Πανεπιστήμιο Πατρών

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Εξόρυξη Δεδομένων και Αλγόριθμοι Μάθησης

Υπεύθυνοι Καθηγητές: Μακρής Χρήστος
Μεγαλοοικονόμου Βασίλειος

Φοιτητής: Κουτσοχέρας Ιωάννης 1059638

Σεπτέμβριος 2021

Ερώτημα 1:

Γενική Περιγραφή: Στο συγκεκριμένο ερώτημα κλήθηκα να προβλέψω την πιθανότητα εμφάνισης εγκεφαλικού σε διάφορους ανθρώπους χρησιμοποιώντας δεδομένα που σχετίζονται με την κατάσταση τους (είτε είναι επαγγελματική, είτε υγειονομική, είτε κάποια άλλη). Περισσότερες λεπτομέρειες ανά ερώτημα παρουσιάζονται παρακάτω.

Υποερώτημα 1:

Σε αυτο το ερώτημα δημιούργησα διάφορα γραφήματα τα οποία περιγράφουν την κατάσταση των δεδομένων μου. Επιπλέον γίνεται μια σύντομη περιγραφή του είδους των στηλών. Παραθέτω τα παρακάτω screenshots μετά την εκτέλεση του κώδικα καθώς και τα γραφήματα.

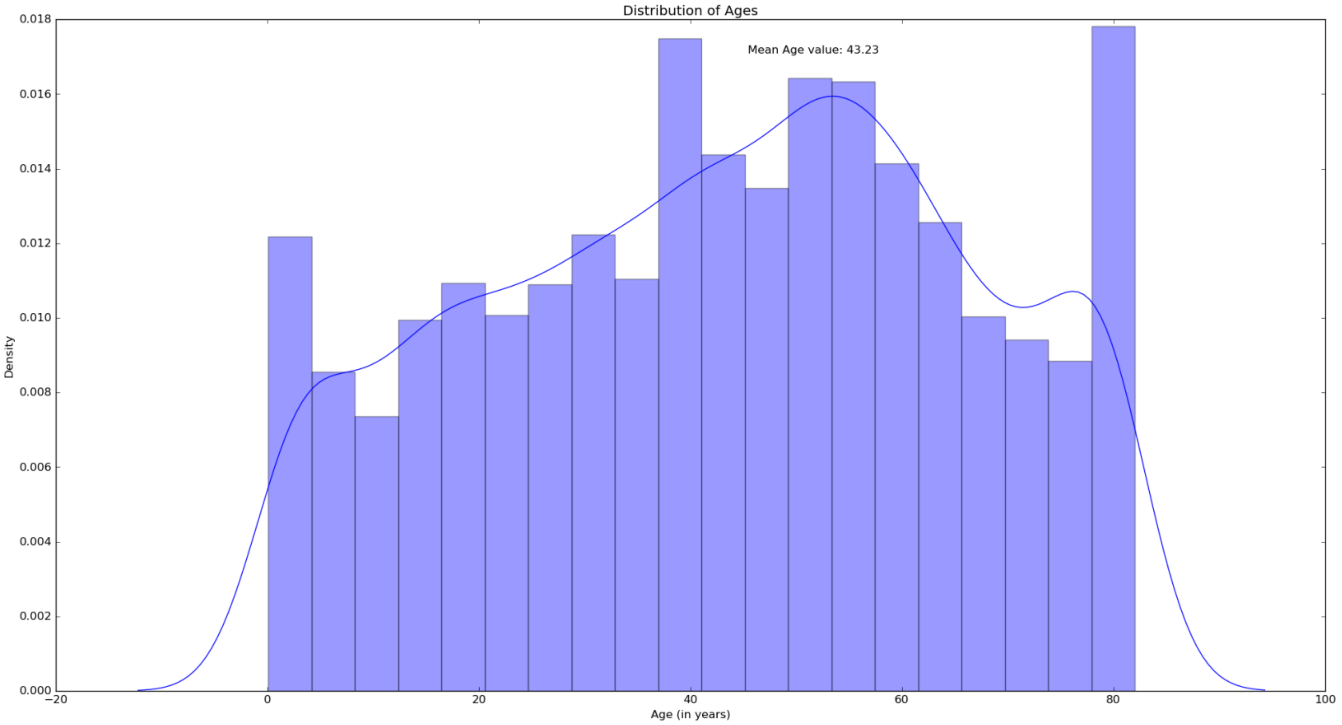
Από το αρχείο **exercise1/question_a.py**

