ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Data Mining Techniques

Assignment 2

Spring Semester 2016-2017

Zacharopoulou Lida - AM: 1115201100004
Seintaridis Dimitrios - AM: 1115201100197

1. Visualization - Οπτικοποίηση Δεδομένων

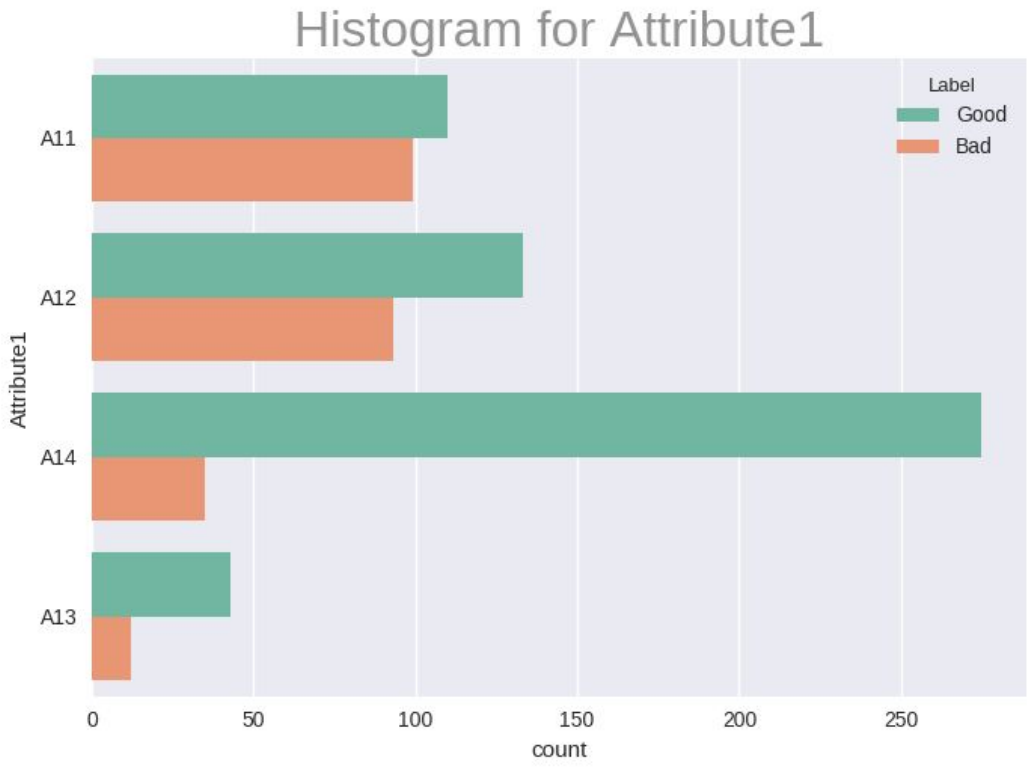
Διαγράμματα

Για την οπτικοποίηση των δεδομένων, χρησιμοποιήσαμε τη βιβλιοθήκη Seaborn της Python, που είναι βασισμένη στην matplotlib.

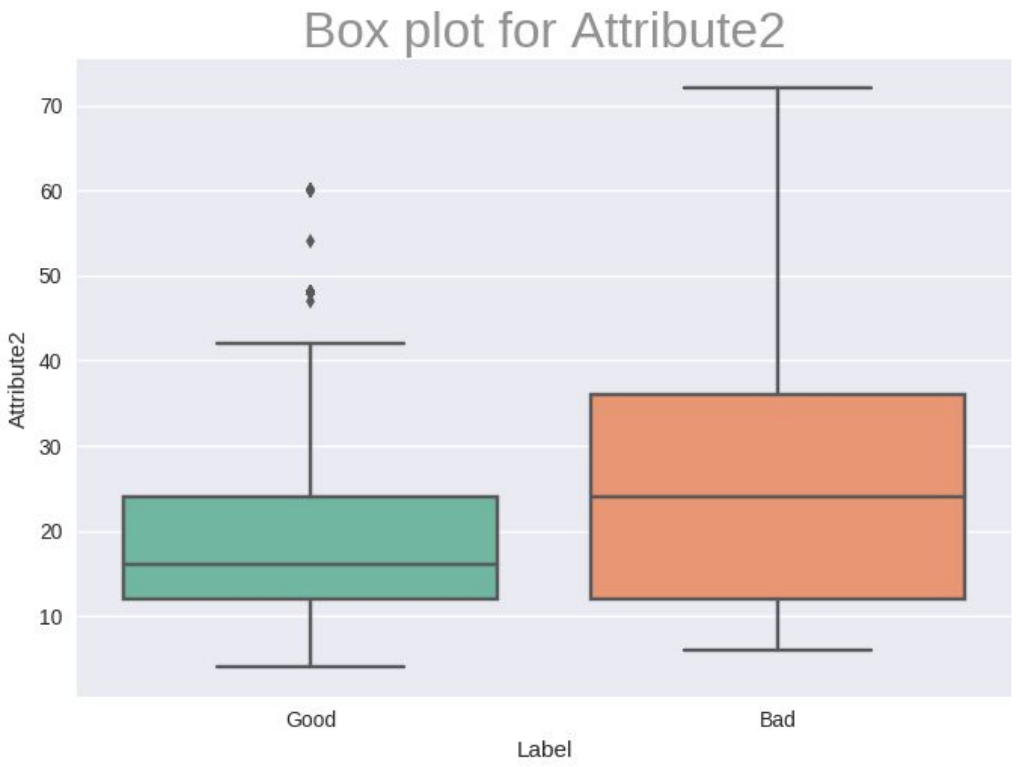
Για κάθε categorical feature, παρουσιάζουμε ένα Histogram με τον αριθμό των clients που έχουν χαρακτηριστεί ως Good και Bad, ενώ αντίστοιχα για κάθε numerical feature χρησιμοποιούμε ένα Box plot.

Σε κάθε περίπτωση οι Good και οι Bad οπτικοποιούνται στο ίδιο plot με διαφορετικό χρώμα.

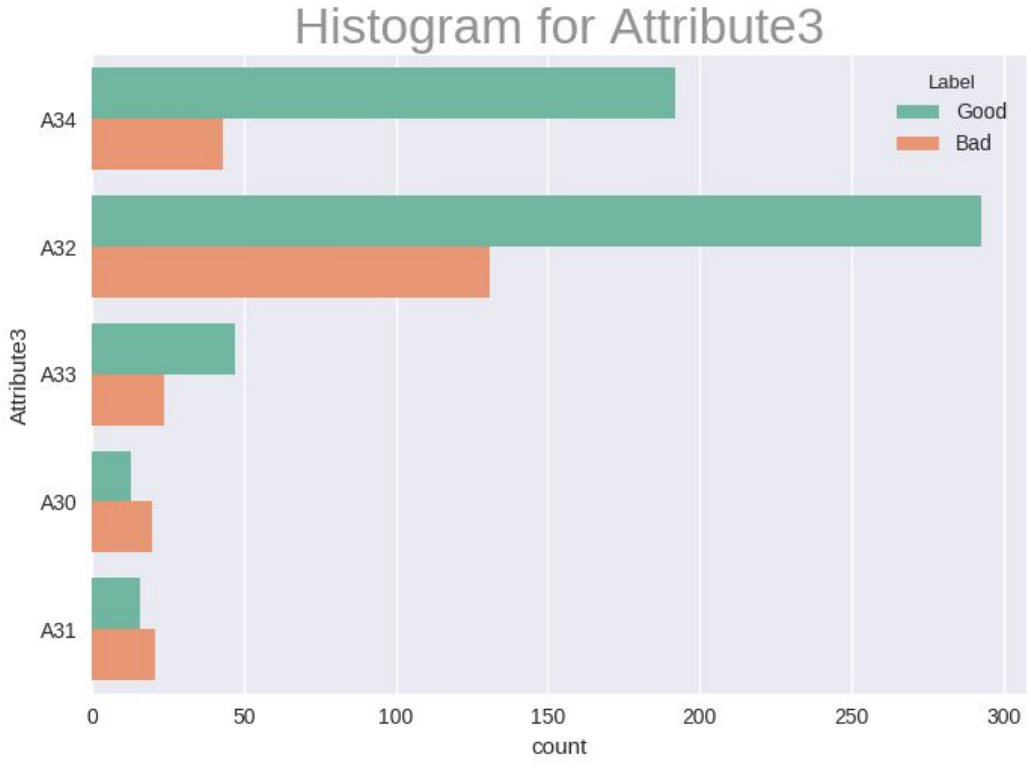
Attribute 1: Status of existing checking account (Categorical)



