Εξόρυξη και Προετοιμασία Δεδομένων

Μεταπτυχιακή απαλλακτική εργασία ΦΕΒ. 2023-24

Καθηγητής:
Μ. Χαλκίδη

Καμπάσης Παναγιώτης
Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Μεγάλα Δεδομένα και Αναλυτική
22 Ιανουαρίου 2024

Εξόρυξη και Προετοιμασία Δεδομένων

Πάνος Καμπάσης
27/01/2024

Περιεχόμενα

I Εισαγωγή 2
I-Α′ Η βάση δεδομένων . . . 2

II Η διαδικασία σε βήματα 2

III Προ επεξεργασία 2

IV Μερικά Στατιστικά 2
IV-Α′ Λογιστική παλινδρόμηση χρησιμοποιώντας την μέθοδο $TF - IDF$ . . . 2
IV-Β′ *Support vector classification* χρησιμοποιώντας την μέθοδο $TF - IDF$ . . . 2
IV-Γ′ Εξετάζουμε την βάση στην ολότητα της . . . 3

V Μετατροπή των *unsup* τιμών με τα δικά μας μοντέλα κατηγοριοποίησης 3
V-Α′ Λογιστική Παλινδρόμηση . . . 3
V-Β′ $Support\,Vector\,Classification$ . . . 3

VI $Word2Vec\,Vectorizer$ 3
VI-Α′ Λόγοι για τους οποίους συγκρίνουμε τις δύο τεχνικές . . . 3
VI-Β′ Λογιστική παλινδρόμηση . . . 4
VI-Γ′ Η τεχνική χρησιμοποιώντας ολόκληρη τη βάση δεδομένων . . . 4

VII Δοκιμή 3 ακόμα μοντέλων και σύγκριση αποτελεσμάτων 4
VII-Α′ $Random\,Forest$, $Decision\,Tree$, $Gradient\,Boosting$ . . . 4
VII-Β′ Ο τρόπος εκπαίδευσης των μοντέλων . . . 5

VIII Ανακεφαλαίωση και Συμπεράσματα 6

Περίληψη—
Στο παρόν θα επιχειρήσουμε να εξετάσουμε την προβλεπτική ικανότητα καθώς και την κατηγοριοποίηση σε 2 κλάσεις (θετικό και αρνητικό) αντλώντας από σχόλια τύπου κριτικής. Σκοπός είναι η εύρεση του αποδοτικότερου μοντέλου καθώς και ο πειραματισμός με τεχνικές που δημιουργούν διανύσματα από λέξης και σύγκριση αποτελεσμάτων των μοντέλων μηχανικής μάθησης που θα εξετάσουμε.
Λέξεις κλειδιά:
Σχόλια $IMDB$, Μέθοδος $TF-IDF$, Μηχανική Μάθηση, Μέθοδος $Word2Vec$,$classification$, Λογιστική Παλινδρόμηση, $Random\,Forest$, $Decision\,Tree$, $Gradient\,Boosting$.

I. Εισαγωγή

Αναπτύσσοντας με την βοήθεια της *python* και συγκεκριμένα των βιβλιοθηκών *ntlk*, *sklearn*, *gensim* αλγορίθμους $Word2Vec$ και $TF-IDF$ επιχειρούμε να μετατρέψουμε όλες τις λέξεις του εκάστοτε σχόλιου σε διανύσματα με σκοπό την εκπαίδευση του μοντέλου στην αναγνώριση της κατηγορίας του σχόλιου και την εξαγωγή δεδομένων από αυτό ούτως ώστε να εφαρμόσουμε αλγορίθμους κατηγοριοποίησης ($classification$) για την πρόβλεψη της κριτικής σε 2 κατηγορίες (pos,neg), ώστε να επιτύχουμε μια τεχνική ανίχνευσης συναισθήματος. Θα εξετάσουμε 2 μοντέλα με 2 μεθόδους διανυσματοποίησης. Πρώτα $TF-IDF$ με λογιστική και SVC και έπειτα τα ίδια με $Word2Vec$. Διαλέγοντας την αποδοτικότερη θα κάνουμε κατηγοριοποίηση στις *unsup* τιμές και θα επιλέξουμε άλλα 4 μοντέλα στη νέα βάση δεδομένων συγκρίνοντας τα αποτελέσματα τους.