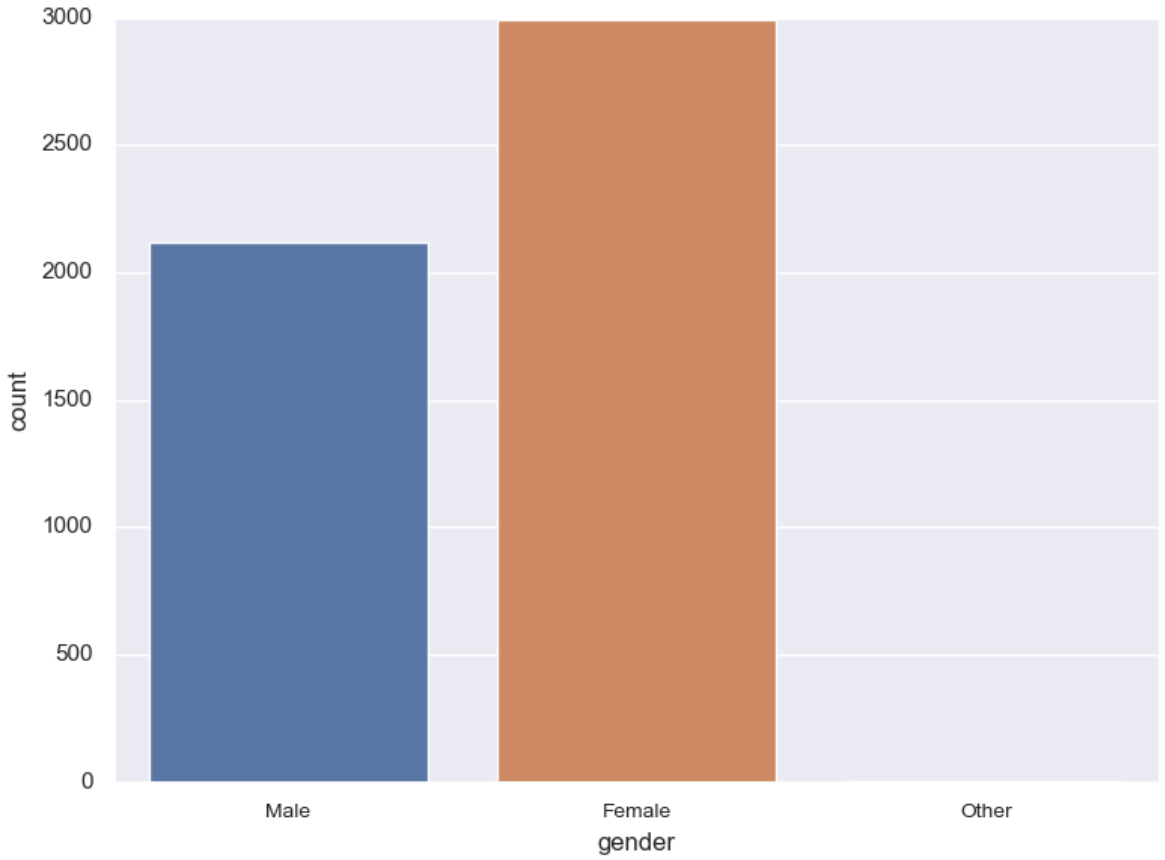
df.describe()

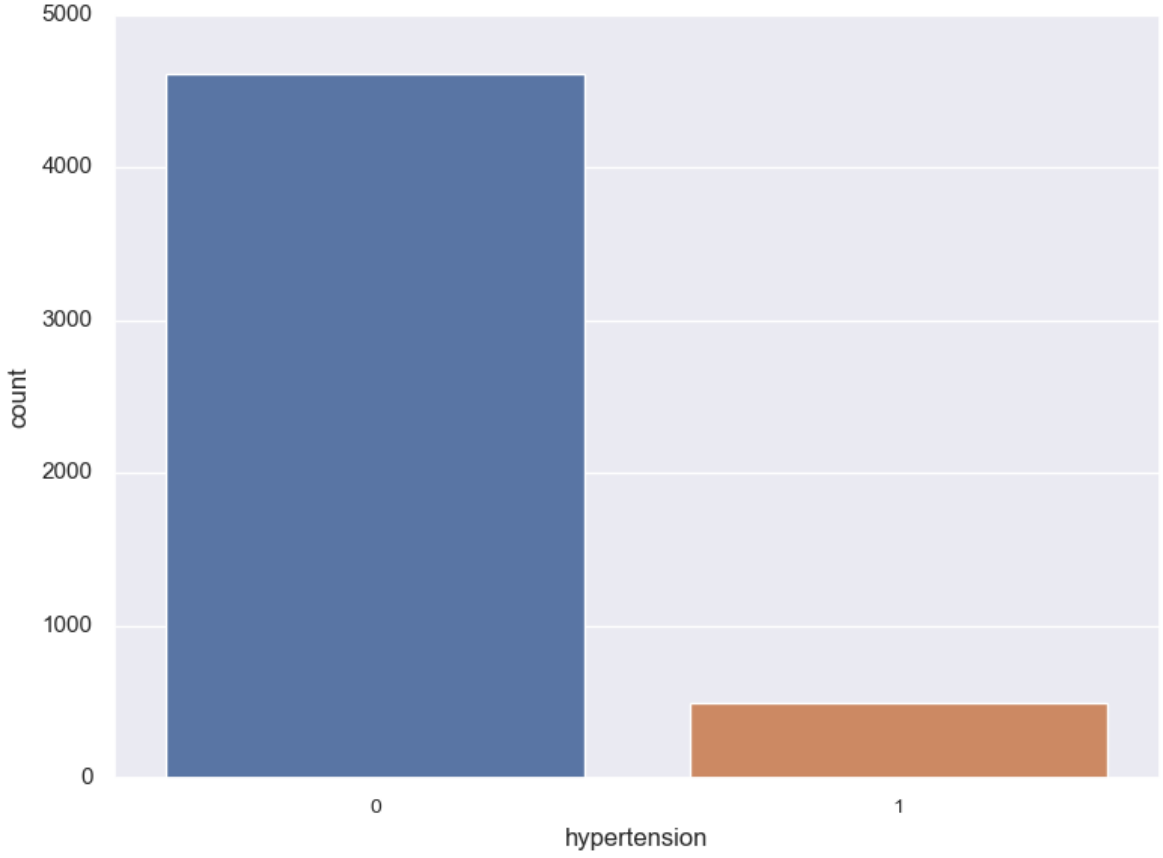
	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	4909.000000	5110.000000
mean	36517.829354	43.226614	0.097456	0.054012	106.147677	28.893237	0.048728
std	21161.721625	22.612647	0.296607	0.226063	45.283560	7.854067	0.215320
min	67.000000	0.080000	0.000000	0.000000	55.120000	10.300000	0.000000
25%	17741.250000	25.000000	0.000000	0.000000	77.245000	23.500000	0.000000
50%	36932.000000	45.000000	0.000000	0.000000	91.885000	28.100000	0.000000
75%	54682.000000	61.000000	0.000000	0.000000	114.090000	33.100000	0.000000
max	72940.000000	82.000000	1.000000	1.000000	271.740000	97.600000	1.000000