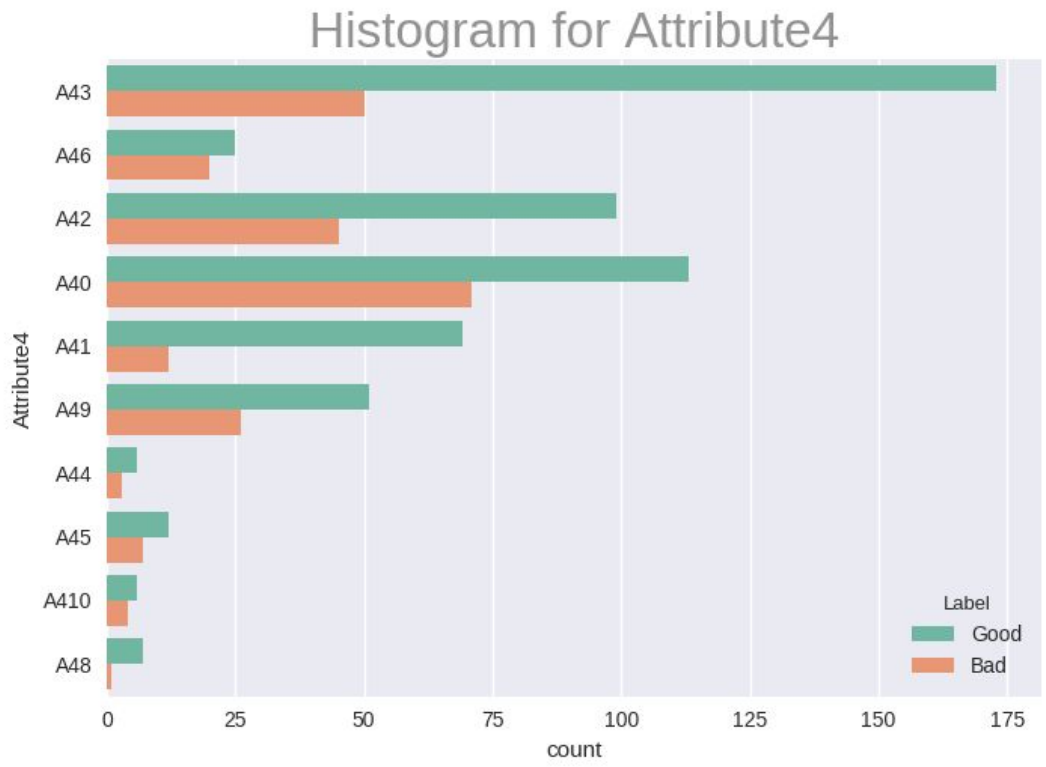
Attribute 2: Duration in months (Numerical)



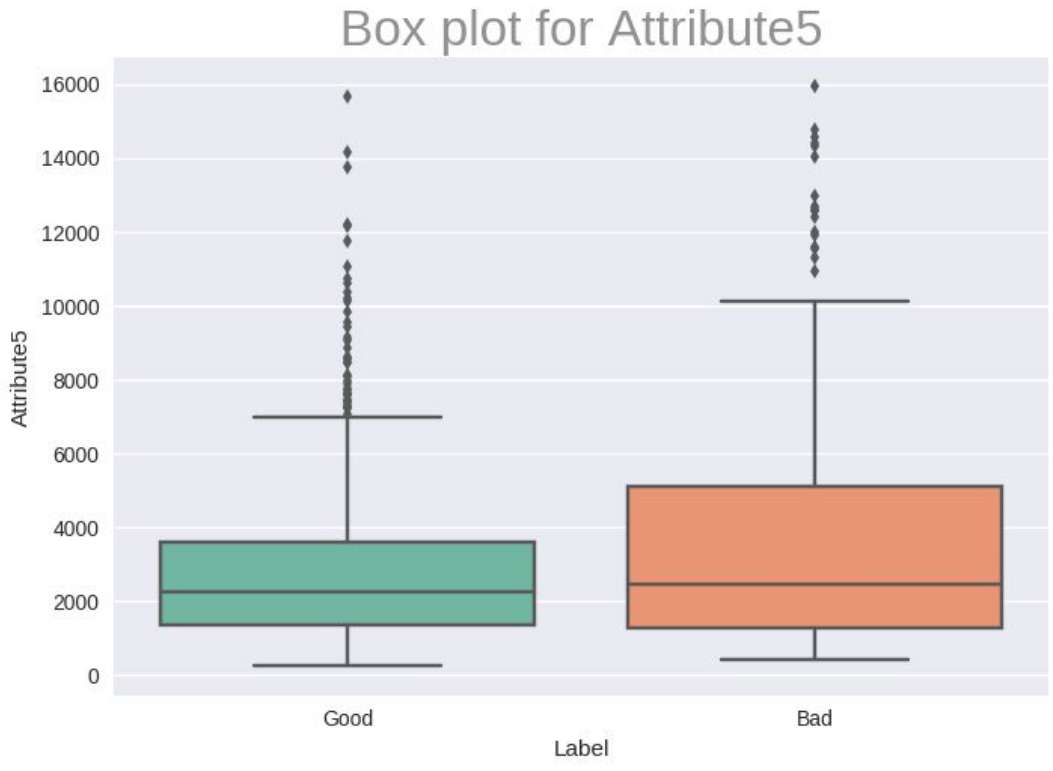
Attribute 3: Credit history (Categorical)



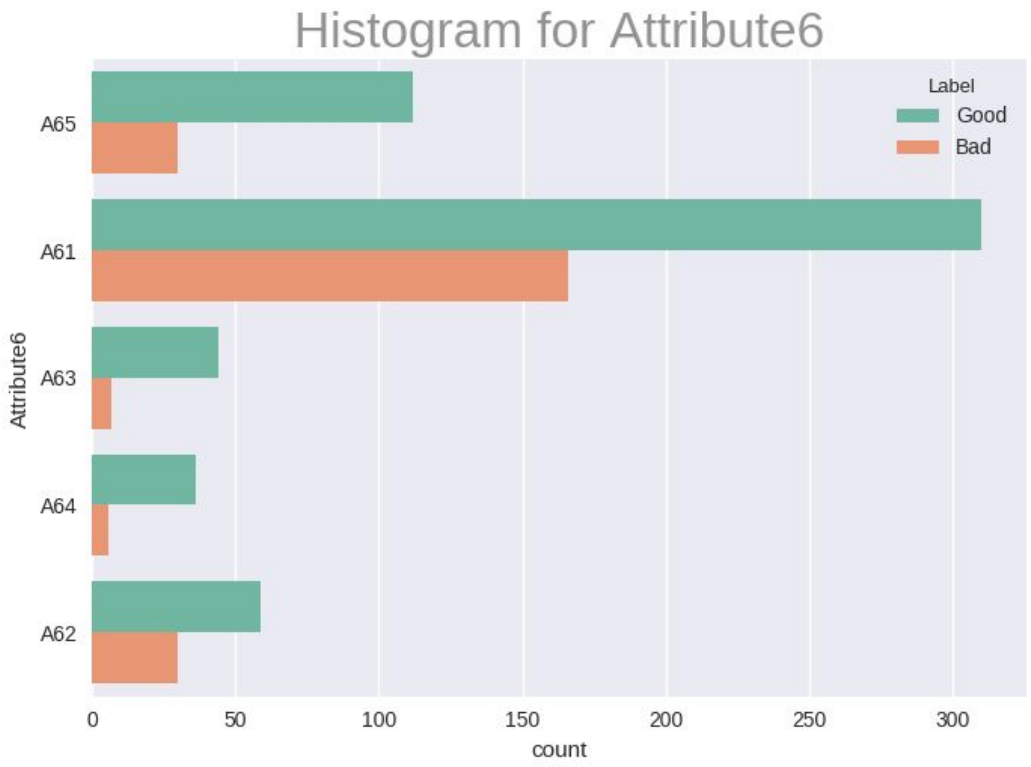
Attribute 4: Purpose (Categorical)



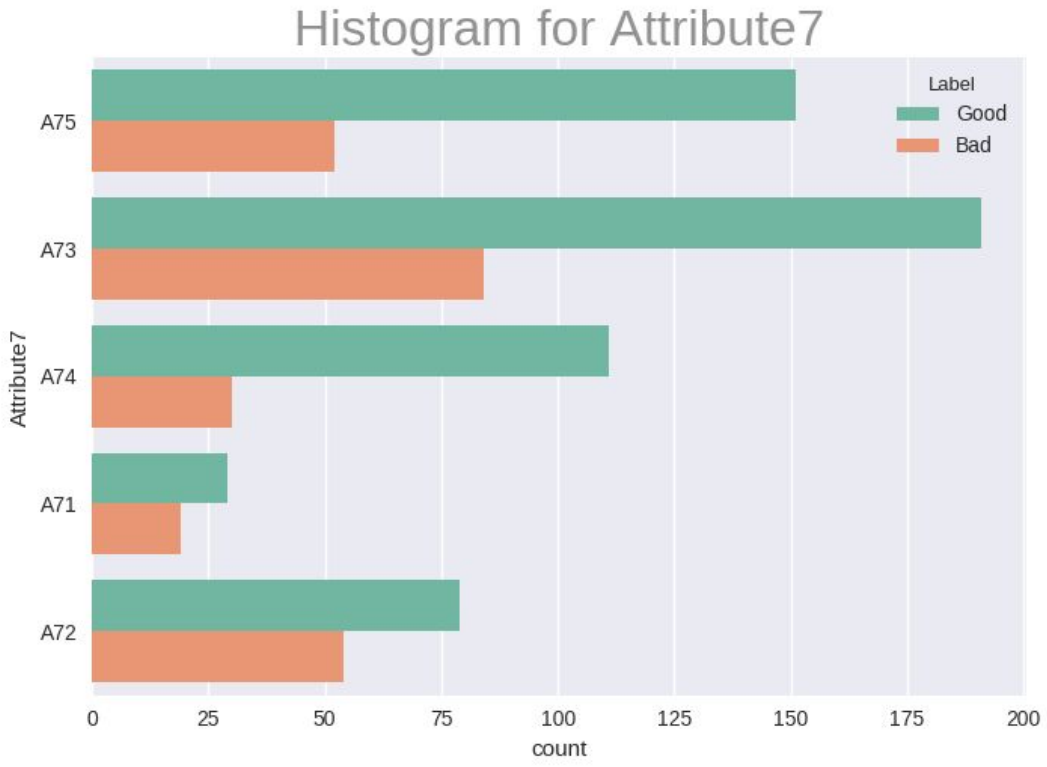
Attribute 5: Credit amount (Numerical)



