-Estimation

sms_test = ['Free SMS service for anyone']
sms_seq = tokenizer.texts_to_sequences(sms_test)

sms_pad = pad_sequences(sms_seq, maxlen=max_length, padding='post')
tokenizer.index_word
sms_pad
lstm_model.predict_classes(sms_pad)
#classified the tet as spam. Correct again!
```

```
array([[1]])
```

Περεταίρω ανάλυση :

```
Train: 0.999, Test: 0.976
```