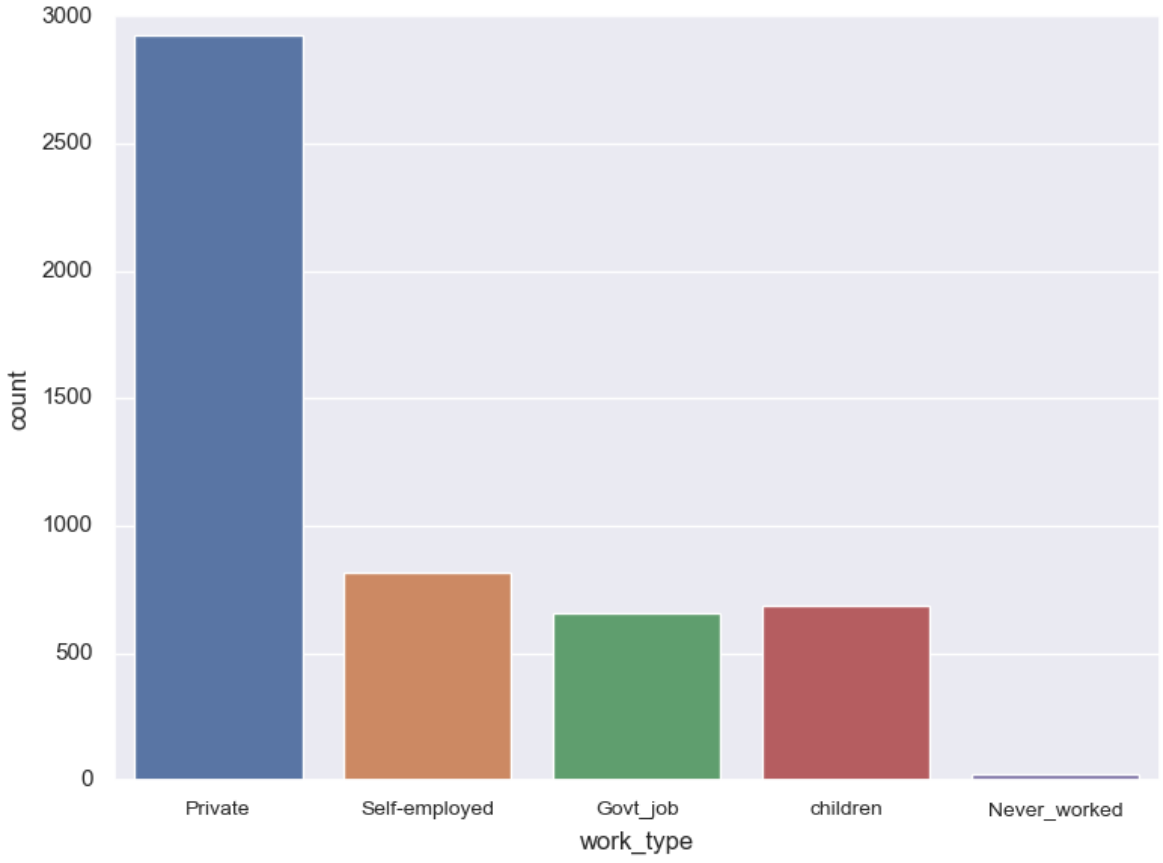
df.isna().sum()