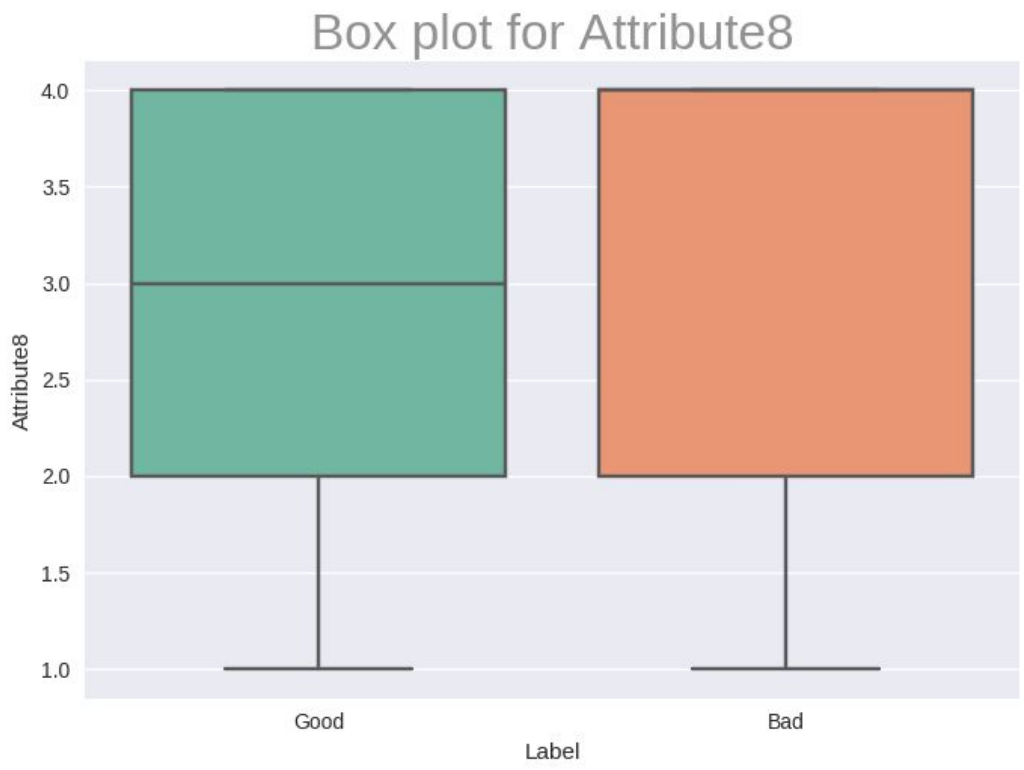
Attribute 6: Savings account/bonds (Categorical)



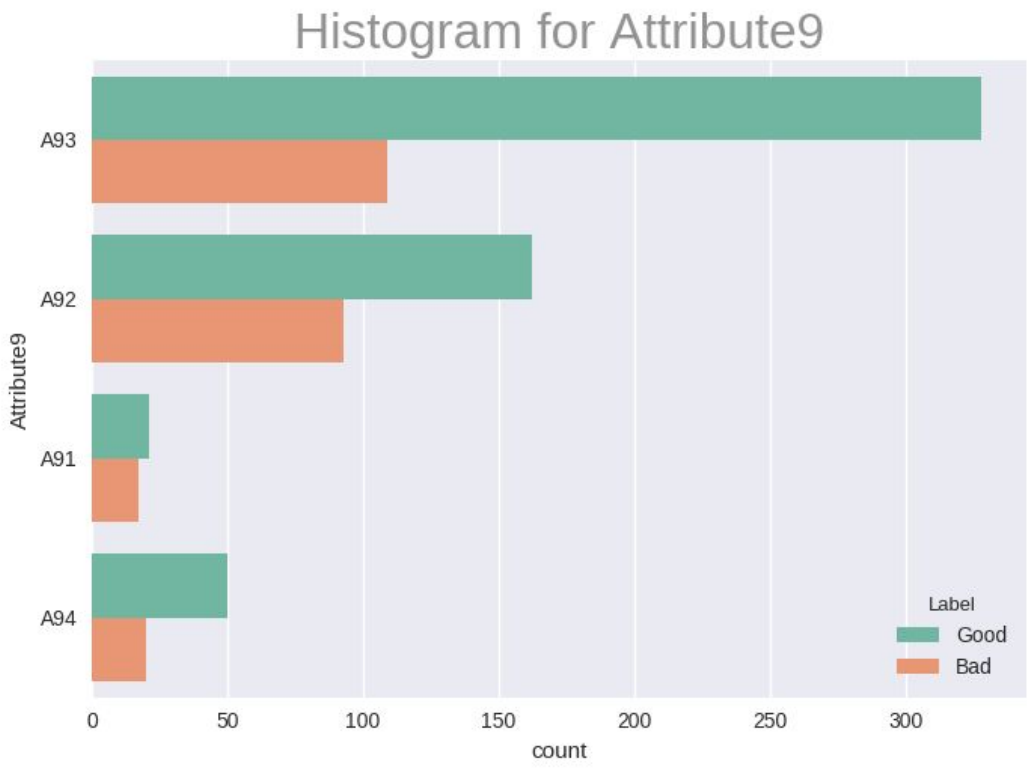
Attribute 7: Present employment since (Categorical)



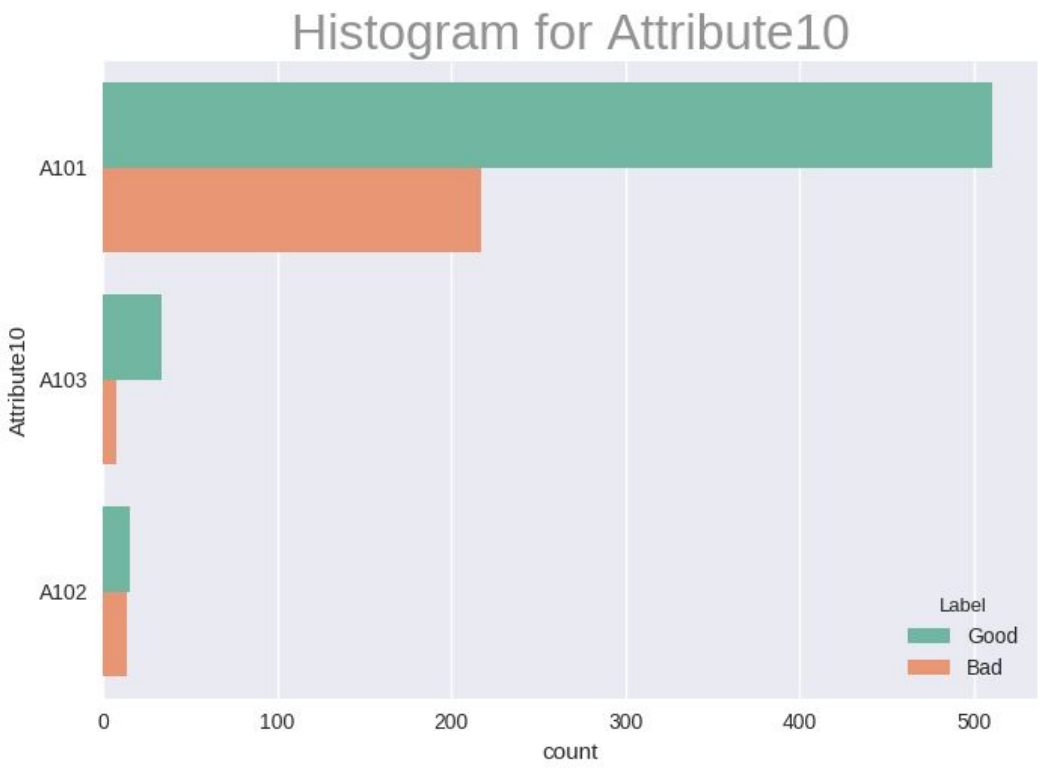
Attribute 8: Installment rate in percentage of disposable income (Numerical)



