ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ
ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ 1
DECISION TREES

Σκοπός της συγκεκριμένης εργασίας είναι η εξοικείωση με την υλοποίηση των αλγορίθμων των Δέντρων Απόφασης (Decision Trees) και των Τυχαίων Δασών (Random Forest), για την επίλυση προβλημάτων Ταξινόμησης (Clasification problems). Τα αποτελέσματα των αλγορίθμων συγκρίθηκαν στο τέλος με αυτά του αλγορίθμου Παλινδρόμησης Logistic Regression.

Η αξιολόγηση του κάθε αλγορίθμου έγινε με χρήση των παρακάτω μετρικών (στον υπολογισμό των μετρικών επιλέχθηκε τιμή παραμέτρου average='binary'):

- **accuracy**: Ορίζεται ως (TP+TN) / (TP+TN+FP+FN). Υπολογίζεται παράλληλα και στα 2 σετ δεδομένων (εκπαίδευσης και επικύρωσης) για να προσδιοριστεί το φαινόμενο underfitting ή overfitting του αλγορίθμου.
- **precision**: Ορίζεται ως TP / (TP+FP)
- **recall**: Ορίζεται ως TP / (TP+FN)
- **f1**: Ορίζεται ως 2*precision*recall / (precision+recall)

Ακολουθήθηκαν οι υποδείξεις της εκφώνησης και των συνοδευτικών templates κώδικα. Η εργασία υλοποιήθηκε σε γλώσσα προγραμματισμού Python.

A. Decision Trees

Οι κρίσιμες παράμετροι που καθορίζουν τη λειτουργία του αλγορίθμου είναι οι:

- **criterion**: Είναι το κριτήριο διαχωρισμού, που χρησιμοποιεί ο αλγόριθμος σε κάθε κόμβο του δέντρου. Οι δυνατές τιμές είναι 'gini' ή 'entropy'.
- **max_depth**: Το μέγιστο επιτρεπτό βάθος ανάπτυξης του δέντρου. Οι δυνατές τιμές είναι από 1 έως και το πλήθος των χαρακτηριστικών (Features). Στην υλοποίηση χρησιμοποίησα αρχική τιμή 3 και αύξανα σταδιακά με βήμα 3.

Πέρα από τις παραπάνω παραμέτρους, που δηλώνονται άμεσα στον αλγόριθμο, κεντρικό ρόλο στην απόδοσή του παίζει και το πλήθος των χαρακτηριστικών (features) των δεδομένων που θα χρησιμοποιηθούν. Το συγκεκριμένο σετ δεδομένων περιέχει 30 συνολικά χαρακτηριστικά. Ξεκίνησα την εξέταση χρησιμοποιώντας τα 10 πρώτα και στη συνέχεια τα αύξανα σταδιακά με βήμα 5.

Έγινε επαναληπτική εκτέλεση του αλγορίθμου με όλους τους δυνατούς συνδυασμούς των παραπάνω παραμέτρων. Ο κώδικας υλοποίησης της επαναληπτικής εκτέλεσης υπάρχει στο συνοδευτικό αρχείο *3Methods_Evaluate.py*. Τα αποτελέσματα φαίνονται στον παρακάτω πίνακα:

Criterion	Features	Max Depth	Training Accuracy	Test Accuracy	Recall	Precission	f1
gini	10	3	0,955	0,937	0,921	0,976	0,948
gini	10	6	0,993	0,916	0,910	0,953	0,931
gini	10	9	1,000	0,902	0,899	0,941	0,920
gini	15	3	0,955	0,930	0,910	0,976	0,942
gini	15	6	0,995	0,923	0,910	0,964	0,936
gini	15	9	1,000	0,909	0,921	0,932	0,927
gini	20	3	0,955	0,930	0,910	0,976	0,942
gini	20	6	0,995	0,951	0,944	0,977	0,960
gini	20	9	1,000	0,902	0,910	0,931	0,920
gini	***25***	***3***	***0,972***	***0,958***	***0,966***	***0,966***	***0,966***