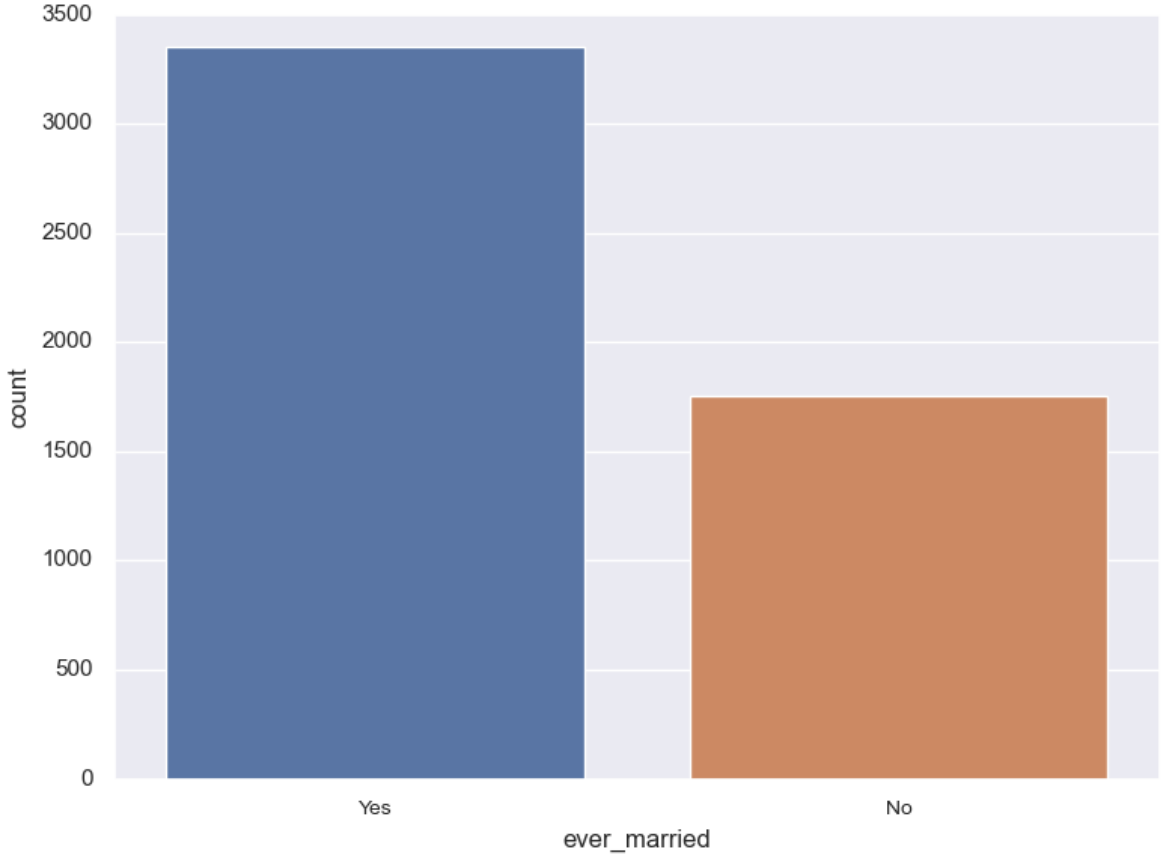
id	0
gender	0
age	0
hypertension	0
heart_disease	0
ever_married	0
work_type	0
Residence_type	0
avg_glucose_level	0
bmi	201
smoking_status	0
stroke	0
dtype: int64	

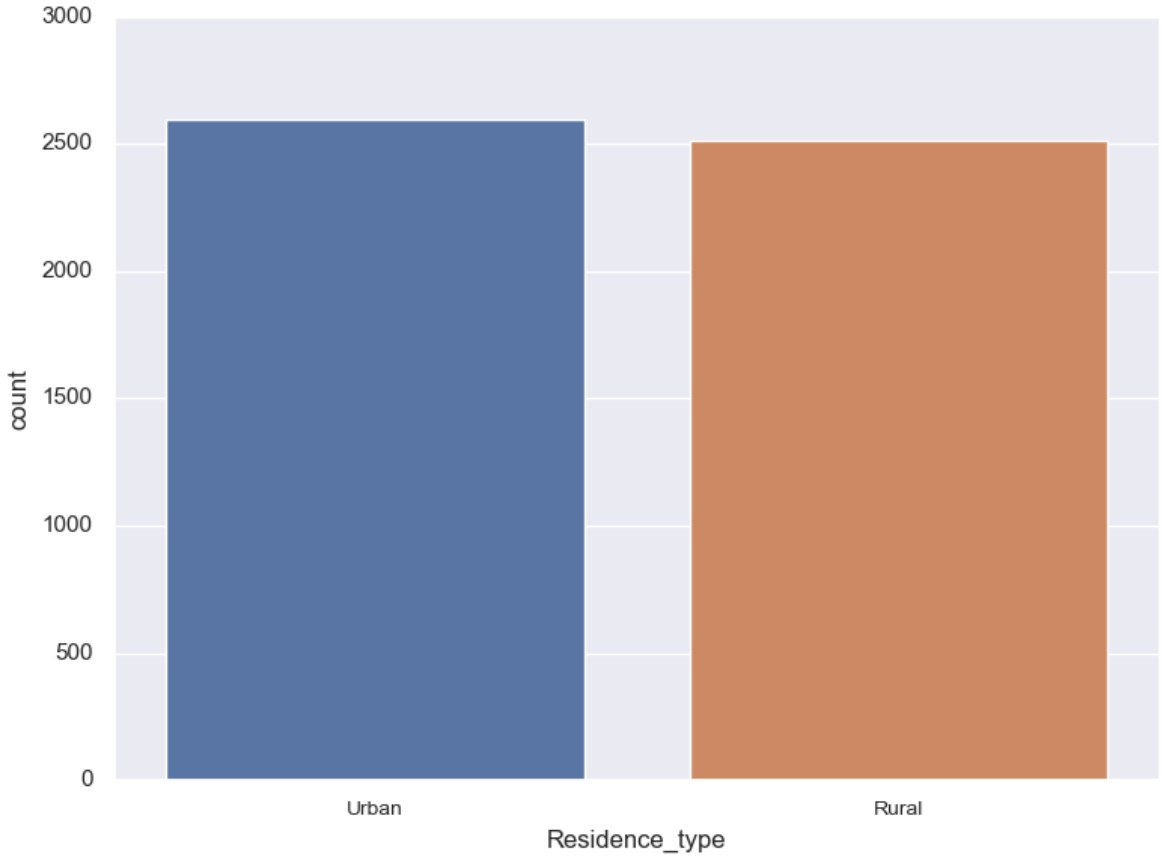

Distribution of Ages
Mean Age value: 43.23
Density
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
−20
0
20
40
60
80
100
Age (in years)