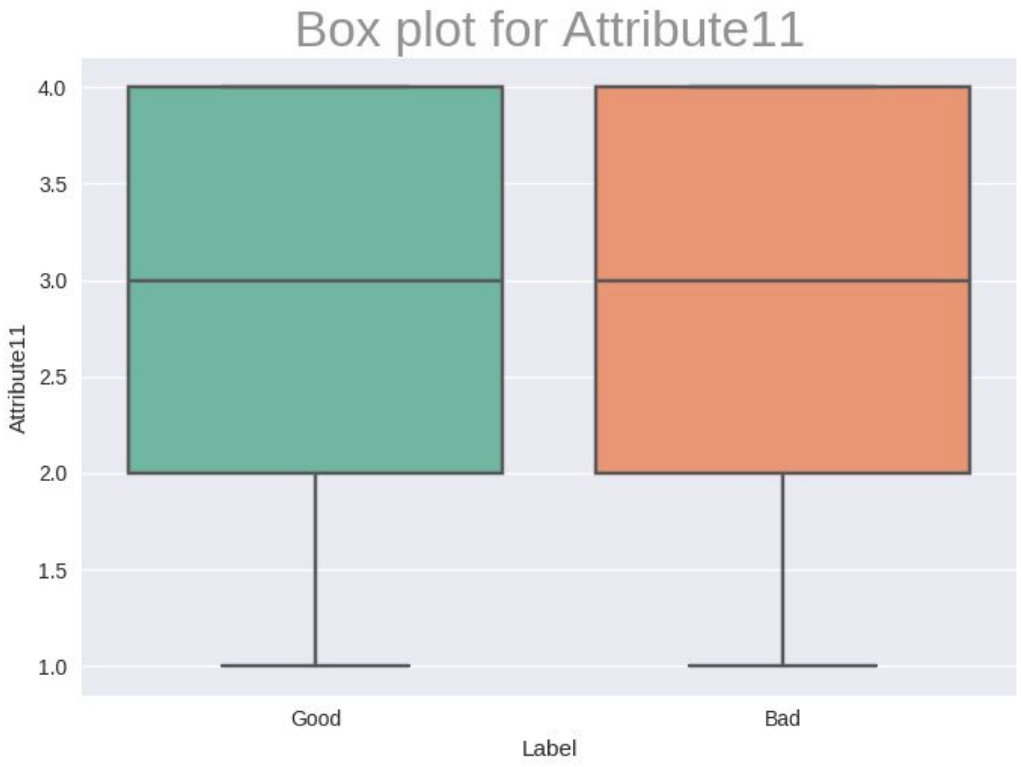
Attribute 9: Personal status and sex (Categorical)



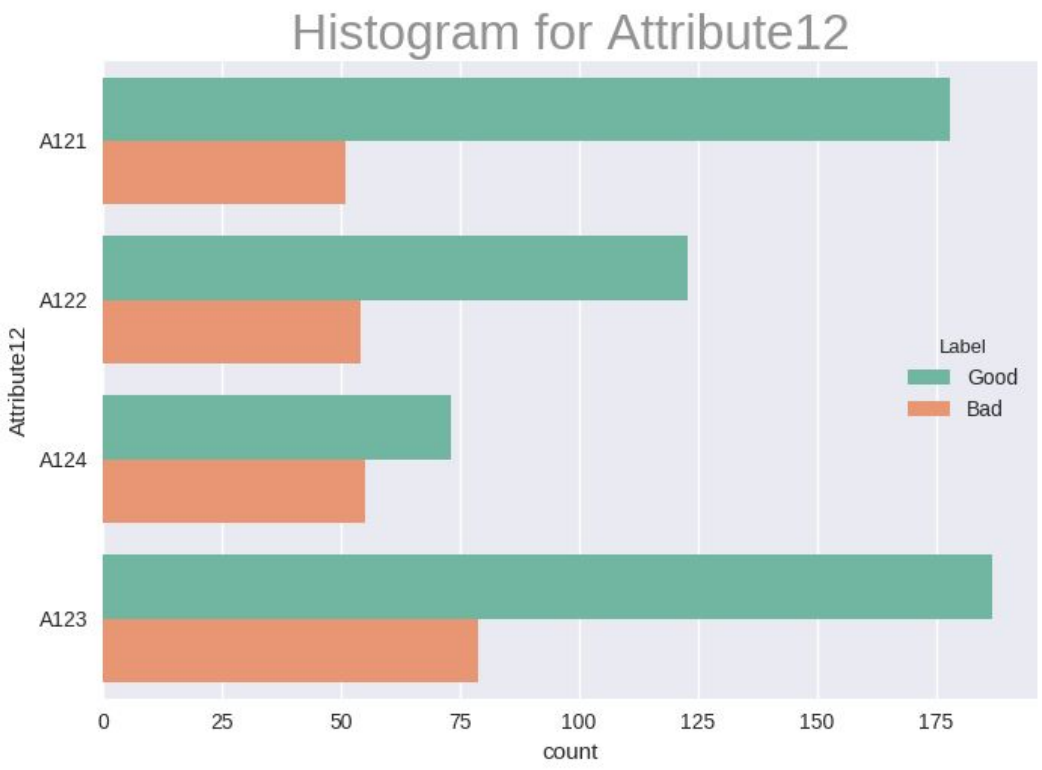
Attribute 10: Other debtors / guarantors (Categorical)



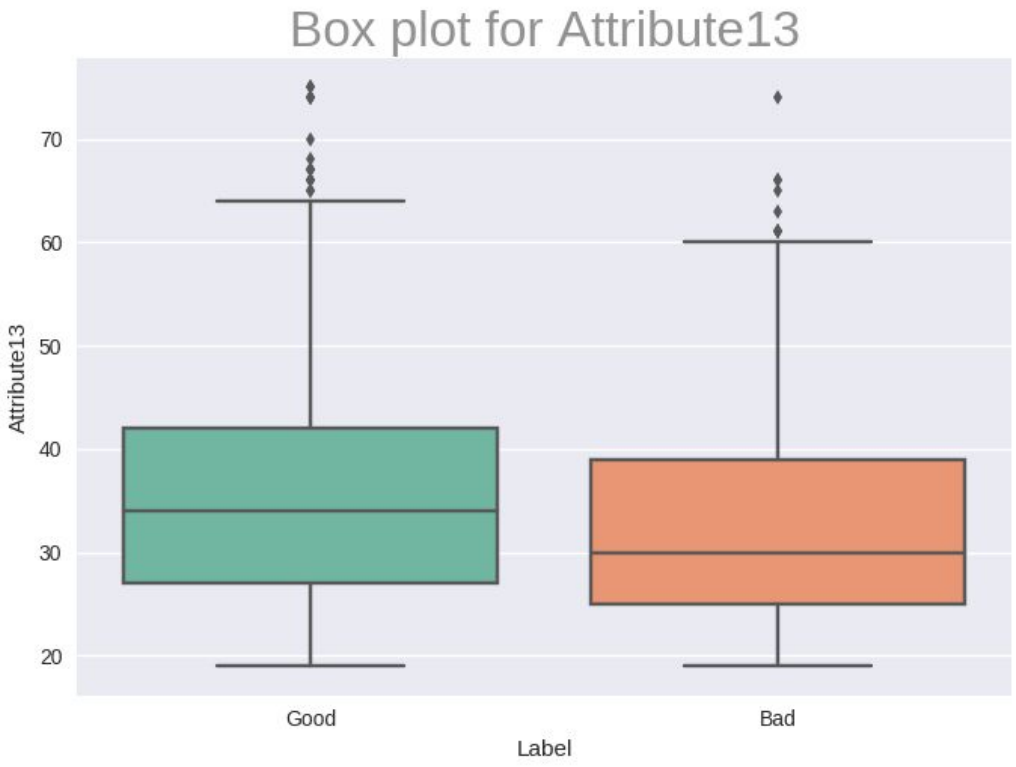
Attribute 11: Present residence since (Numerical)



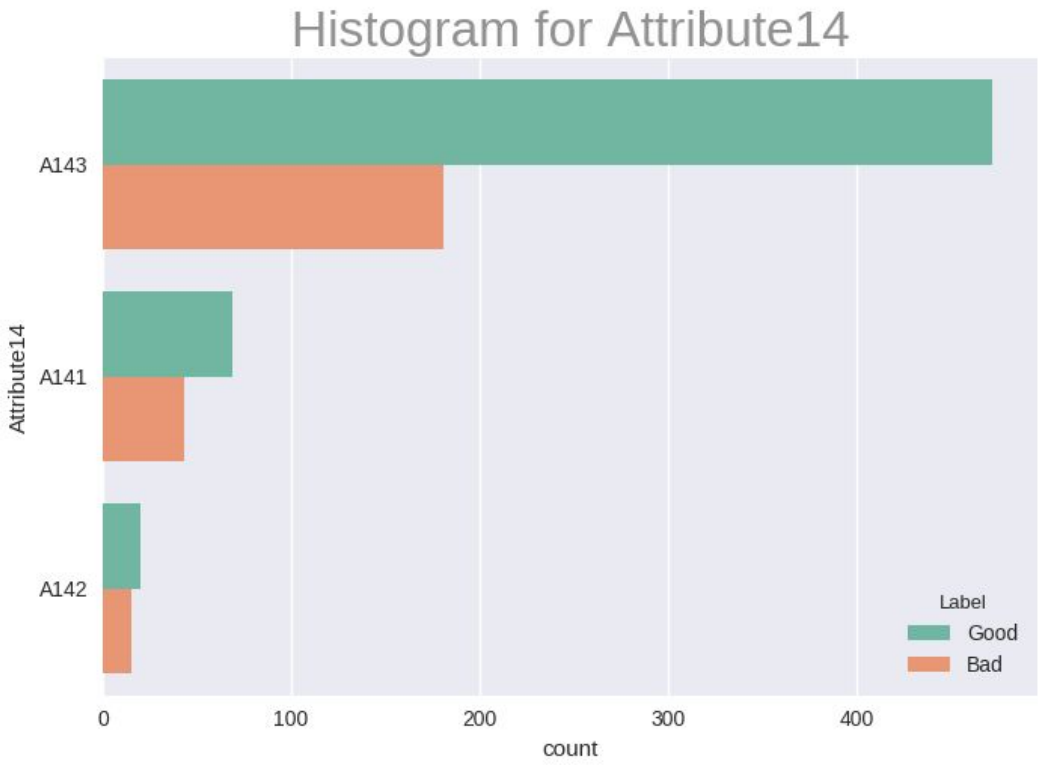
Attribute 12: Property (Categotical)