gini	25	6	0,998	0,937	0,921	0,976	0,948
gini	25	9	1,000	0,937	0,933	0,965	0,949
gini	30	3	0,972	0,958	0,978	0,956	0,967
gini	30	6	0,998	0,951	0,955	0,966	0,960
gini	30	9	1,000	0,951	0,955	0,966	0,960
entropy	10	3	0,939	0,909	0,876	0,975	0,923
entropy	10	6	0,977	0,958	0,955	0,977	0,966
entropy	10	9	0,998	0,930	0,933	0,954	0,943
entropy	15	3	0,939	0,902	0,865	0,975	0,917
entropy	15	6	0,986	0,930	0,966	0,925	0,945
entropy	15	9	1,000	0,944	0,955	0,955	0,955
entropy	20	3	0,939	0,902	0,865	0,975	0,917
entropy	***20***	***6***	***0,991***	***0,965***	***0,989***	***0,957***	***0,972***
entropy	20	9	1,000	0,923	0,921	0,953	0,937
entropy	25	3	0,981	0,958	0,978	0,956	0,967
entropy	25	6	0,998	0,958	0,978	0,956	0,967
entropy	25	9	1,000	0,951	0,966	0,956	0,961
entropy	30	3	0,981	0,965	0,989	0,957	0,972
entropy	30	6	0,998	0,958	0,978	0,956	0,967
entropy	30	9	1,000	0,958	0,978	0,956	0,967

Αξιολόγηση: Αναφορικά με την παράμετρο max_depth, παρατήρησα ότι για τιμές μεγαλύτερες του 6 παρατηρείται overfitting του αλγορίθμου, όπως κρίνεται από την τιμή της Training Accuracy, που γίνεται 1, καθώς και από την πτώση στη τιμή της Test Accuracy. Για το λόγο αυτό στον πίνακα φαίνονται οι δοκιμές για τιμές max_depth έως και 9.

Αναφορικά με το πλήθος των χαρακτηριστικών (features) που χρησιμοποιούνται, παρατήρησα ότι η απόδοση του αλγορίθμου, όπως καθορίζεται από την τιμή της Test Accuracy, φτάνει σε υψηλά ποσοστά όταν χρησιμοποιηθούν τουλάχιστον τα πρώτα 20 χαρακτηριστικά του σετ δεδομένων.

Η επιλογή του κριτηρίου (criterion) ‘gini’ για το διαχωρισμό των κόμβων, παρατηρώ ότι οδηγεί τον αλγόριθμο σε overfitting σε μικρότερο max_depth σε σχέση με την επιλογή ‘entropy’. Παρόλα αυτά ελάχιστα επηρεάζεται η τελική απόδοση του αλγορίθμου.

Με βάση τα παραπάνω επιλέγω, για κάθε κριτήριο, τις βέλτιστες συνθήκες του αλγορίθμου, αυτές δηλαδή που μεγιστοποιούν την τιμή της Test Accuracy. Όπως ήταν αναμενόμενο από αυτά που ανάλυσα παραπάνω, παράγεται δέντρο βάθους 3 όταν επιλέγεται ως κριτήριο το ‘gini’ και δέντρο βάθους 6 όταν επιλέγεται ως κριτήριο το ‘entropy’. Παρατηρώ ότι οι συνθήκες αυτές βελτιστοποιούν παράλληλα και τις υπόλοιπες μετρικές που χρησιμοποιήθηκαν. Οι δύο αυτές καταστάσεις σημειώνονται στον πίνακα με διαφορετική μορφοποίηση.

Γραφική αναπαράσταση των Δέντρων: Έγινε γραφική απεικόνιση των 2 δέντρων που παράγονται από τις βέλτιστες συνθήκες του κάθε κριτηρίου. Οι απεικονίσεις φαίνονται στο τέλος της εργασίας, καθώς και σε ξεχωριστά συνοδευτικά .png αρχεία. Ο κώδικας που χρησιμοποιήθηκε για την παραγωγή τους υπάρχει στο συνοδευτικό αρχείο *TreesGraphicRepresentation.py*.

B. Random Forest

Οι κρίσιμες παράμετροι που καθορίζουν τη λειτουργία του αλγορίθμου είναι παρόμοιες με αυτές του Decision Tree, και τις χειρίζομαι με παρόμοιο τρόπο, απλά δεν υπάρχει η max_depth. Η μόνη διαφορετική παράμετρος που χρησιμοποιείται είναι η:

- **estimators**: Καθορίζει το πλήθος των παραγόμενων δέντρων του δάσους. Στην υλοποίηση χρησιμοποίησα αρχική τιμή 5 και αύξανα σταδιακά έως και την τιμή 100.