Α΄. Η βάση δεδομένων

Αντλούμε δεδομένα από $Kaggle$ και συγκεκριμένα την $IMDB\,Review\,Dataset$. Αποτελείται από δεδομένα από 100.000 σχόλια σε ταινίες όπου προ-υπάρχει ο διαχωρισμός σε *test* και *train* καθώς και μια κατηγοριοποίηση σε *pos*,*neg* και *unsup*.

II. Η διαδικασία σε βήματα

- Αρχικά εφαρμόζουμε 2 μεθόδους για την μετατροπή των λέξεων σε διανύσματα.
- Έπειτα προσμετρούμε την επίδοση από δύο μοντέλα μηχανικής μάθησης.
- Η βάση δεδομένων εμπεριέχει την κατηγορία *unsup* των σχολίων που δεν έχει αποφασιστεί η κατηγορία τους. Αρχικά επιχειρούμε να θεωρήσουμε αυτά τα δεδομένα ως $missing\,values$ για να δούμε τι στατιστικά μπορούμε να αντλήσουμε από αυτό

III. Προ επεξεργασία

Όταν έχουμε βάσεις δεδομένων με κείμενο υπάρχει μια σειρά καλών πρακτικών που μπορούμε να εφαρμόσουμε ούτως ώστε να βοηθήσουμε στην καλύτερη ανάλυση των κειμένων από τους αλγορίθμους και τα μοντέλα μας. Θα παραθέσουμε μερικά από αυτά τα βήματα αλλά δεν θα τα υλοποιήσουμε εις βάθος καθώς με την χρήση της μεθόδου $TF-IDF$ πραγματοποιείται $Tokenization$ και αφαιρούνται τα *stopwords* που δεν προσδίδουν ¨νόημα' στην ανάλυση. Στον τομέα λοιπόν της προεπεξεργασίας θα επεξεργαζόμασταν το κείμενο αφαιρώντας τα *stopwords* και αμέσως μετά θα κάναμε $Tokenization$. Αυτά θα βοηθούσαν στην καλύτερη εκπαίδευση των μοντέλων μηχανικής μάθησης που θα αναπτύξουμε στην πορεία. Ωστόσο θα κάνουμε μια μικρή αναφορά στον τρόπο που υλοποιούμε τις παραπάνω τεχνικές καθώς η μέθοδος $Word2Vec$ δεν τις πραγματοποιεί. Τέλος ένα ακόμη βήμα θα ήταν το *steming* όπου αφαιρώντας τις καταλήξεις και την γραμματική και κρατώντας μόνο την ρίζα κάθε λέξης θα βελτιώναμε τον χρόνο εκπαίδευσης των μοντέλων καθώς μικρότερο πλήθος θα χρειαζόταν να μετατραπεί σε διανύσματα. Δεν θα επεκταθούμε όμως σε αυτό το βήμα καθώς, στην παρούσα ανάλυση θα επικεντρωθούμε στην απόδοση των μοντέλων μηχανικής μάθησης. Στο τελευταίο κεφάλαιο συζητούμε λίγο παραπάνω την προεπεξεργασία.

IV. Μερικά Στατιστικά

Α΄. Λογιστική παλινδρόμηση χρησιμοποιώντας την μέθοδο $TF-IDF$

Από δείγμα περίπου 100.000 σχολίων μετατρέπουμε τις λέξεις σε διανύσματα με την μέθοδο $TF-IDF$ και έπειτα εκπαιδεύουμε ένα απλό μοντέλο λογιστικής παλινδρόμησης. Η ακρίβεια που πετυχαίνουμε ανέρχεται στο 87%.Προσμετρούμε επίσης και τα παρακάτω:

