

Μερος Γ΄

build_vocab.py:

Ο κώδικας `build_vocab.py` δημιουργεί ένα λεξιλόγιο για την επεξεργασία φυσικής γλώσσας χρησιμοποιώντας το σύνολο δεδομένων IMDB. Φιλτράρει το λεξιλόγιο βάσει συχνότητας εμφάνισης των λέξεων και το αποθηκεύει για μελλοντική χρήση. Ο κώδικας διαβάζει το σύνολο εκπαίδευσης, εφαρμόζει `tokenization`, μετράει τις εμφανίσεις κάθε λέξης και φιλτράρει τις πιο συχνές και λιγότερο συχνές λέξεις, διατηρώντας ένα σύνολο από επιλεγμένες λέξεις.

Περιλαμβάνει τρεις βασικές υπερπαραμέτρους (n , k , m) που επηρεάζουν τη διαδικασία δημιουργίας του λεξιλογίου. Αυτές οι παράμετροι επιλέγονται με βάση εμπειρικές γνώσεις και πρακτικές από την επεξεργασία φυσικής γλώσσας και την ανάλυση του συνόλου δεδομένων IMDB. Η παράμετρος $n = 5000$ καθορίζει τον αριθμό των πιο συχνών λέξεων που αφαιρούνται από το λεξιλόγιο. Ο λόγος που αφαιρούμε τις 5000 συχνότερες λέξεις είναι ότι σε ένα σύνολο δεδομένων όπως το IMDB, αυτές οι λέξεις περιλαμβάνουν κυρίως άχρηστα ή πολύ γενικά στοιχεία της γλώσσας (όπως "the", "is", "and", "was"), τα οποία δεν συμβάλλουν σημαντικά στη διάκριση μεταξύ των κλάσεων (θετικές ή αρνητικές κριτικές). Η παράμετρος $k = 1000$ αντιπροσωπεύει τον αριθμό των λιγότερο συχνών λέξεων που αφαιρούνται. Αυτή η επιλογή γίνεται επειδή λέξεις που εμφανίζονται πολύ σπάνια (λιγότερο από λίγες φορές στο dataset) δεν έχουν σημασία και μπορεί να εισάγουν θόρυβο στο μοντέλο. Πολλές από αυτές είναι πιθανότατα τυπογραφικά λάθη, εξειδικευμένες λέξεις ή ονόματα που δεν προσφέρουν πληροφορία που μπορεί να γενικευτεί σε νέο κείμενο. Η παράμετρος $m = 5000$ καθορίζει τον τελικό αριθμό των επιλεγμένων λέξεων με βάση το Information Gain. Ο αριθμός 5000 επιλέγεται γιατί είναι αρκετά μεγάλος ώστε να διατηρεί χρήσιμη πληροφορία για την κατηγοριοποίηση του κειμένου, αλλά όχι τόσο μεγάλος ώστε να δημιουργήσει υπερβολική διάσταση στο μοντέλο. Η επιλογή αυτών των τιμών βασίζεται σε δοκιμές που πραγματοποιήθηκαν στο σύνολο δεδομένων IMDB και στην αξιολόγηση της απόδοσης του μοντέλου. Ανάλογα με τις ανάγκες της εφαρμογής, αυτές οι τιμές μπορούν να προσαρμοστούν.

Ο κώδικας ξεκινά με την εισαγωγή των απαραίτητων βιβλιοθηκών, όπως `torch` για αποθήκευση δεδομένων, `nltk` για επεξεργασία κειμένου, `Counter` για καταμέτρηση συχνοτήτων και `torchtext.datasets.IMDB` για τη φόρτωση του dataset. Επιπλέον, γίνεται λήψη του `punkt tokenizer` από το `nltk`. Το σύνολο δεδομένων IMDB φορτώνεται στη μεταβλητή `train_data`, το οποίο περιέχει κριτικές ταινιών με τις αντίστοιχες ετικέτες `pos` (θετικές) και `neg` (αρνητικές). Τα δεδομένα αποθηκεύονται ως λίστα από πλειάδες (`label`, `text`). Στη συνέχεια διατρέχει το dataset και δημιουργεί έναν `Counter`, ο οποίος αποθηκεύει τη συχνότητα εμφάνισης κάθε λέξης. Για να μην καταμετρώνται οι ίδιες λέξεις πολλές φορές μέσα στο ίδιο κείμενο, χρησιμοποιείται `set(tokens)` ώστε κάθε λέξη να προσμετράται μόνο μία φορά ανά έγγραφο. Το τελικό λεξιλόγιο δημιουργείται αφαιρώντας τις συχνότερες και σπανιότερες λέξεις. Στη συνέχεια, αποθηκεύεται σε αρχείο `vocab.txt`, ενώ δημιουργείται και ένας `word_to_idx dictionary` που χαρτογραφεί κάθε λέξη σε αριθμητικό δείκτη. Αυτό το mapping αποθηκεύεται ως PyTorch tensor (`word_to_idx.pt`), επιτρέποντας τη χρήση του λεξιλογίου σε μοντέλα μηχανικής μάθησης.