Έγινε επαναληπτική εκτέλεση του αλγορίθμου με όλους τους δυνατούς συνδυασμούς των παραπάνω παραμέτρων. Ο κώδικας υλοποίησης της επαναληπτικής εκτέλεσης υπάρχει στο συνοδευτικό αρχείο *3Methods_Evaluate.py*. Τα αποτελέσματα φαίνονται στον παρακάτω πίνακα:

Criterion	Features	Estimators	Training Accuracy	Test Accuracy	Recall	Precission	f1
gini	10	5	0,993	0,951	0,944	0,977	0,960
gini	10	10	0,993	0,937	0,921	0,976	0,948
gini	10	15	0,995	0,944	0,933	0,976	0,954
gini	15	5	0,979	0,930	0,910	0,976	0,942
gini	15	10	0,988	0,937	0,921	0,976	0,948
gini	15	15	0,993	0,951	0,944	0,977	0,960
gini	20	5	0,984	0,937	0,955	0,944	0,950
gini	20	10	0,991	0,937	0,921	0,976	0,948
gini	20	15	0,993	0,944	0,933	0,976	0,954
gini	25	5	0,988	0,951	0,933	0,988	0,960
gini	25	10	0,995	0,965	0,955	0,988	0,971
gini	25	15	0,998	0,965	0,955	0,988	0,971
gini	30	5	0,991	0,958	0,955	0,977	0,966
gini	30	10	0,998	0,951	0,921	1,000	0,959
gini	***30***	***15***	***0,995***	***0,972***	***0,966***	***0,989***	***0,977***
entropy	10	5	0,988	0,979	0,966	1,000	0,983
entropy	10	10	0,991	0,965	0,944	1,000	0,971
entropy	10	15	0,995	0,958	0,944	0,988	0,966
entropy	15	5	0,986	0,944	0,933	0,976	0,954
entropy	15	10	0,991	0,951	0,944	0,977	0,960
entropy	15	15	0,991	0,951	0,955	0,966	0,960
entropy	20	5	0,986	0,937	0,966	0,935	0,950
entropy	20	10	0,991	0,923	0,933	0,943	0,938
entropy	20	15	0,998	0,944	0,944	0,966	0,955
entropy	25	5	0,991	0,965	0,978	0,967	0,972
entropy	***25***	***10***	***0,998***	***0,979***	***0,978***	***0,989***	***0,983***
entropy	25	15	0,995	0,979	0,989	0,978	0,983
entropy	30	5	0,995	0,958	0,955	0,977	0,966
entropy	30	10	0,998	0,944	0,933	0,976	0,954
entropy	30	15	0,995	0,965	0,966	0,977	0,972

Αξιολόγηση: Αναφορικά με την παράμετρο estimators, παρατήρησα ότι η απόδοση του αλγορίθμου, όπως κρίνεται από την τιμή της Training Accuracy και της Test Accuracy, μεγιστοποιείται για τιμές μικρότερες του 20. Για το λόγο αυτό στον πίνακα φαίνονται οι δοκιμές για τιμές estimators έως και 15.

Αναφορικά με το πλήθος των χαρακτηριστικών (features) που χρησιμοποιούνται, παρατήρησα ότι η απόδοση του αλγορίθμου, όπως καθορίζεται από την τιμή της Test Accuracy, φτάνει σε υψηλά ποσοστά όταν χρησιμοποιηθούν τουλάχιστον τα πρώτα 25 χαρακτηριστικά του σετ δεδομένων.

Η επιλογή του κριτηρίου (criterion) για το διαχωρισμό των κόμβων ελάχιστα επηρεάζει τη συμπεριφορά και την τελική απόδοση του αλγορίθμου.

Με βάση τα παραπάνω επιλέγω, για κάθε κριτήριο, τις βέλτιστες συνθήκες του αλγορίθμου, αυτές δηλαδή που μεγιστοποιούν την τιμή της Test Accuracy. Παρατηρώ ότι οι συνθήκες αυτές βελτιστοποιούν παράλληλα και τις υπόλοιπες μετρικές που χρησιμοποιήθηκαν. Οι δύο αυτές καταστάσεις σημειώνονται στον πίνακα με διαφορετική μορφοποίηση.

Γραφική αναπαράσταση των Δέντρων: Έγινε γραφική απεικόνιση του πρώτου δέντρου των 2 δασών που παράγονται από τις βέλτιστες συνθήκες του κάθε κριτηρίου. Οι απεικονίσεις φαίνονται στο τέλος της εργασίας, καθώς και σε ξεχωριστά συνοδευτικά .png αρχεία. Ο κώδικας που χρησιμοποιήθηκε για την παραγωγή τους υπάρχει στο συνοδευτικό αρχείο *TreesGraphicRepresentation.py*.