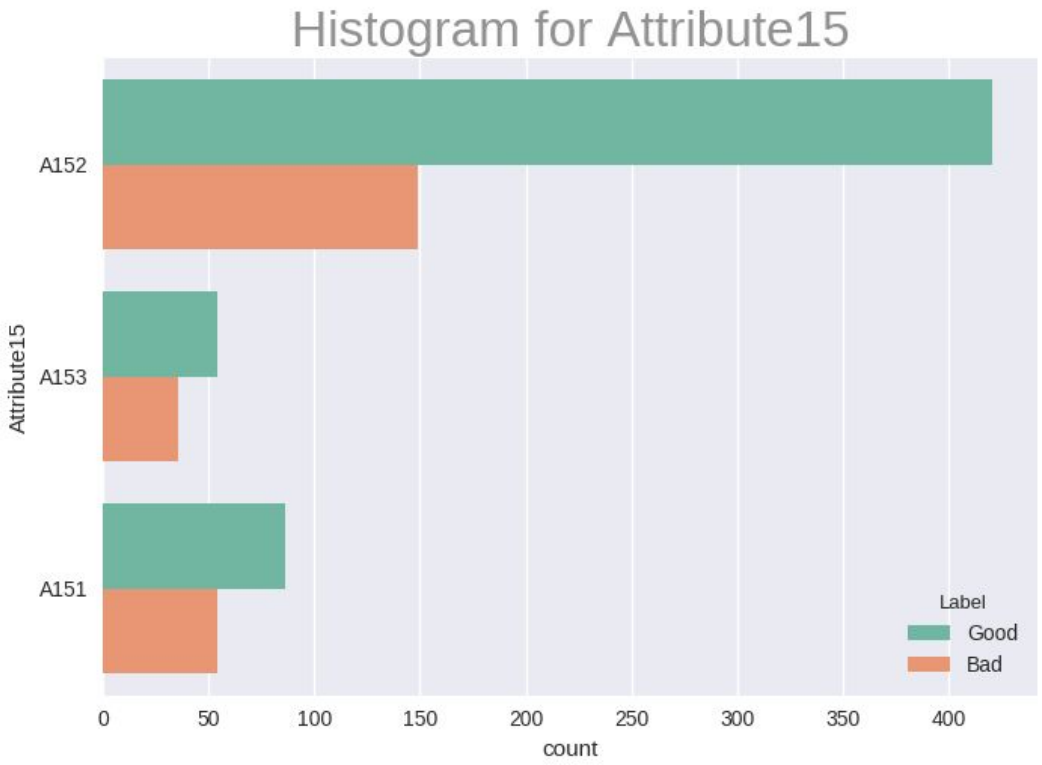
Attribute 13: Age in years (Numerical)



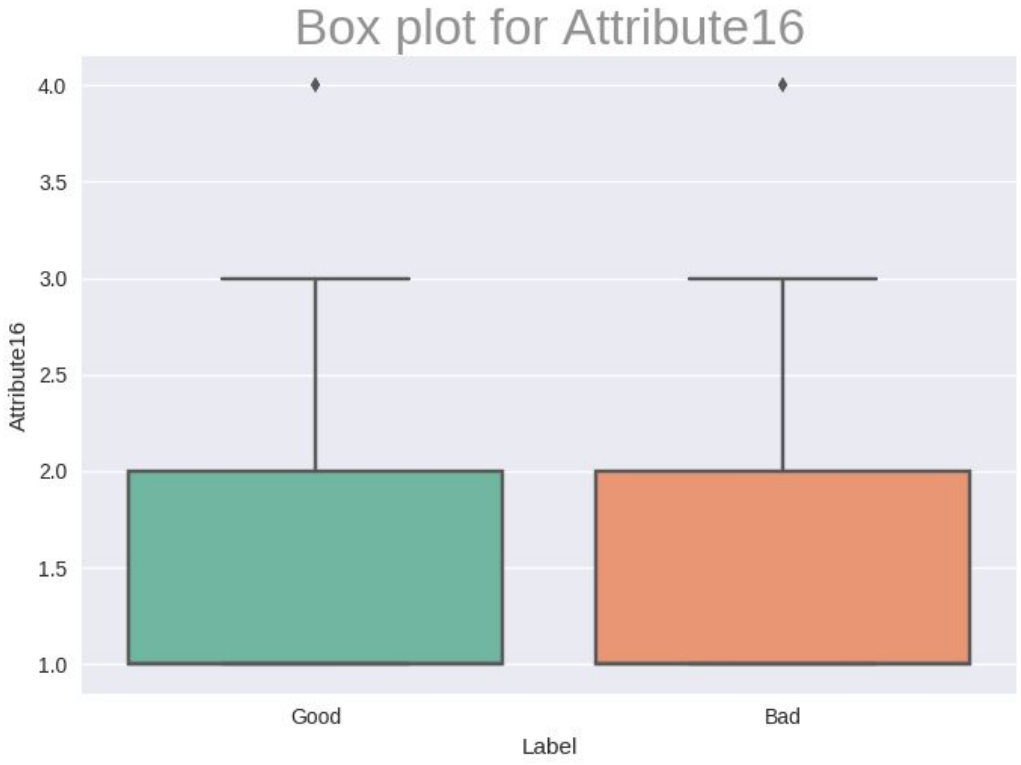
Attribute 14: Other installment plans (Categorical)



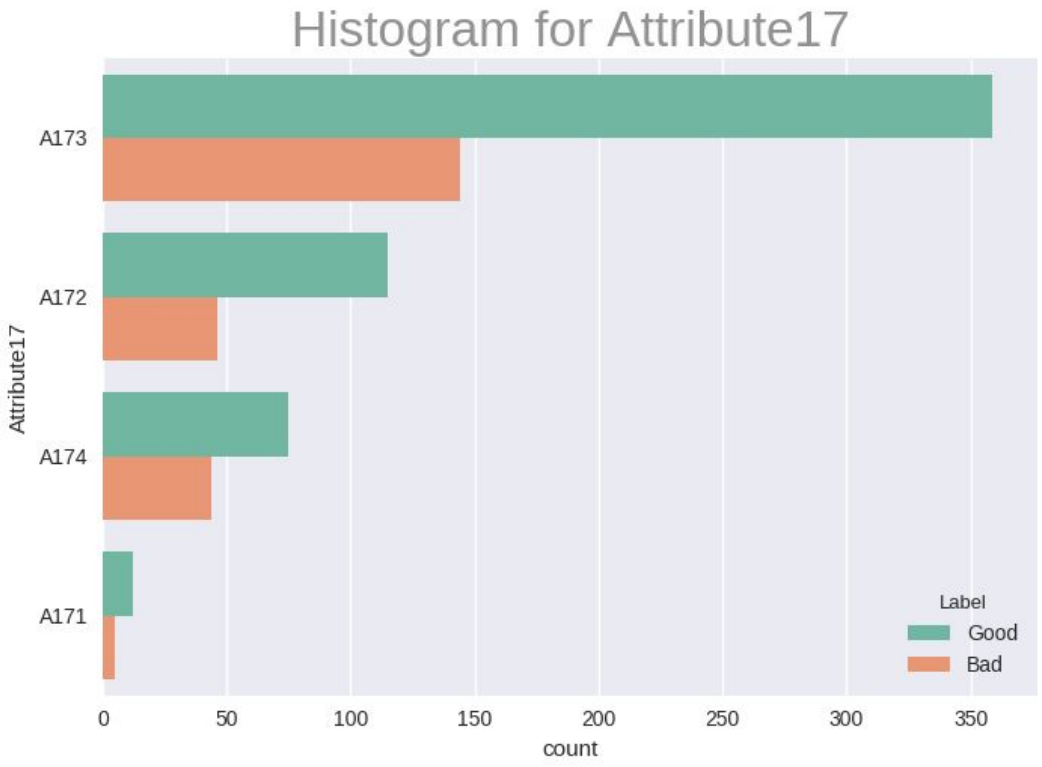
Attribute 15: Housing (Categorical)