	$precision$	$recall$	$f1-score$	$support$
$Negative$	0.88	0.86	0.87	5055
$Positive$	0.86	0.88	0.87	4945
$accuracy$			0.87	10000
$macro\,avg$	0.87	0.87	0.87	10000
$weighted\,avg$	0.87	0.87	0.87	10000

$$MeanSquaredError(MSE) : 0.13$$

$$RootMeanSquaredError(RMSE) : 0.36$$

$$MeanAbsoluteError(MAE) : 0.13$$

$$R-squared(R^2) : 0.48$$

Β΄. $Support\,vector\,classification$ χρησιμοποιώντας την μέθοδο $TF-IDF$

Με την μέθοδο $Support\,vector\,classification$ επιτυγχάνεται ακρίβεια 88%

	$precision$	$recall$	$f1-score$	$support$
$Negative$	0.88	0.87	0.88	5055
$Positive$	0.87	0.88	0.88	4945
$accuracy$			0.88	10000
$macro\,avg$	0.88	0.88	0.88	10000
$weighted\,avg$	0.88	0.88	0.88	10000

$$MeanSquaredError(MSE) : 0.12$$

$$RootMeanSquaredError(RMSE) : 0.35$$

$$MeanAbsoluteError(MAE) : 0.12$$

$$R-squared(R^2) : 0.50$$

Γ′. Εξετάζουμε την βάση στην ολότητα της

Τα χαρακτηριστικά με την ετικέτα *unsup* δημιουργούν προβληματισμό. Εξετάζουμε κατά πόσο επηρεάζει τα μοντέλα μας η περίληψη τους. Όπως είναι φυσικό θα δημιουργήσουμε και μια τρίτη κατηγορία για αυτά. Εκπαιδεύοντας και πάλι τον αλγόριθμο έχουμε:

Class	*precision*	*recall*	$f1-score$	*support*
Negative	0.50	0.30	0.37	5004
Neutral	0.51	0.69	0.58	9965
Positive	0.51	0.36	0.42	5031

και επίσης:

$$MeanSquaredError(MSE) : 0.51$$

$$RootMeanSquaredError(RMSE) : 0.71$$

$$MeanAbsoluteError(MAE) : 0.50$$

$$Accuracy : 0.51$$

V. Μετατροπή των *unsup* τιμών με τα δικά μας μοντέλα κατηγοριοποίησης

Α′. Λογιστική Παλινδρόμηση

Αφού εκπαιδεύσουμε μια φορά το μοντέλο μας με όλα τα δεδομένα εκτός από τις τιμές *unsup*, δίνουμε έπειτα τα σχόλια εκείνων ακριβώς των δεδομένων ώστε να τα κατηγοριοποιήσει. Αυτό αντικαθιστά τις τιμές *unsup* με *pos* και *neg* (αντίστοιχα 1 και 0 για την εκπαίδευση). Έχοντας προσδώσει νέες τιμές σε κάθε σχόλιο αντικαθιστώντας το *unsup* εκπαιδεύουμε εκ νέου το μοντέλο και ελέγχουμε το αποτέλεσμα:

Class	*precision*	*recall*	$f1-score$	*support*
Negative	0.93	0.92	0.92	9804
Positive	0.92	0.93	0.93	10196
accuracy			0.93	20000
macro avg	0.93	0.93	0.93	20000
weighted avg	0.93	0.93	0.93	20000

Παρατηρούμε ότι πετύχαμε καλύτερα αποτελέσματα από όλες τις προηγούμενες προσπάθειες με ακρίβεια 93%. Επιπρόσθετα μετράμε και τα παρακάτω:

$$MeanSquaredError(MSE) : 0.07$$

$$RootMeanSquaredError(RMSE) : 0.27$$

$$MeanAbsoluteError(MAE) : 0.07$$

$$R-squared(R^2) : 0.71$$

Τα ποσοστά που πετυχαίνουμε είναι πολύ αποδοτικά, πράγμα που ενδεχομένως ήταν αναμενόμενο γιατί αυξήσαμε τα δεδομένα μας κατά 50.000 επομένως υπήρχε μεγαλύτερη ευκαιρία στην εκπαίδευση του μοντέλου. Αυτό φυσικά δεν αποτελεί πάντα εγγύηση καθώς ανάλογα με την ποιότητα και το πόσο ¨καθαρά' είναι τα δεδομένα μας δύναται το ποσοστό ακρίβειας να πέσει. Επομένως εδώ αποδεικνύεται η καθαρή φύση της βάσης δεδομένων. Σχολιάζουμε επίσης πως το ποσοστό επιτυχίας μας, αν και ενθαρρυντικό ίσως δεν αντεπεξέρχεται πλήρως καθώς χρησιμοποιήσαμε ως εκπαίδευση σχόλια απροσδιόριστου χαρακτήρα και μάλιστα προσδώσαμε εμείς την θετική η αρνητική τους φύση βάσει του πρώτου μοντέλου μας. Εκεί αναμένετε να υπάρχει κάποιο εύρος σφάλματος. Ωστόσο Επιλέγουμε να το προσθέσουμε καθώς το σφάλμα αυτό δεν επηρεάζει σε επικίνδυνο βαθμό τα δεδομένα μας καθώς η εκπαίδευση του πρώτου μοντέλου πέτυχε ήδη καλό ποσοστό κρίνοντας από των πρώτο πίνακα και τις μετρικές του.