Γ. Logistic Regression

Η κρίσιμη παράμετρος που καθορίζει τη λειτουργία του αλγορίθμου είναι η:

- **C**: Παίρνει θετικές τιμές και καθορίζει αντιστρόφως ανάλογα το βαθμό "χαλαρότητας" (regularization) στην εκπαίδευση του μοντέλου. Θέλει προσοχή στη ρύθμιση, καθώς μικρές τιμές οδηγούν σε underfitting και μεγάλες τιμές σε overfitting του μοντέλου. Στην υλοποίηση χρησιμοποίησα τιμές από 0,001 έως και 100.

Έγινε επαναληπτική εκτέλεση του αλγορίθμου με διάφορες τιμές της παραμέτρου C. Ο κώδικας υλοποίησης της επαναληπτικής εκτέλεσης υπάρχει στο συνοδευτικό αρχείο *3Methods_Evaluate.py*. Τα αποτελέσματα φαίνονται στον παρακάτω πίνακα:

C	Training Accuracy	Test Accuracy	Recall	Precission	f1
0,001	0,918	0,965	1,000	0,947	0,973
0,01	0,925	0,958	0,989	0,946	0,967
1	0,958	0,958	0,978	0,956	0,967
5	***0,965***	***0,965***	***0,978***	***0,967***	***0,972***
10	0,974	0,958	0,966	0,966	0,966
25	0,974	0,958	0,966	0,966	0,966
50	0,974	0,958	0,966	0,966	0,966
100	0,977	0,958	0,966	0,966	0,966

Αξιολόγηση: Με βάση τα στοιχεία του πίνακα επιλέγω, τις βέλτιστες συνθήκες του αλγορίθμου, αυτές δηλαδή που μεγιστοποιούν την τιμή της Test Accuracy. Παρατηρώ ότι οι συνθήκες αυτές βελτιστοποιούν παράλληλα και τις υπόλοιπες μετρικές που χρησιμοποιήθηκαν. Η κατάσταση αυτή σημειώνεται στον πίνακα με διαφορετική μορφοποίηση.

ΣΥΓΚΡΙΣΗ ΑΠΟΔΟΣΗΣ ΑΛΓΟΡΙΘΜΩΝ

Παρατηρώ ότι, ανάμεσα στους αλγορίθμους Decision Tree και Random Forest, την καλύτερη απόδοση, όπως την αποδίδει το Test Accuracy, παρουσιάζει όπως είναι αναμενόμενο ο Random Forest, καθώς εμφανίζει σταθερά καλύτερες τιμές (0.972 vs 0.958 και 0.979 vs 0.965), ανεξάρτητα από την τιμή που επιλέγεται για το criterion. Συγκριτικά, ο αλγόριθμος Logistic Regression αποδίδει εξίσου καλά με τον Decision Tree, με απόδοση που φτάνει το 0.965.

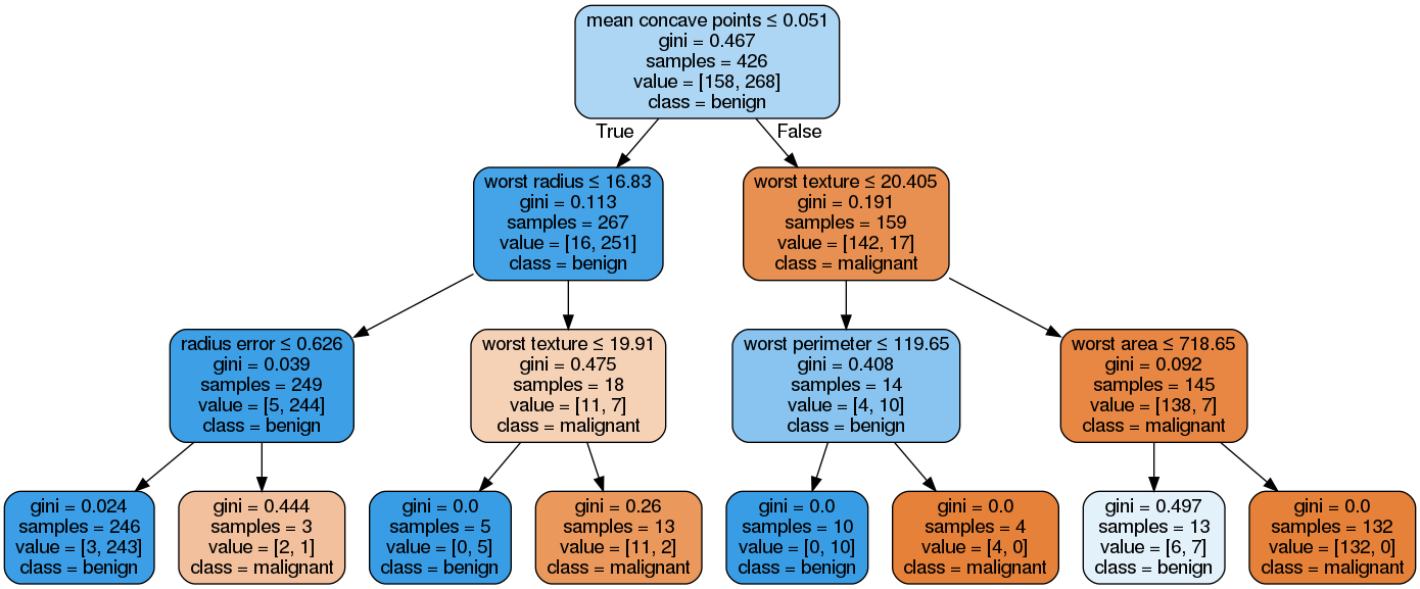
ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Υλικό μαθήματος