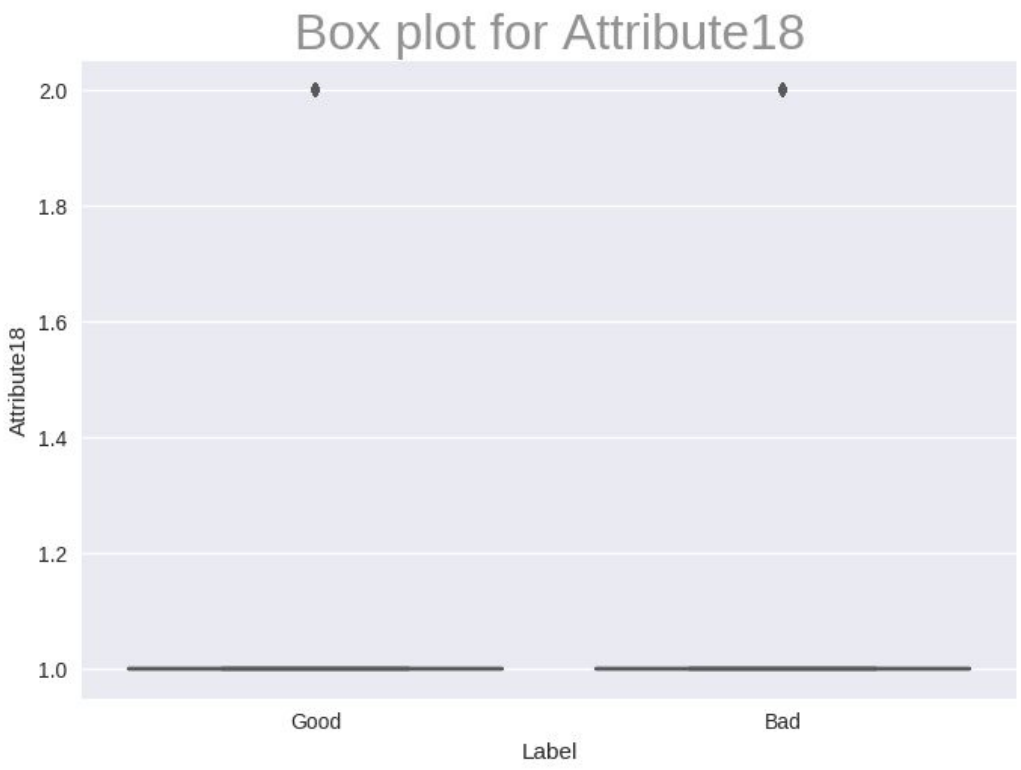
Attribute 16: Number of existing credits at this bank (Numerical)



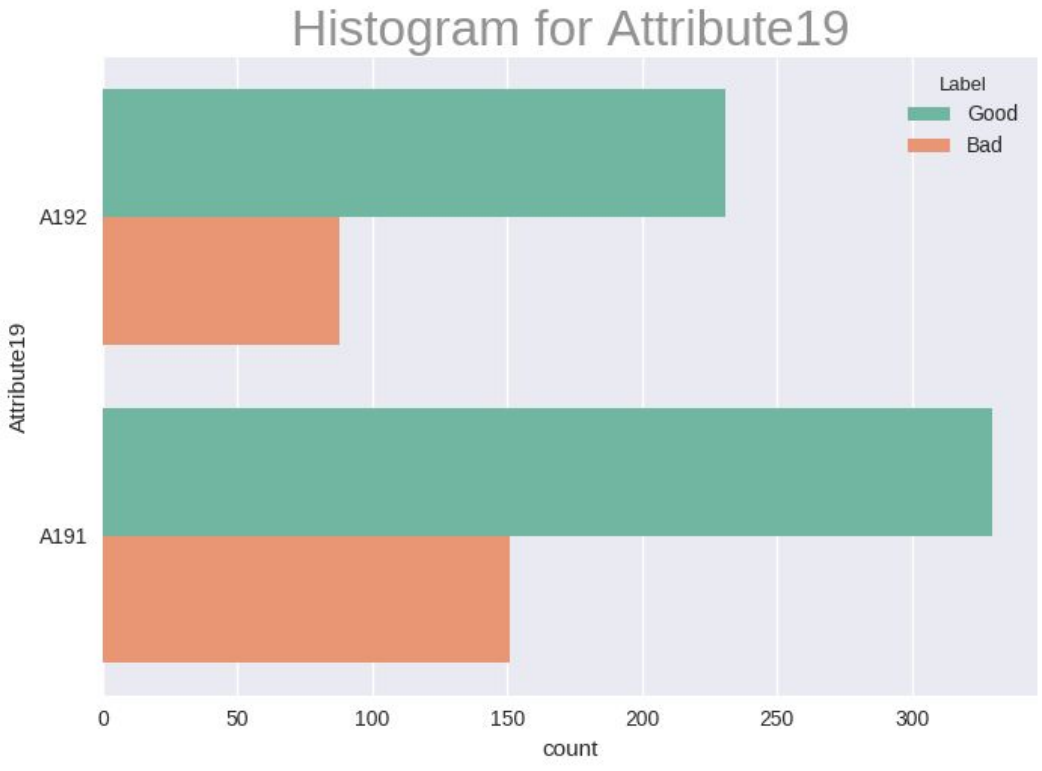
Attribute 17: Job (Categorical)



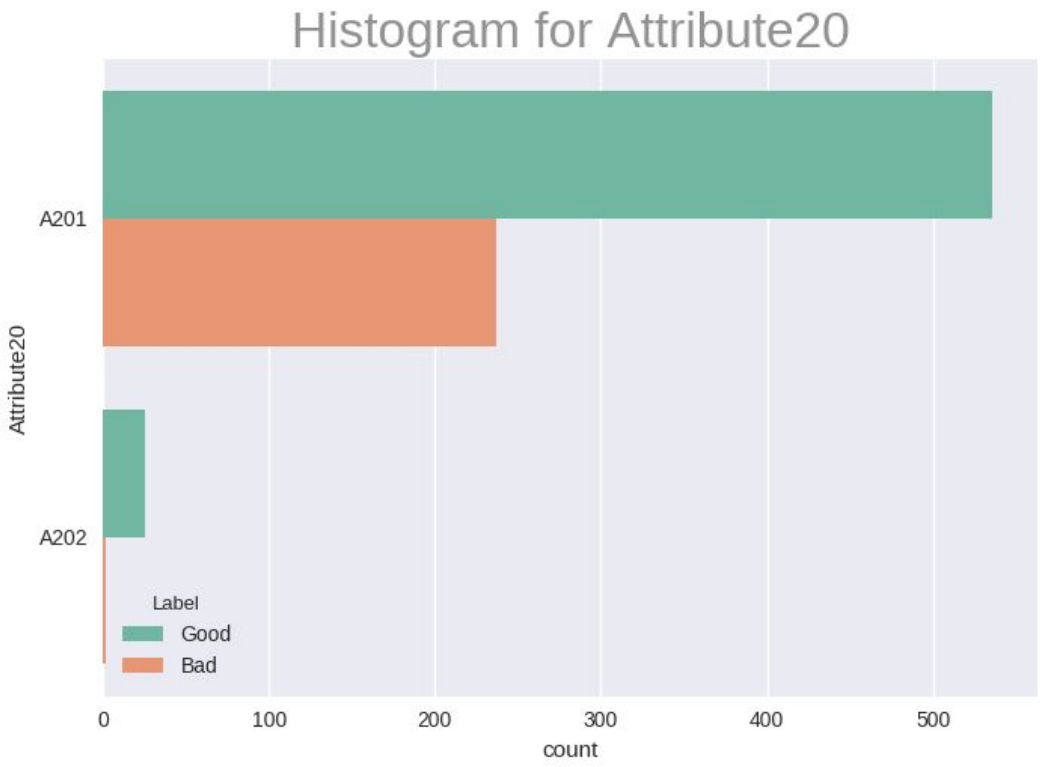
Attribute 18: Number of people being liable to provide maintenance for (Numerical)



Attribute 19: Telephone (Categorical)



Attribute 20: Foreign worker (Categorical)



Παρατηρήσεις

Από τα διαγράμματα παρατηρούμε ότι ορισμένα features είναι πολύ πιο χρήσιμα από άλλα και θα μας βοηθήσουν περισσότερο στην κατηγοριοποίηση των πελατών, καθώς για ορισμένες τιμές τους, ο αριθμός των clients που έχουν χαρακτηριστεί ως Good είναι πολύ μεγαλύτερος από αυτούς που έχουν χαρακτηριστεί Bad.

Χρήσιμα features είναι για παράδειγμα το Attribute1 (Status of existing checking account) καθώς για την τιμή A14 (No checking account) βλέπουμε πως η πλειοψηφία των clients είναι Good, κι έτσι περιμένουμε πως αν στο dataset συναντήσουμε την τιμή A14 στο Attibute1 υπάρχει μεγάλη πιθανότητα ο client να είναι Good καθώς και το Attribute 3 (Credit history) το οποίο εμφανίζει πολύ μεγαλύτερο ποσοστό Good στην τιμή A34 (other credits existing) και μεγαλύτερο ποσοστό Bad στην τιμή A30 (no credits taken)

Άλλα τέτοια features είναι Attribute2 (Duration in months) και το Attribute4 (Purpose).
Στο Box plot του Attribute2 βλέπουμε ότι ο median της διάρκειας για τους Good βρίσκεται γύρω στους 16-17 μήνες ενώ για τους Bad γύρω στους 23-24, δίνοντας μας έτσι αρκέτα διακριτά αποτελέσματα. Αντίστοιχα, στο Histogram Attribute4 παρατηρούμε ότι για τις τιμές A43 (radio/television) και A41 (used car) η πιθανότητα να είναι ο client Good είναι μεγαλύτερη.

Τα features αυτά περιμένουμε στο ερώτημα 3 να έχουν μεγαλύτερο Information Gain score από τα υπόλοιπα. Παρατηρούμε βέβαια ότι υπάρχουν και features που φαίνεται από τα διαγράμματα ότι δε μας δίνουν σχεδόν καθόλου χρήσιμη πληροφορία. Τέτοια είναι το Attribute18 (Number of people being liable to provide maintenance for), το Attribute11 (Present residence since), στο οποίο βλεπουμε ότι ο median στο Box plot σχεδόν συμπίπτει.