process_data.py:

Ο κώδικας προετοιμάζει το σύνολο δεδομένων IMDB για χρήση σε ένα δυαδικό ταξινομητή. Φορτώνει το λεξιλόγιο που έχει δημιουργηθεί από το `build_vocab.py`, μετατρέπει τις κριτικές ταινιών σε διανυσματικές αναπαραστάσεις και αποθηκεύει το επεξεργασμένο dataset σε μορφή PyTorch.

Αρχικά, εισάγονται οι απαραίτητες βιβλιοθήκες. Το `torch` χρησιμοποιείται για την επεξεργασία των δεδομένων και την αποθήκευσή τους σε PyTorch tensors. Η `torchtext.datasets.IMDB` φορτώνει το σύνολο δεδομένων IMDB, ενώ η `torchtext.data.utils.get_tokenizer` χρησιμοποιείται για το tokenization των κειμένων. Οι `Dataset` και `DataLoader` της PyTorch επιτρέπουν τη διαχείριση των δεδομένων, ενώ η βιβλιοθήκη `random` χρησιμοποιείται για την τυχαία ανακατάταξη των δειγμάτων.

Το λεξιλόγιο φορτώνεται από το αρχείο `vocab.txt`, το οποίο περιέχει τις επιλεγμένες λέξεις μετά το φιλτράρισμα που έγινε στο `build_vocab.py`. Κάθε λέξη αποθηκεύεται στη λίστα `vocab`, αφαιρώντας τυχόν κενά. Στη συνέχεια, δημιουργείται ένας αντιστοιχιστής λέξεων σε δείκτες (`word_to_idx`), ο οποίος μετατρέπει κάθε λέξη σε έναν μοναδικό αριθμό, διευκολύνοντας τη μετατροπή των κειμένων σε αριθμητικές αναπαραστάσεις.

Ορίζεται η κλάση `IMDBBinaryDataset`, η οποία κληρονομεί από το `Dataset` της PyTorch. Αυτή η κλάση μετατρέπει τις κριτικές ταινιών του IMDB σε μια δυαδική μορφή που μπορεί να χρησιμοποιηθεί από ένα νευρωνικό δίκτυο. Κατά την αρχικοποίησή της, φορτώνει τον tokenizer `basic_english` και αποθηκεύει τις ετικέτες και τις αναπαραστάσεις των δεδομένων.

Οι ετικέτες IMDB αρχικά αποθηκεύονται ως 1 (αρνητικές κριτικές) και 2 (θετικές κριτικές). Ο κώδικας τις μετατρέπει σε 0 (αρνητικές) και 1 (θετικές), ώστε να χρησιμοποιηθούν σε έναν δυαδικό ταξινομητή. Αν εντοπιστεί μη αναμενόμενη ετικέτα, εμφανίζεται ένα μήνυμα σφάλματος.

Κάθε κείμενο αναλύεται σε λέξεις με τον tokenizer και μετατρέπεται σε ένα δυαδικό διάνυσμα μήκους ίσου με το λεξιλόγιο. Αν μια λέξη υπάρχει στο κείμενο και περιλαμβάνεται στο λεξιλόγιο, η αντίστοιχη θέση στο διάνυσμα παίρνει την τιμή 1, αλλιώς παραμένει 0. Τα δεδομένα αποθηκεύονται σε μορφή PyTorch tensors.

Η κλάση `IMDBBinaryDataset` περιλαμβάνει τις μεθόδους `__len__()` και `__getitem__()`. Η `__len__()` επιστρέφει τον αριθμό των δειγμάτων, ενώ η `__getitem__()` επιτρέπει την πρόσβαση σε μεμονωμένα δείγματα, κάνοντας το dataset συμβατό με το `DataLoader` της PyTorch.