Β′. *Support Vector Classification*

Αν και στο παρόν πρόβλημα θα προτιμήσουμε την λογαριθμική παλινδρόμηση δίνουμε και τις μετρικές του *Support Vector Classification*. Προτιμούμε την λογαριθμική παλινδρόμηση καθώς είναι ένας πολύ πιο γρήγορος αλγόριθμος σε αντίθεση με τον *Support Vector Classification*. Πρακτικά ο δεύτερος είναι καλύτερος αλλά τα υψηλά ποσοστά του πρώτου μας κάνει να προτιμούμε το ταχύτερο μοντέλο. Ο χρόνος που πήρε η εκπαίδευση του *Support Vector Classification* με τα 100.000 σχόλια είναι περίπου 30 λεπτά και δίνει:

$$MeanSquaredError(MSE) : 0.08$$

$$RootMeanSquaredError(RMSE) : 0.29$$

$$MeanAbsoluteError(MAE) : 0.08$$

$$R-squared(R2) : 0.67$$

Class	*precision*	*recall*	$f1-score$	*support*
Negative	0.92	0.91	0.92	4995
Positive	0.91	0.93	0.92	5005

VI. *Word2Vec Vectorizer*

Επιχειρούμε να εξετάσουμε την επίδοση τον παραπάνω αλλά και άλλων μοντέλων χρησιμοποιώντας διαφορετικό τρόπο παραγωγής διανυσμάτων. Παραπάνω πειραματιστήκαμε με την τεχνική $Tf-Idf$ και θα δοκιμάσουμε την *Word2Vec*. Πριν από την χρήση της μεθόδου θα εφαρμόσουμε τον δικό μας κώδικα για *tokenization* και *stopword remove*.

Α′. Λόγοι για τους οποίους συγκρίνουμε τις δύο τεχνικές

Η μέθοδος $Tf-Idf$:

- Καθώς η $Tf-Idf$ παράγει διανύσματα των οποίων η διάσταση απευθύνεται σε ένα μοναδικό όρο έχουμε καλύτερη απόδοση όταν έχουμε πολλούς μοναδικούς όρους.
  Ως εκ τούτου ίσως αναμένουμε καλύτερα αποτελέσματα με $Tf-Idf$ αφού είναι λογικό τα σχόλια για διαφορετικές ταινίες να εμπεριέχουν πολλές διαφορετικές και ίσως μοναδικές λέξεις.
- Η μέθοδος βοηθά στην καλύτερη και αμεσότερη κατανόηση του σχολίου λόγο τις μοναδικότητας των

διανυσμάτων επομένως επιφέρει γρηγορότερη εκπαίδευση

Η μέθοδος $Word2Vec$:

- Αποδίδει μια συνεκτικότερη εικόνα για το σχόλιο καθώς εκπαιδεύεται με βάσει γειτονικές λέξεις (και έπειτα διανύσματα) και έτσι εκτιμάται περισσότερο το ¨νόημα' μιας πρότασης υπό την άποψη της συχνότητας εμφάνισης συγκεκριμένης σειράς λέξεων.
- Δοκιμάζουμε την επίδοση της μεθόδου καθώς πολλά σχόλια δύναται να περιγράφουν το ποιόν μιας ταινίας περιγράφοντας το περιεχόμενο της και έτσι αξίζει να αποτιμήσουμε το νόημα της κάθε πρότασης.