2. Classification - Κατηγοριοποίηση Δεδομένων

Σχόλια

Για την κατηγοριοποίηση των δεδομένων, δοκιμάσαμε τρεις μεθόδους Classification: Naive Bayes, Random Forests και Support Vector Machines (SVM).

Αρχικά, στο στάδιο της προεπεξεργαίας δεδομένων, έπρεπε να μετατρέψουμε τα categorical features του dataset σε numerical ώστε να μπορούν να χρησιμοποιηθούν από τους classifiers. Για το σκοπό αυτό, δοκιμάσαμε αρχικά τον DictVectorizer καθώς και τον OneHotEncoder του scikit-learn, αλλά τελικά καταλήξαμε στη συνάρτηση get_dummies() του pandas.

Για κάθε μέθοδο Classification, πειραματιστήκαμε με τις παραμέτρους της ώστε να βρούμε τον συνδυασμό για τον οποίο έχει την καλύτερη απόδοση.

Έτσι για τον Naive Bayes δοκιμάσμε εκτός από GaussianNB και MultinomialNB, που όπως περιμέναμε δε μας έδωσε καλύτερη απόδοση, αφού τα δεδομένα μας δεν έχουν πολυωνυμική κατανομή.

Για τον Random Forests δοκιμάσαμε πολλές διαφορετικές τιμές για τον αριθμό δέντρων στο δάσος (n_estimators) και καταλήξαμε ότι έχει την καλύτερη απόδοση για n_estimators=128. Ενδεικτικά για τα διάφορα n,το accuracy ήταν:

n=2	0.6925	n=100	0.7575
n=10	0.73875	n=120	0.76
n=50	0.74875	n=150	0.7475

Για τον SVM, πειραματιστήκαμε με διαφορετικές τιμές στις παραμέτρους C (penalty of the error term) και gamma (kernel coefficient) και διαπιστώσαμε ότι την καλύτερη απόδοση έχει για C=1.5 gamma = 0.5. Συγκεκριμένα το accuracy από 0.69125 που ήταν για τις default τιμές των παραμέτρων, έφτασε στο 0.70125

Για να αξιολογήσουμε την απόδοση κάθε μεθόδου, χρησιμοποιήσαμε 10-fold Cross Validation με τη μετρική Accuracy.

Αποτελέσματα

Έπειτα από πολλές μετρήσεις καταλήξαμε στο ότι ο Classifier που είχε την καλύτερη απόδοση από όλους είναι ο Random Forests, με το μέσο Accuracy να κυμαίνεται από 0.75 μέχρι 0.77.

Τα αποτελέσματα αυτά υπάρχουν στο αρχείο *EvaluationMetric_10fold.csv*:

Statistic Measure	Naive Bayes	**Random Forest**	SVM
Accuracy	0.7	**0.77375**	0.70125

Τα αποτελέσματα από την κατηγοριοποίηση των πελατών του test dataset με χρήση του RandomForest βρίσκονται στο αρχείο *testSet_Predictions.csv* . Σε συνοποπτική μορφή, τα αποτελέσματα έχουν ως εξής:

ID		ID		ID		ID		ID	
10902	Good	**10922**	Good	**10942**	Good	**10962**	Good	**10982**	Good
10903	Good	**10923**	Good	**10943**	Good	**10963**	Bad	**10983**	Good
10904	Good	**10924**	Good	**10944**	Good	**10964**	Good	**10984**	Good

10905	Good	**10925**	Good	**10945**	Good	**10965**	Good	**10985**	Good
10906	Bad	**10926**	Good	**10946**	Good	**10966**	Good	**10986**	Bad
10907	Good	**10927**	Good	**10947**	Good	**10967**	Good	**10987**	Good
10908	Good	**10928**	Good	**10948**	Good	**10968**	Good	**10988**	Bad
10909	Bad	**10929**	Good	**10949**	Good	**10969**	Good	**10989**	Good
10910	Bad	**10930**	Good	**10950**	Good	**10970**	Bad	**10990**	Good
10911	Good	**10931**	Good	**10951**	Good	**10971**	Good	**10991**	Good
10912	Good	**10932**	Bad	**10952**	Good	**10972**	Good	**10992**	Good
10913	Good	**10933**	Bad	**10953**	Good	**10973**	Good	**10993**	Good
10914	Bad	**10934**	Good	**10954**	Bad	**10974**	Good	**10994**	Good
10915	Bad	**10935**	Good	**10955**	Good	**10975**	Good	**10995**	Good
10916	Bad	**10936**	Bad	**10956**	Good	**10976**	Good	**10996**	Good
10917	Good	**10937**	Good	**10957**	Good	**10977**	Bad	**10997**	Bad
10918	Good	**10938**	Good	**10958**	Good	**10978**	Good	**10998**	Good
10919	Bad	**10939**	Good	**10959**	Good	**10979**	Good	**10999**	Good
10920	Good	**10940**	Good	**10960**	Good	**10980**	Good	**11000**	Good
10921	Good	**10941**	Good	**10961**	Good	**10981**	Good	**11001**	Good

ID		ID		ID		ID		ID	
11002	Good	**11022**	Good	**11042**	Good	**11062**	Good	**11082**	Good
11003	Good	**11023**	Good	**11043**	Good	**11063**	Good	**11083**	Good
11004	Good	**11024**	Good	**11044**	Good	**11064**	Good	**11084**	Good
11005	Good	**11025**	Bad	**11045**	Good	**11065**	Good	**11085**	Good
11006	Good	**11026**	Bad	**11046**	Bad	**11066**	Good	**11086**	Bad
11007	Good	**11027**	Bad	**11047**	Good	**11067**	Good	**11087**	Bad
11008	Good	**11028**	Good	**11048**	Good	**11068**	Good	**11088**	Good
11009	Good	**11029**	Good	**11049**	Good	**11069**	Good	**11089**	Good
11010	Good	**11030**	Bad	**11050**	Good	**11070**	Good	**11090**	Good

11011	Good	**11031**	Good	**11051**	Good	**11071**	Good	**11091**	Good
11012	Good	**11032**	Good	**11052**	Good	**11072**	Good	**11092**	Good
11013	Good	**11033**	Good	**11053**	Good	**11073**	Bad	**11093**	Good
11014	Good	**11034**	Good	**11054**	Bad	**11074**	Bad	**11094**	Good
11015	Bad	**11035**	Good	**11055**	Good	**11075**	Good	**11095**	Good
11016	Bad	**11036**	Bad	**11056**	Good	**11076**	Good	**11096**	Good
11017	Good	**11037**	Good	**11057**	Good	**11077**	Good	**11097**	Good
11018	Good	**11038**	Good	**11058**	Good	**11078**	Good	**11098**	Good
11019	Good	**11039**	Bad	**11059**	Good	**11079**	Good	**11099**	Bad
11020	Good	**11040**	Good	**11060**	Good	**11080**	Bad	**11100**	Good
11021	Good	**11041**	Good	**11061**	Good	**11081**	Good		

3. Features Selection

Αρχικά, για να μπορέσουμε να υπολογίσουμε το Information Gain, έπρεπε να μετατρέψουμε όλα τα δεδομένα σε categorical, κάτι που υλοποιήσαμε χρησιμοποιώντας τη συνάρτηση cut του pandas με 5 bins.

Για τον υπολογισμό της Entropy, χρησιμοποιήσαμε τον τύπο:

```
Hp = -sum(P[i]*log(P[i])
```

όπου P[1] η πιθανότητα ο client να είναι Good και P[2] η πιθανότητα να είναι Bad.
Η βάση του log είναι το e, καθώς είδαμε ότι αυτή χρησιμοποιούν και οι βιβλιοθήκες της Python για τον υπολογισμό της Entropy (πχ η scipy.stats.entropy)

Για τον υπολογισμό του Information Gain, χρησιμοποιήσαμε τον τύπο:

```
IG((T,a) = H_T - sum(wv*H_v)
```

όπου

`H_T = entropy(T)`	η Entropy του dataset
`wv = values[v]/len(T)`	το ποσοστό των rows του dataset όπου το attribute a έχει την τιμή v
`H_v = entropy(T[T[a] == v])`	η Entropy των rows του dataset όπου το attribute a έχει την τιμή v

Για τον έλεγχο των συναρτήσεων που χρησιμοποιήσαμε για να υλοποιήσουμε τους παραπάνω τύπους, δημιουργήσαμε ένα μικρό dataset 20 σειρών (*train_small.csv*) και εκτυπώναμε σε κάθε βήμα τα αποτελέσματα των υπολογισμών ώστε να επαληθεύσουμε την ορθότητα τους.

Στον παρακάτω πίνακα, βλεπουμε το feature που επιλέξαμε να αφαιρούμε σε κάθε επανάληψη καθώς και το αντίστοιχο Information Gain, σε κατάταξη από το μικρότερο στο μεγαλύτερο.