Για να αποφευχθεί η ανισορροπία μεταξύ θετικών και αρνητικών κριτικών, το dataset ανακατεύεται τυχαία πριν περιοριστεί σε 5000 δείγματα. Αυτή η προσέγγιση βοηθά στη βελτίωση της γενίκευσης του μοντέλου, αποτρέποντας φαινόμενα `overfitting`. Αφού δημιουργηθεί το `train_dataset`, αποθηκεύεται ως αρχείο PyTorch (`train_dataset.pt`). Αυτό επιτρέπει τη γρήγορη φόρτωσή του σε επόμενα βήματα εκπαίδευσης του μοντέλου.

train_model.py:

Ο κώδικας `train_model.py` υλοποιεί την εκπαίδευση ενός Stacked BiRNN (Bidirectional Recurrent Neural Network) για την ταξινόμηση κριτικών IMDB σε θετικές και αρνητικές. Περιλαμβάνει τα στάδια προετοιμασίας των δεδομένων, ορισμού του μοντέλου, εκπαίδευσης και αξιολόγησης του ταξινομητή.

Ο κώδικας αρχικά φορτώνει τις απαραίτητες βιβλιοθήκες. Το PyTorch χρησιμοποιείται για την υλοποίηση του νευρωνικού δικτύου, ενώ το TorchText παρέχει το dataset IMDB. Η βιβλιοθήκη Matplotlib επιτρέπει την απεικόνιση των απωλειών εκπαίδευσης και επικύρωσης. Επιπλέον, η Scikit-learn χρησιμοποιείται για τον υπολογισμό των μετρικών Precision, Recall και F1-score.

Η Focal Loss είναι μια παραλλαγή της Binary Cross Entropy Loss, η οποία προσπαθεί να μειώσει την επιρροή εύκολα ταξινομημένων δειγμάτων, δίνοντας μεγαλύτερη έμφαση στα δύσκολα δείγματα.

Στον κώδικα, η Focal Loss ορίζεται με $\alpha=0.25$ και $\gamma=2.0$. Το α είναι ένας συντελεστής που βοηθά στην αντιστάθμιση της ανισορροπίας των κλάσεων, δίνοντας μεγαλύτερη έμφαση στη λιγότερο συχνή κατηγορία. Το γ ελέγχει πόσο έντονα θα μειώνεται η απώλεια για εύκολα δείγματα. Με την τιμή $\gamma=2.0$, η επίδραση των εύκολα ταξινομημένων δειγμάτων μειώνεται σημαντικά, επιτρέποντας στο μοντέλο να συγκεντρωθεί σε πιο δύσκολα δείγματα.

Το σύνολο δεδομένων IMDB φορτώνεται και μετατρέπεται σε δυαδική μορφή, με 0 για αρνητικές κριτικές και 1 για θετικές. Αν δεν υπάρχει αποθηκευμένο σύνολο ελέγχου (test_dataset.pt), το πρόγραμμα το δημιουργεί εκ νέου, αποθηκεύοντας το λεξιλόγιο (word_to_idx.pt) και φορτώνοντας τις δοκιμαστικές κριτικές.

Το σύνολο εκπαίδευσης (train_dataset.pt) διαχωρίζεται σε 80% training και 20% development. Η χρήση αυτής της αναλογίας είναι συνηθισμένη στην εκπαίδευση νευρωνικών δικτύων, καθώς επιτρέπει στο μοντέλο να μάθει από αρκετά δεδομένα, ενώ διαθέτει και ένα σύνολο για έλεγχο της απόδοσης του κατά τη διάρκεια της εκπαίδευσης.

Τα δεδομένα οργανώνονται σε DataLoader, όπου χρησιμοποιείται μέγεθος παρτίδας (batch_size=64). Αυτό σημαίνει ότι το μοντέλο εκπαιδεύεται σε ομάδες των 64 δειγμάτων ανά βήμα, κάτι που επιτρέπει αποδοτική χρήση της μνήμης GPU και ταχύτερη εκπαίδευση.

Το Stacked BiRNN είναι ένα διπλής κατεύθυνσης LSTM (Bidirectional Long Short-Term Memory Network). Αυτό σημαίνει ότι μπορεί να επεξεργάζεται το κείμενο τόσο από την αρχή προς το τέλος, όσο και από το τέλος προς την αρχή, επιτρέποντας στο μοντέλο να μάθει πιο πλούσιες γλωσσικές σχέσεις. Το hidden_dim=256 καθορίζει το μέγεθος του κάθε κρυφού επιπέδου LSTM. Η τιμή 256 είναι συχνά αρκετά μεγάλη ώστε να επιτρέπει την εκμάθηση πολύπλοκων σχέσεων στο κείμενο, χωρίς όμως να κάνει το μοντέλο υπερβολικά μεγάλο και αργό. Ο αριθμός επιπέδων (num_layers=2) σημαίνει ότι το μοντέλο αποτελείται από δύο διαδοχικά επίπεδα LSTM. Η χρήση περισσότερων από ενός επιπέδου βοηθά στην εκμάθηση πιο αφηρημένων γλωσσικών σχέσεων, αλλά αυξάνει το κόστος υπολογισμού. Η τιμή dropout=0.3 εφαρμόζει ένα ποσοστό 30% dropout στις συνδέσεις μεταξύ των επιπέδων. Το Dropout χρησιμοποιείται για την αποφυγή υπερεκπαίδευσης (overfitting), αποτρέποντας το μοντέλο από το να απομνημονεύει τα δεδομένα εκπαίδευσης. Το τελικό επίπεδο του μοντέλου είναι μια γραμμική συνάρτηση ενεργοποίησης (self.fc), η οποία προβλέπει την πιθανότητα μιας κριτικής να είναι θετική.