Καθώς για συγκεκριμένες παραμέτρους της μεθόδου $Word2Vec$ ο αλγόριθμος αργεί σημαντικά να δώσει αποτέλεσμα επιλέγουμε να δώσουμε μικρή διάσταση στα διανύσματα ($Word2Vec$). Το κάναμε αυτό για να πετύχουμε ανάλογους χρόνους επίτευξης αποτελέσματος με την $Tf - Idf$. Σαφώς η μικρής διάστασης διάνυσμα μειώνει σημαντικά την επίδοση του μοντέλου αλλά θα δούμε παρακάτω πως η απόδοση του είναι εντυπωσιακή για διάσταση διανύσματος 50 ενώ η προτεινόμενη στην βιβλιογραφία διάσταση είναι 100 για δεδομένα μικρότερα το πλήθος από 10.000 και 200 για 100.000.

Β′. Λογιστική παλινδρόμηση

Τα αποτελέσματα που έχουμε είναι:

$$Word2VecEmbeddings - LogisticRegression:$$

$$MeanSquaredError(MSE) : 0.51$$

$$RootMeanSquaredError(RMSE) : 0.71$$

$$MeanAbsoluteError(MAE) : 0.51$$

$$Word2VecEmbeddings - LogisticRegression:$$

$$Accuracy : 0.49$$

Και μετά την πρόσθεση των *unsup*:

Class	*precision*	*recall*	$f1 - score$	*support*
Negative	0.86	0.83	0.85	5055
Positive	0.84	0.86	0.85	4945

$$MeanSquaredError(MSE) : 0.25$$

$$RootMeanSquaredError(RMSE) : 0.50$$

$$MeanAbsoluteError(MAE) : 0.25$$

$$Word2VecEmbeddings - LogisticRegression:$$

$$Accuracy : 0.75$$

Class	*precision*	*recall*	$f1 - score$	*support*
Negative	1.00	0.00	0.00	5004
Positive	0.75	1.00	0.86	14996

Πράγματι, επιτυγχάνουμε σημαντικά αποτελέσματα παρά τους περιορισμούς, πράγμα το οποίο σχεδόν εγγυάται οτι η μέθοδος $Word2Vec$ είναι αποδοτικότερη, αλλά και ταυτόχρονα έχει μεγάλο κόστος σε χρόνο και πόρους.

Γ′. Η τεχνική χρησιμοποιώντας ολόκληρη τη βάση δεδομένων

Αφήνοντας το αλγόριθμο για περίπου 25 λεπτά επιφέρει αποτέλεσμα ακρίβειας 89%. Έτσι ως τώρα τείνουμε ελαφρώς προς την μέθοδο $Tf - Idf$.

VII. Δοκιμή 3 ακόμα μοντέλων και σύγκριση αποτελεσμάτων

Στο σημείο αυτό θα συγκρίνουμε με πιο περιληπτικό τρόπο 3 ακόμα μοντέλα για να εξάγουμε ποιο μας δίνει καλύτερα αποτελέσματα χρησιμοποιώντας και τις δύο μεθόδους διανυσματοποίησης.

Α′. *Random Forest*, *Decision Tree*, *Gradient Boosting*

Classifier	*Precision*	*Recall*	$F1 - Score$	*Support*
RandomForest	0.86	0.86	0.86	10196
DecisionTree	0.74	0.74	0.74	10196
GradientBoosting	0.81	0.88	0.85	10196