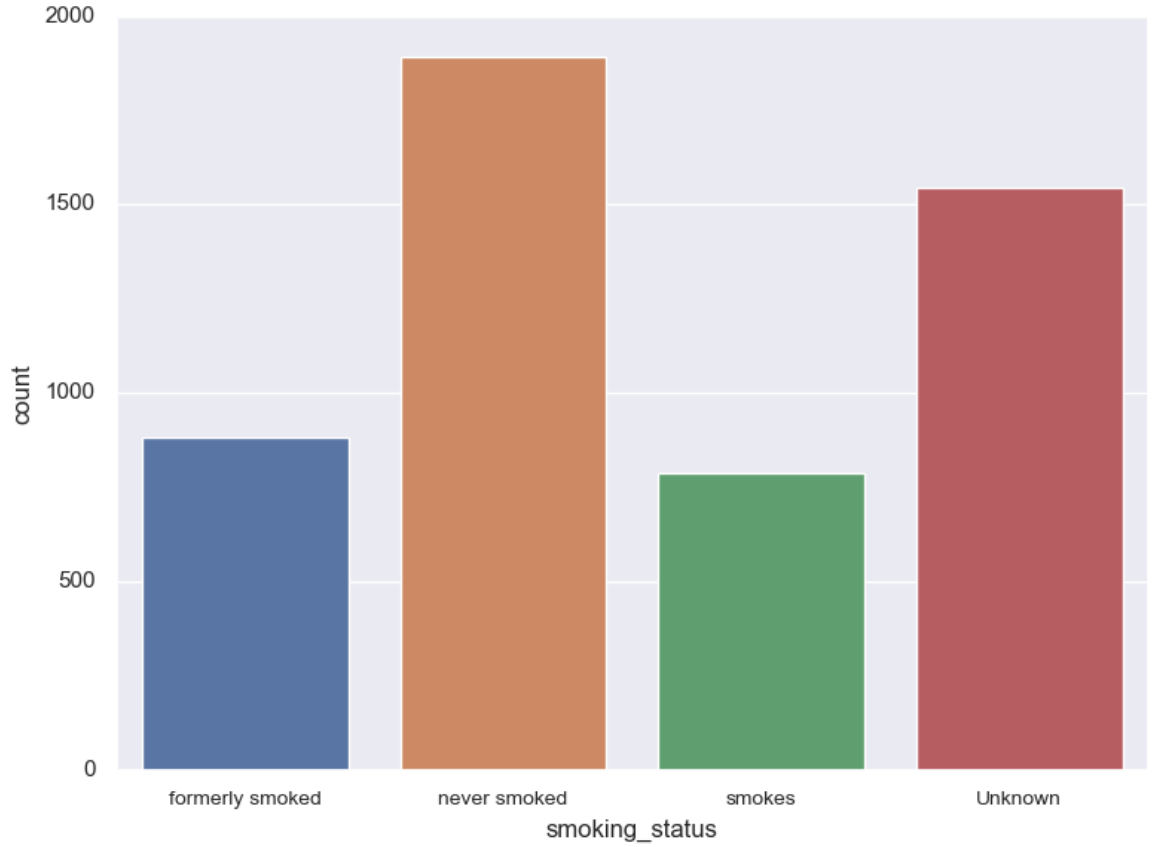

3000
2500
2000
1500
1000
500
0
count
Male
Female
Other
gender

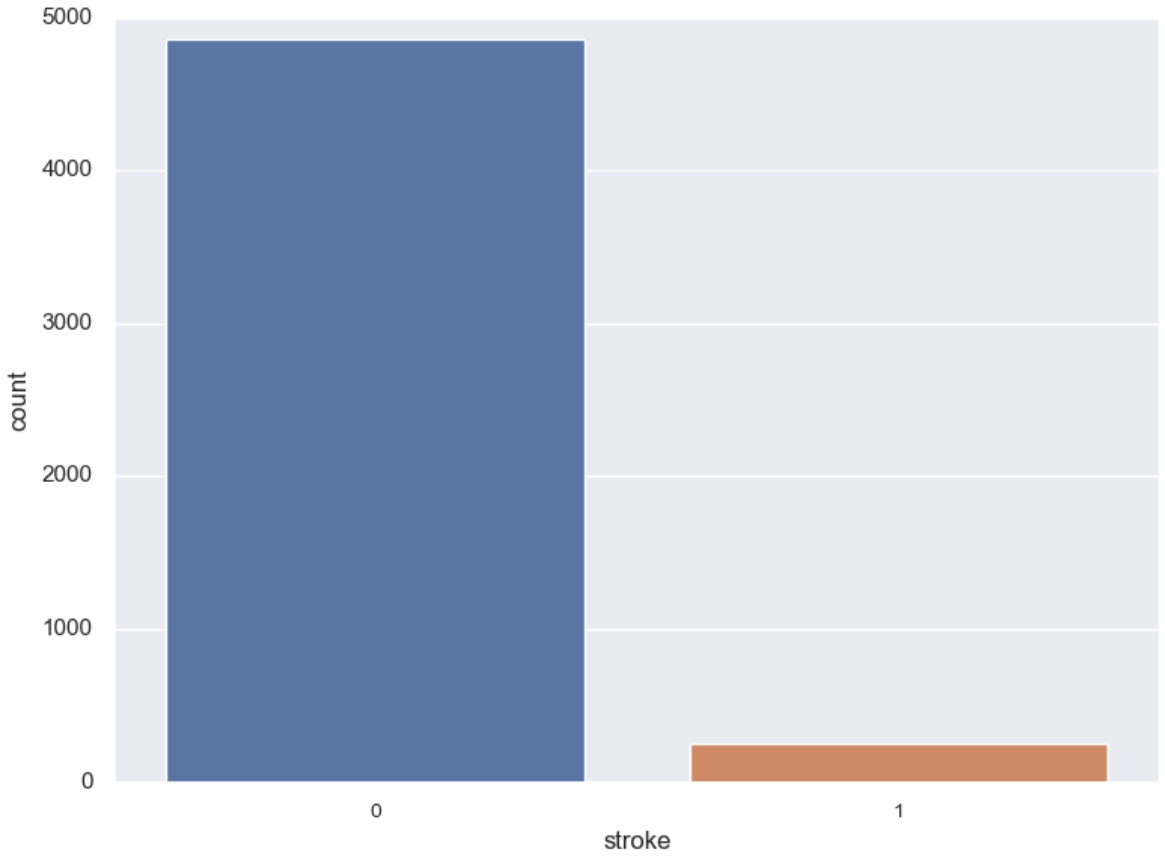

5000
4000
3000
2000
1000
0
count
0
1
hypertension

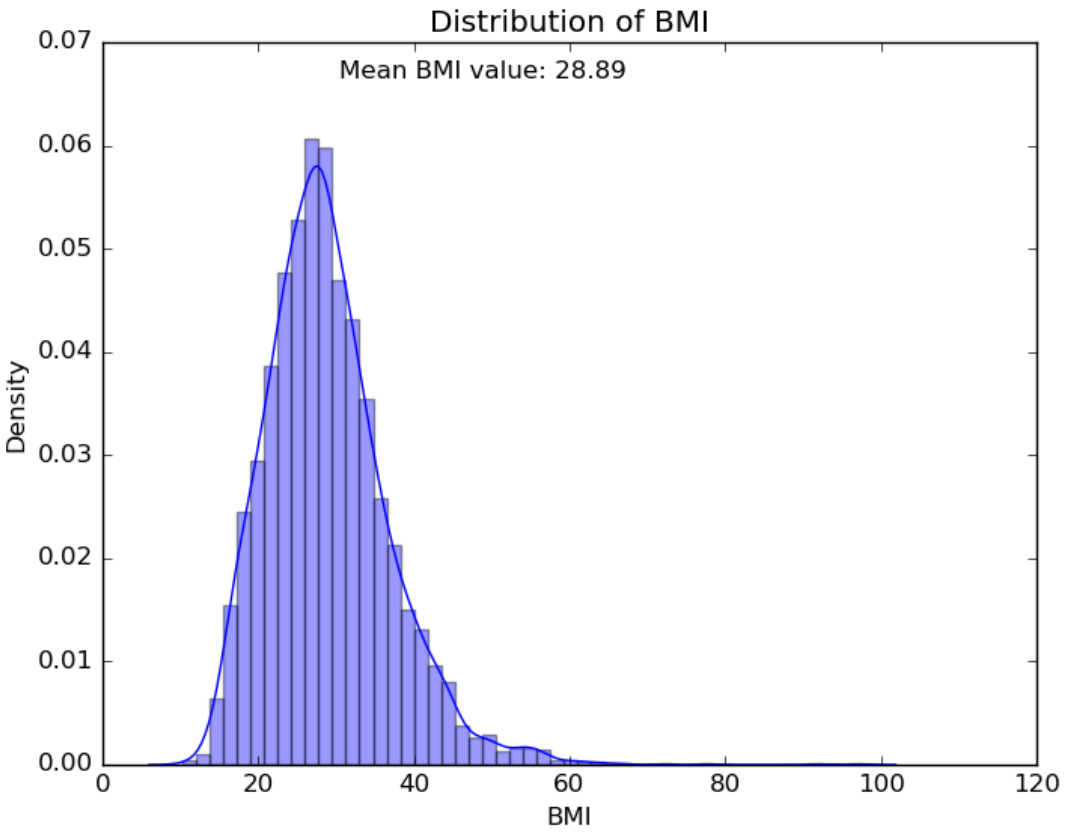

3000
2500
2000
1500
1000
500
0
count
Private
Self-employed
Govt_job
children
Never_worked
work_type


3500
3000
2500
2000
1500
1000
500
0
count
Yes
No
ever_married


3000
2500
2000
1500
1000
500
0
count
Urban
Rural
Residence_type


2000
1500
1000
500
0
count
formerly smoked
never smoked
smokes
Unknown
smoking_status


5000
4000
3000
2000
1000
0
count
0
1
stroke



Αναλυτικές πληροφορίες για τα παραπάνω γραφήματα:

```
Female    2994
Male      2115
Other        1
Name: gender, dtype: int64
0    4612
1     498
Name: hypertension, dtype: int64
Private          2925
Self-employed     819
children          687
Govt_job          657
Never_worked       22
Name: work_type, dtype: int64
Yes    3353
No     1757
Name: ever_married, dtype: int64
Urban    2596
Rural    2514
Name: Residence_type, dtype: int64
never smoked       1892
Unknown            1544
formerly smoked     885
smokes              789
Name: smoking_status, dtype: int64
0    4861
1     249
Name: stroke, dtype: int64
```

Υποερώτημα 2:

Σε αυτο το ερώτημα χειρίζομαι τις ελλιπείς τιμές (missing values) με τις διάφορες μεθόδους. Παραθέτω τα παρακάτω screenshots μετά την εκτέλεση του κώδικα.