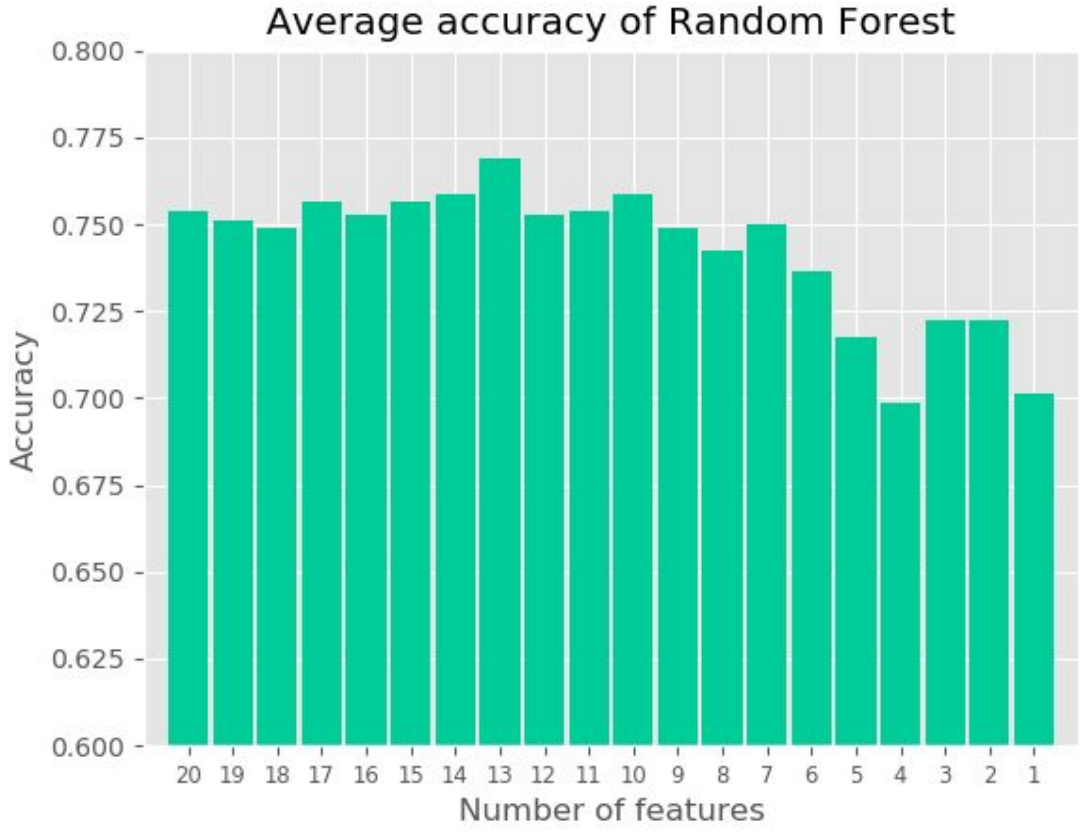
	Information Gain
Attribute18	0.000090
Attribute11	0.000153
Attribute19	0.000834

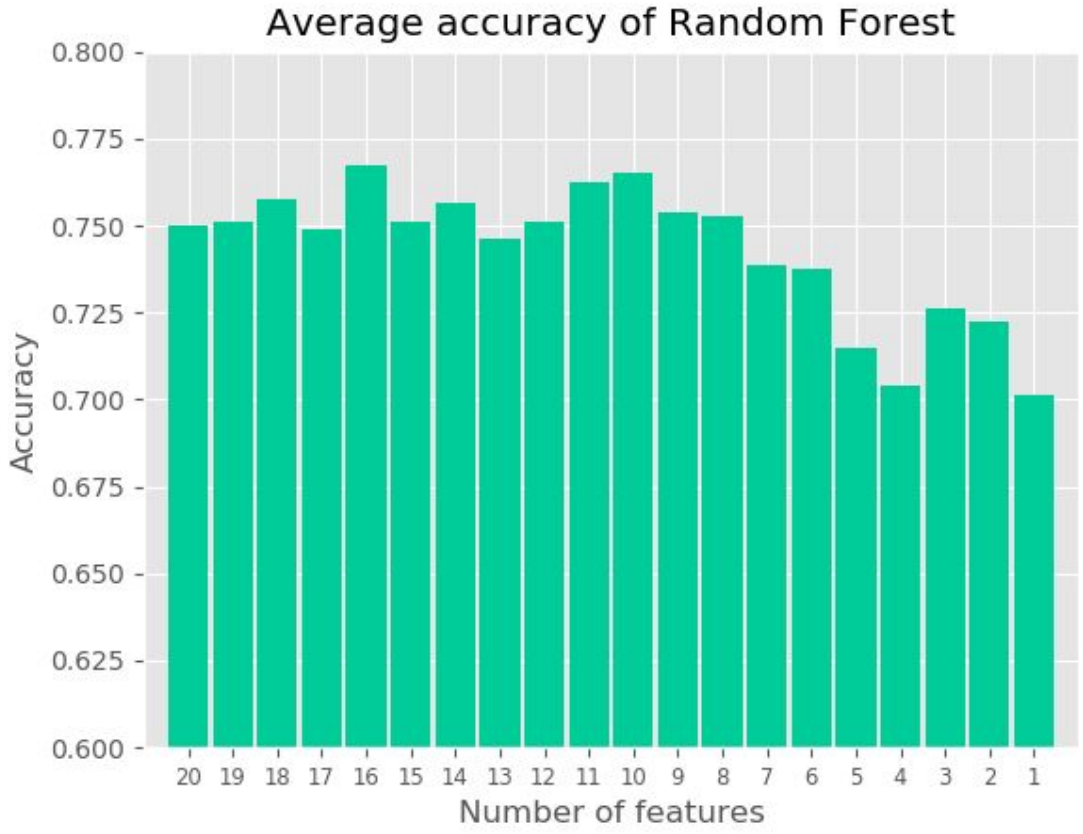
Attribute16	0.001661
Attribute17	0.002038
Attribute10	0.003933
Attribute14	0.004881
Attribute8	0.005081
Attribute20	0.005340
Attribute15	0.008054
Attribute9	0.008835
Attribute13	0.009297
Attribute7	0.010084
Attribute12	0.010332
Attribute5	0.012796
Attribute6	0.015387
Attribute4	0.018644
Attribute2	0.022849
Attribute3	0.026263
Attribute1	0.065037

Παρατηρούμε ότι τα αποτελέσματα του Information Gain συμφωνούν με τις προβλεψεις που κάναμε με βάση τα διαγράμματα στο Ερώτημα 1 σχετικά με το ποιά είναι τα περισσότερο καιποιά τα λιγότερο χρήσιμα features.

Αφαιρόντας ένα-ένα τα features από το dataset με τη σειρά που ορίζεται στον παραπάνω πίνακα, και ξανατρέχοντας τον classifier, θα δημιουρήσουμε ένα plot για να δείξουμε πώς μεταβάλλεται το μέσο accuracy για 10-fold cross-validation.

Για δύο διαφορετικές δοκιμές, έχουμε τα Plot 1 και Plot 2 όπως εμφανίζονται παρακάτω:


Average accuracy of Random Forest
0.800
0.775
0.750
0.725
0.700
0.675
0.650
0.625
0.600
Accuracy
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
Number of features


Average accuracy of Random Forest
0.800
0.775
0.750
0.725
0.700
0.675
0.650
0.625
0.600
Accuracy
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
Number of features

Παρατηρήσεις

Παρατηρούμε ότι και στα δύο plots, το accuracy αρχικά αυξάνεται, φτάνοντας το μέγιστό λίγο πριν τη μέση και στη συνέχεια μειώνεται.

Αυτή ακριβώς είναι και η μορφή που περιμέναμε: στην αρχή, αφαιρώντας features που δε μας έδιναν χρήσιμη πληροφορία,το dataset γίνεται καλύτερο και το accuracy ανεβαίνει. Όταν όμως αφαιρέσουμε πάρα πολλά features, ακόμα κι αν αυτά έχουν υψηλό information gain, δεν επαρκούν για να μας δώσουν όλη την πληροφορία που χρειαζόμαστε για την κατηγοριοποίηση των δεδομένων και το accuracy μειώνεται.

Ακόμη, βλέπουμε πως στο Plot 1, το accuracy ήταν μέγιστο (**0.76875**) μετά την αφαίρεση 7 features, ενώ στο Plot 2 μετά την αφαίρεση 4 features (**0.7675**). Σε άλλες δοκιμές που πραγματοποιήσαμε είδαμε ότι μπορεί έχουμε το μέγιστο accuracy μετά την αφαίρεση 10 (**0.7725**), 9, 5, 3, 2 (**0.7725**) ή και 1 feature.

Τέλος, και στα 2 Plots παρατηρούμε ότι θα μπορούσαμε να χρησιμοποιήσουμε ακριβώς τα μισά features, και το accuracy θα ήταν το ίδιο ή και καλύτερο!

Τα αναλυτικά αποτελέσματα εμφανίζονται στον παρακάτω πίνακα:

Number of Features	Feature dropped	Plot 1: Accuracy of Random Forest	Plot 2: Accuracy of Random Forest
20	-	0.75375	0.75
19	Attribute18	0.75125	0.75125
18	Attribute11	0.74875	0.7575
17	Attribute19	0.75625	0.74875
16	Attribute16	0.7525	**0.7675**
15	Attribute17	0.75625	0.75125
14	Attribute10	0.75875	0.75625
13	Attribute14	**0.76875**	0.74625
12	Attribute8	0.7525	0.75125
11	Attribute20	0.75375	0.7625
10	Attribute15	0.75875	0.765

9	Attribute9	0.74875	0.75375
8	Attribute13	0.7425	0.7525
7	Attribute7	0.75	0.73875
6	Attribute12	0.73625	0.7375
5	Attribute5	0.7175	0.715
4	Attribute6	0.69875	0.70375
3	Attribute4	0.7225	0.72625
2	Attribute2	0.7225	0.7225
1	Attribute3	0.70125	0.70125