Πίνακας I: $ClassificationReports$

$$RandomForest:$$

$$MeanSquaredError(MSE) : 0.14$$

$$RootMeanSquaredError(RMSE) : 0.38$$

$$MeanAbsoluteError(MAE) : 0.14$$

$$R - squared(R2) : 0.43$$

$$DecisionTree:$$

$$MeanSquaredError(MSE) : 0.26$$

$$RootMeanSquaredError(RMSE) : 0.51$$

$$MeanAbsoluteError(MAE) : 0.26$$

$$R - squared(R2) : -0.05$$

$$GradientBoosting:$$

$$MeanSquaredError(MSE) : 0.16$$

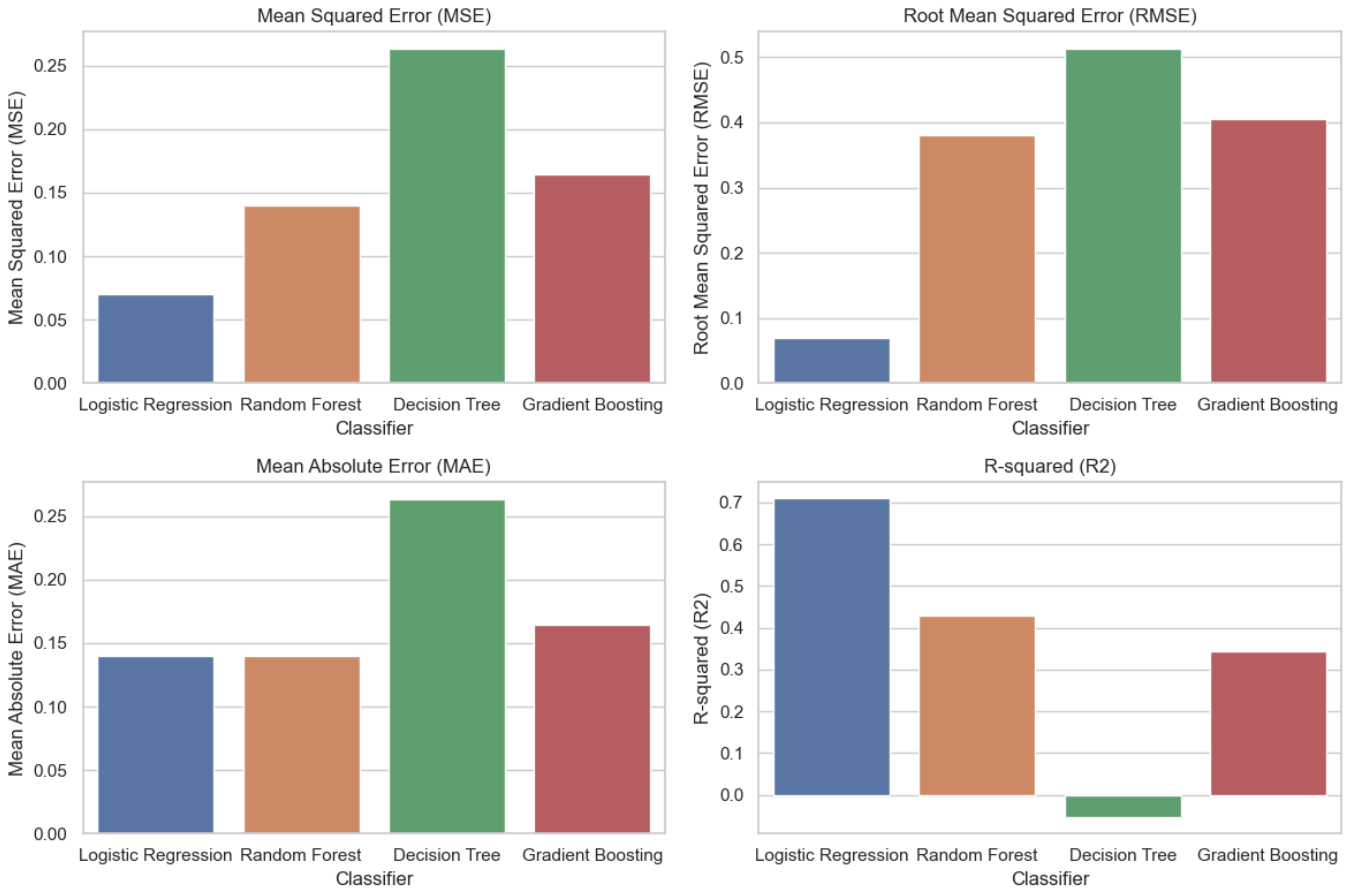
$$RootMeanSquaredError(RMSE) : 0.41$$

$$MeanAbsoluteError(MAE) : 0.16$$

$$R - squared(R2) : 0.34$$

Β΄. Ο τρόπος εκπαίδευσης των μοντέλων

Τα παραπάνω μοντέλα ακολούθησαν την εξής διαδικασία. Εφόσον, σύμφωνα με τις παραπάνω μελέτες η λογιστική παλινδρόμηση δίνει πολύ καλά ποσοστά απόδοσης σε πολύ λίγο χρόνο επιλέγουμε εκείνη για να κατηγοριοποιήσουμε τα δεδομένα που περιέχουν την τιμή *unsup*. Έτσι κατασκευάζουμε την βάση δεδομένων με λογιστική και μετέπειτα εκπαιδεύουμε τα *Random Forest*, *Decision Tree*, *Gradient Boosting*. Σαν τελευταία δοκιμή επιβεβαίωσης εκπαιδεύσαμε το ενδιάμεσο βήμα (δηλαδή τον κατηγοριοποιητή που δίνει τιμές στα *unsup*) και με την μέθοδο *Random Forest*, έναντι της *Logistic Regression* απλώς για να δοκιμάσουμε τον συνδυασμό και επειδή το πρώτο πετυχαίνει τα καλύτερα ποσοστά σύμφωνα με τον παρακάτω πίνακα (δηλάδη στον παρακάτω πίνακα βλέπουμε ότι η *Random Forest* είναι η αμέσως επόμενη καλύτερη μας επιλογή και έτσι δοκιμάσαμε την κατηγοριοποίηση των *unsup* με *Random Forest*. Εφόσον δεν βρήκαμε καλύτερα ποσοστά κρατήσαμε την λογιστική).Έπειτα υλοποιήσαμε ξανά τα 4 μοντέλα αλλά τα αποτελέσματα δεν ήταν καλύτερα σε *Logistic Regression* , *Random Forest* και *Gradient Boosting*, ωστόσο το *Decision Tree* πέτυχε καλύτερα ποσοστά με ακρίβεια 80% αντί του 74% με την λογιστική.