Από το αρχείο **exercise1/question_b.py**

Το αρχικό dataset:

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

Το πρώτο βήμα που κάνω είναι να κωδικοποιήσω (encode) όλες τις categorical μεταβλητές σε numerical (π.χ για το gender Male -> 0 και Female -> 1)

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

1ο ζητούμενο:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	stroke
0	0	67.0	0	1	0	0	0	228.69	1
1	1	61.0	0	0	0	1	1	202.21	1
2	0	80.0	0	1	0	0	1	105.92	1
3	1	49.0	0	0	0	0	0	171.23	1
4	1	79.0	1	0	0	1	1	174.12	1
...	...	...	...	...	...	...	...	...	...
5105	1	80.0	1	0	0	0	0	83.75	0
5106	1	81.0	0	0	0	1	0	125.20	0
5107	1	35.0	0	0	0	1	1	82.99	0
5108	0	51.0	0	0	0	0	1	166.29	0
5109	1	44.0	0	0	0	2	0	85.28	0

2ο ζητούμενο:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	stroke
0	0	67.0	0	1	0	0	0	228.69	36.600000	1
1	1	61.0	0	0	0	1	1	202.21	28.893237	1
2	0	80.0	0	1	0	0	1	105.92	32.500000	1
3	1	49.0	0	0	0	0	0	171.23	34.400000	1
4	1	79.0	1	0	0	1	1	174.12	24.000000	1
...	...	...	...	...	...	...	...	...	...	...
5105	1	80.0	1	0	0	0	0	83.75	28.893237	0
5106	1	81.0	0	0	0	1	0	125.20	40.000000	0
5107	1	35.0	0	0	0	1	1	82.99	30.600000	0
5108	0	51.0	0	0	0	0	1	166.29	25.600000	0
5109	1	44.0	0	0	0	2	0	85.28	26.200000	0

3ο ζητούμενο:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	stroke
0	0	67.0	0	1	0	0	0	228.69	36.60	1
1	1	61.0	0	0	0	1	1	202.21	34.55	1
2	0	80.0	0	1	0	0	1	105.92	32.50	1
3	1	49.0	0	0	0	0	0	171.23	34.40	1
4	1	79.0	1	0	0	1	1	174.12	24.00	1
...	...	...	...	...	...	...	...	...	...	...
5105	1	80.0	1	0	0	0	0	83.75	29.30	0
5106	1	81.0	0	0	0	1	0	125.20	40.00	0
5107	1	35.0	0	0	0	1	1	82.99	30.60	0
5108	0	51.0	0	0	0	0	1	166.29	25.60	0
5109	1	44.0	0	0	0	2	0	85.28	26.20	0

4ο ζητούμενο:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	smoking_status	stroke
0	0.0	67.0	0.0	1.0	0.0	0.0	0.0	228.69	0.0	1.0
1	1.0	61.0	0.0	0.0	0.0	1.0	1.0	202.21	1.0	1.0
2	0.0	80.0	0.0	1.0	0.0	0.0	1.0	105.92	1.0	1.0
3	1.0	49.0	0.0	0.0	0.0	0.0	0.0	171.23	2.0	1.0
4	1.0	79.0	1.0	0.0	0.0	1.0	1.0	174.12	1.0	1.0
...	...	...	...	...	...	...	...	...	...	...
5105	1.0	80.0	1.0	0.0	0.0	0.0	0.0	83.75	1.0	0.0
5106	1.0	81.0	0.0	0.0	0.0	1.0	0.0	125.20	1.0	0.0
5107	1.0	35.0	0.0	0.0	0.0	1.0	1.0	82.99	1.0	0.0
5108	0.0	51.0	0.0	0.0	0.0	0.0	1.0	166.29	0.0	0.0
5109	1.0	44.0	0.0	0.0	0.0	2.0	0.0	85.28	1.0	0.0

5ο ζητούμενο:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	0	67.0	0	1	0	0	0	228.69	36.60	0.0	1
1	1	61.0	0	0	0	1	1	202.21	34.55	1.0	1
2	0	80.0	0	1	0	0	1	105.92	32.50	1.0	1
3	1	49.0	0	0	0	0	0	171.23	34.40	2.0	1
4	1	79.0	1	0	0	1	1	174.12	24.00	1.0	1
...	...	...	...	...	...	...	...	...	...	...	...
5105	1	80.0	1	0	0	0	0	83.75	29.30	1.0	0
5106	1	81.0	0	0	0	1	0	125.20	40.00	1.0	0
5107	1	35.0	0	0	0	1	1	82.99	30.60	1.0	0
5108	0	51.0	0	0	0	0	1	166.29	25.60	0.0	0
5109	1	44.0	0	0	0	2	0	85.28	26.20	1.0	0

Σημείωση: Ο αναλυτικός τρόπος καθαρισμού και σκέψης θα παρουσιαστεί στην προφορική εξέταση καθώς θα έπρεπε να γράψω μια τεράστια αναφορά. Επισημαίνω όμως ότι ο κώδικας έχει παντού λεπτομερή σχόλια που μπορούν να βοηθήσουν.