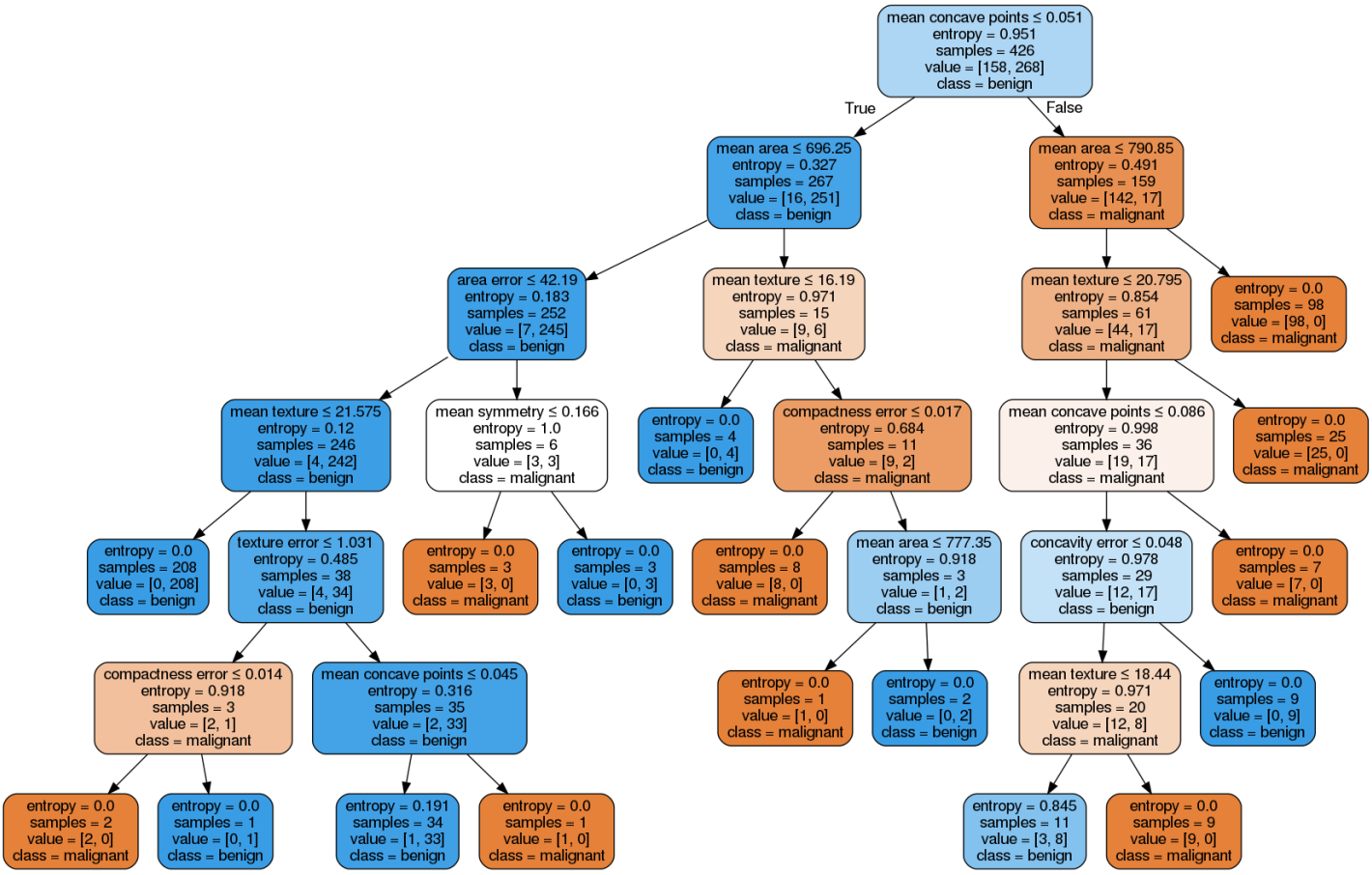
2. Müller, A.C & Guido, S. - Introduction to Machine Learning with Python. A Guide for Data Scientists, O' Reilly, 2017