Στο παρακάνω σχήμα έχουμε συγκεντρωτικά τα αποτελέσματα των παραπάνω μοντέλων, τα οποία εκπαιδεύτηκαν με την λογιστική παλινδρόμηση για την επισύναψη των τιμών *unsup*. Όλοι οι συνδυασμοί των μοντέλων εκπαιδεύτηκαν με διανύσματα που έδωσε η μέθοδος $Tf - Idf$:



Σχήμα 1: Μετρικές κάθε μοντέλου. Στο παραπάνω και για την μέτρηση της επίδοσης ενός μοντέλο κατηγοριοποίησης δεν είναι πάντοτε αποδοτικό να μετρούμε το R^2. Ο συντελεστής συσχέτισης βοηθάει στη διασαφήνιση της απόδοσης μοντέλων πανιδρόμησης. Η αρνητική τιμή που παίρνει στο *Decision Tree* είναι σαφώς άτοπη, απλά το αποτυπώνουμε καθώς τα υπολογίζαμε και για τα υπόλοιπα μοντέλα

VIII. Ανακεφαλαίωση και Συμπεράσματα

Υπενθυμίζουμε ότι αυτή είναι η καλύτερη δυνατή ανάλυση που πετύχαμε για τα συγκεκριμένα μοντέλα και επιτεύχθει μόνον επειδή βασιστήκαμε στην αντικατάσταση των τιμών $unsup$ που άλλωστε αποτελούσαν τη μισή βάση δεδομένων. Αν και αυτό αποτελεί σημαντικό παράγοντα κλονισμού της επιτυχίας αυτής της έρευνας σημειώνεται ότι το μοντέλο που έκανε την κατηγοριοποίηση των $unsup$ εκπαιδεύτηκε και απέδωσε ποσοστό ακρίβειας 87% επομένως καταλήγουμε ότι η κατηγοριοποίηση ήταν αρκετά ασφαλής. Από το παραπάνω σχεδιάγραμμα βλέπουμε πως με $Tf - Idf$, η Λογιστική παλινδρόμηση + Λογιστική παλινδρόμηση είναι η καλύτερη μέθοδος με την αμέσως ακόλουθη Λογιστική παλινδρόμηση + $Random\ Forest$. Η $Word2Vec$ αποδίδει καλά αποτελέσματα αλλά λόγο της απλότητας της βάσεως δεδομένων και του γεγονότος ότι η $Tf - Idf$ εκτελέστηκε γρηγορότερα καταλήξαμε στην δεύτερη.
Μελλοντική μελέτη εμπλουτισμού της παραπάνω ανάλυσης είναι η εξαγωγή του $polarity$ των σχολίων και στην συνέχεια η κατηγοριοποίηση τους. Ωστόσο στο ήδη υψηλό ποσοστό του 93% η παραπάνω μέθοδος έχει εξαιρετική πιθανότητα να υποπέσει σε $overfit$ και έτσι θα χρειαζόταν μικρορύθμιση οι διάφοροι παράμετροι των μοντέλων.