Υποερώτημα 3:

Σε αυτο το ερώτημα πρέπει να προβλέψω αν ένας ασθενής είναι επιρρεπής ή όχι να πάθει εγκεφαλικό χρησιμοποιώντας Random Forest. Για την πραγματοποίηση αυτού του ερωτήματος προσπάθησα πολλούς τρόπους. Αρχικά να δοκιμάσω για τα 5 διαφορετικα dataset του προηγούμενου ερωτήματος. Επιπλέον χρησιμοποίησα ένα Min-Max Scaler ώστε να μη αλλοιώνουν τα αποτελέσματα πολύ μεγάλες (π.χ επίπεδα γλυκόζης) ή πολύ μικρές τιμές και να είναι στο ίδιο εύρος όλα τα δεδομένα. Ακόμα χρησιμοποίησα την μέθοδο GridSearchCV ώστε να πάρω μετά από εξονυχιστικό ψάξιμο του προγράμματος τις καλύτερες παραμέτρους για το

RandomForestClassifier. Παρόλα Αυτά επειδή το dataset είναι πολύ ανισόρροπο όπως είδαμε από το πρώτο ερώτημα στα stroke values και επειδή στην εξόρυξη δεδομένων το σημαντικότερο πράγμα είναι το καλό και ισορροπημένο dataset, τα αποτελέσματα (precision, recall, f1-score) δεν ήταν τα βέλτιστα. Παραθέτω το παρακάτω screenshot μετά την εκτέλεση του κώδικα.

** τα built-in εργαλεία confusion_matrix και classification_report της βιβλιοθήκης scikit-learn χρησιμοποιήθηκαν για την εξαγωγή των παρακάτω αποτελεσμάτων. **

Από το αρχείο **exercise1/question_c.py**

```
[[1225    1]
 [  50    2]]
              precision    recall  f1-score   support

         0.0       0.96      1.00      0.98      1226
         1.0       0.67      0.04      0.07        52

    accuracy                           0.96      1278
   macro avg       0.81      0.52      0.53      1278
weighted avg       0.95      0.96      0.94      1278
```

Ερώτημα 2:

Γενική Περιγραφή: Στο συγκεκριμένο ερώτημα κλήθηκα να προβλέψω αν ένα email είναι spam ή όχι. Για να το λύσω αρχικά έσπασα σε tokens το email row και να μετατρέψω τις λέξεις σε αριθμούς. Μέτα εφάρμοσα padding (αν πχ μια πρόταση έχει 10 λέξεις και μια άλλη εχει 20, θα γεμίσω με 0 την πρώτη για όσα της λείπουν και φυσικά η αναπαράσταση των προτάσεων/λέξεων γίνεται με αριθμούς σε αυτό το στάδιο). Στην συνέχεια χρησιμοποιώ ένα pre-trained μοντέλο GloVe το οποίο μπορείτε να κατεβάσετε από εδώ: https://nlp.stanford.edu/projects/glove/
για να μετατρέψω το κείμενο των email σε διανύσματα. Τέλος καλώ ένα νευρωνικό δίκτυο του οποία η δομή φαίνεται παρακάτω:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 80, 100)           2212200
_________________________________________________________________
conv1d (Conv1D)              (None, 79, 256)           51456
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 39, 256)           0
_________________________________________________________________
dropout (Dropout)            (None, 39, 256)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 38, 128)           65664
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 19, 128)           0
_________________________________________________________________
dropout_1 (Dropout)          (None, 19, 128)           0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 18, 64)            16448
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 9, 64)             0
_________________________________________________________________
dropout_2 (Dropout)          (None, 9, 64)             0
_________________________________________________________________
flatten (Flatten)            (None, 576)               0
_________________________________________________________________
dense (Dense)                (None, 1)                 577
=================================================================
Total params: 2,346,345
Trainable params: 134,145
Non-trainable params: 2,212,200
```

για να εκπαιδεύτει και να προβλέψει στο testing dataset αν ένα email είναι spam ή όχι.

Τα αποτελέσματα σε αυτό το ερώτημα ήταν πολύ καλύτερα σε σχέση με το πρώτο ερώτημα και αυτό οφείλεται στο πιο ισορροπημένο dataset. Ενδεικτικά αναφέρω με 50 επαναλήψεις/epochs:

```
Accuracy: 99.733335
[[239   0]
 [ 21 115]]
              precision    recall  f1-score   support

           0       0.92      1.00      0.96       239
           1       1.00      0.85      0.92       136

    accuracy                           0.94       375
   macro avg       0.96      0.92      0.94       375
weighted avg       0.95      0.94      0.94       375
```

Εργαλεία που χρησιμοποιήθηκαν

Περιβάλλον υλοποίησης
Visual Studio Code

Βιβλιοθήκες λογισμικού
Pandas
Numpy
Matplotlib
Seaborn
Tensorflow
Keras
scikit-learn

Εγκατάσταση
Αρχικά εγκαθιστούμε το VSC (Visual Studio Code) στον υπολογιστή μας. Στη συνέχεια κάνουμε install το extension της python στο VSC. Ύστερα, στο terminal που υπάρχει μέσα στο VSC, εγκαθιστούμε κάθε βιβλιοθήκη που θέλουμε να χρησιμοποιήσουμε πληκτρολογόντας “pip install” + την βιβλιοθήκη που θέλουμε να χρησιμοποιήσουμε.