ΓΡΑΦΙΚΗ ΑΠΕΚΟΝΙΣΗ ΜΟΝΤΕΛΩΝ

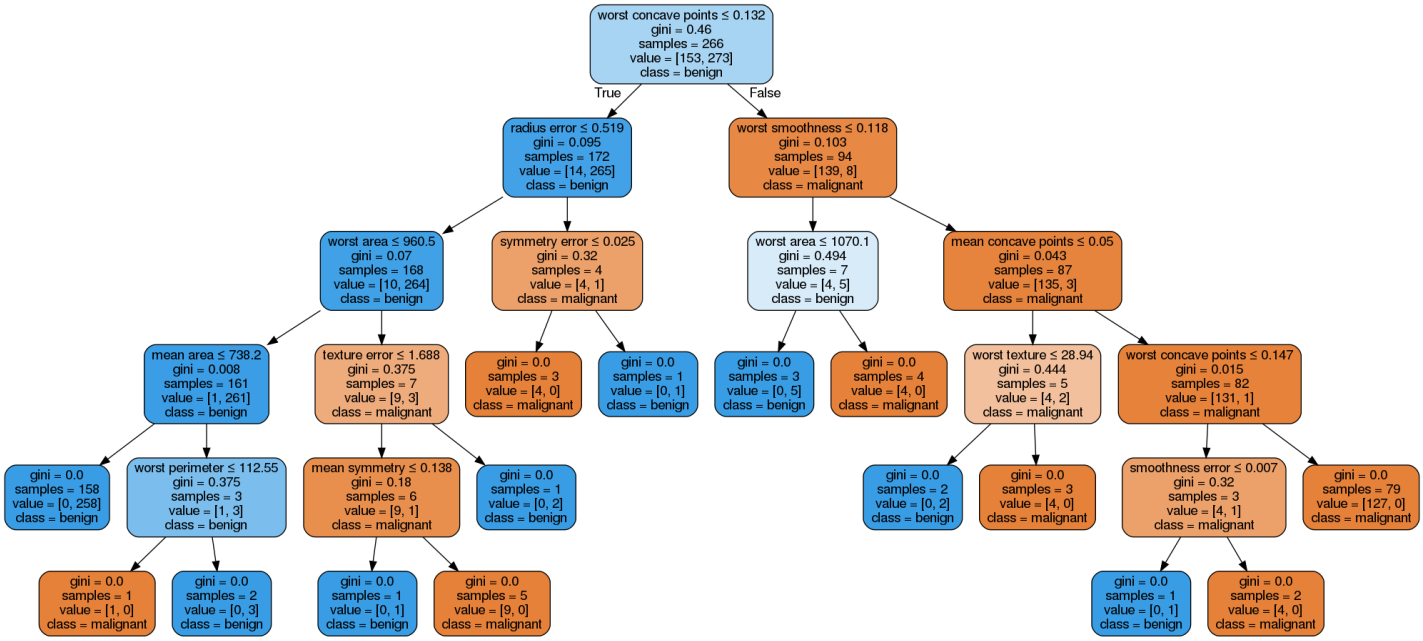
1. Decision Tree, criterion: “gini”, features: 25, max_depth: 3



2. Decision Tree, criterion: “entropy”, features: 20, max_depth: 6



3. Random Forest, criterion: “gini”, features: 30, estimators: 15



4. Random Forest, criterion: “entropy”, features: 25, estimators: 10