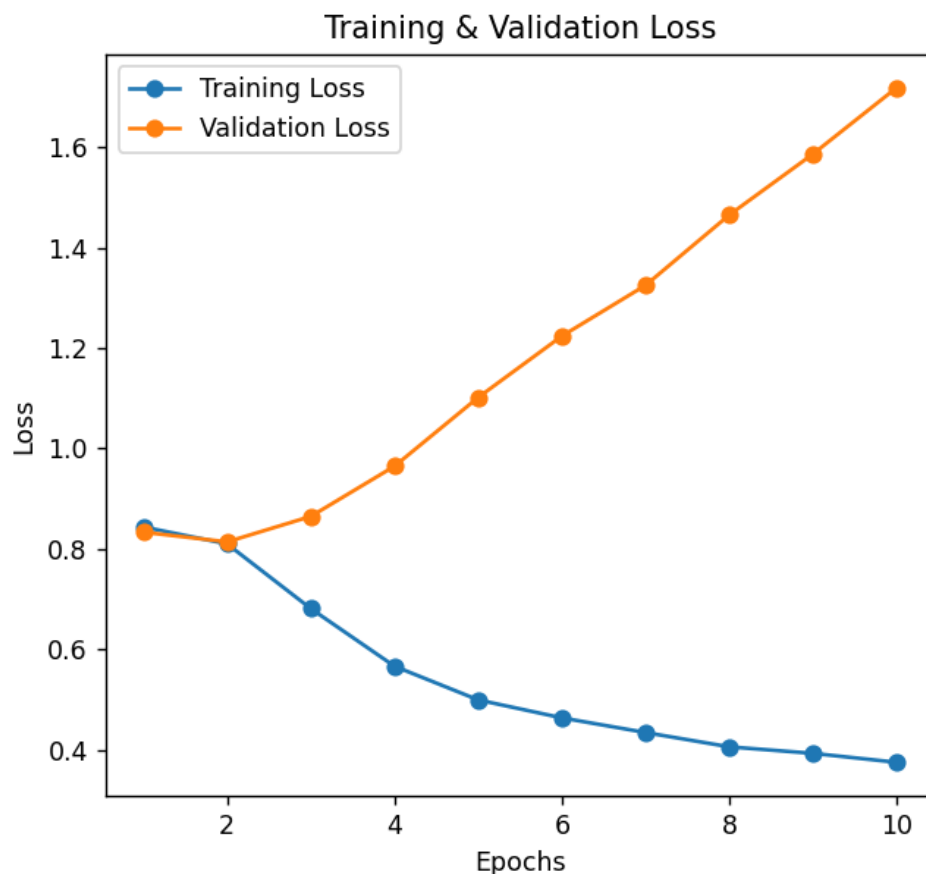
Για τη συνάρτηση απωλειών, χρησιμοποιείται Binary Cross Entropy Loss με pos_weight=1.3. Ο συντελεστής pos_weight=1.3 δίνει μεγαλύτερη έμφαση στις θετικές κριτικές, σε περίπτωση που υπάρχει ανισορροπία μεταξύ των δύο κλάσεων.

Ο Adam Optimizer χρησιμοποιείται για την εκπαίδευση του μοντέλου, με $lr=0.0005$ και $weight_decay=1e-5$. Το learning rate (lr) καθορίζει πόσο γρήγορα προσαρμόζονται τα βάρη του μοντέλου κατά τη διάρκεια της εκπαίδευσης. Η τιμή 0.0005 είναι αρκετά μικρή ώστε να εξασφαλίζει σταθερή εκπαίδευση, αλλά και αρκετά μεγάλη ώστε να επιτρέπει ταχύτερη σύγκλιση. Το $weight_decay$ ($1e-5$) χρησιμοποιείται για να αποτρέψει το overfitting, περιορίζοντας την προσαρμογή των βαρών σε μεγάλες τιμές.

Το μοντέλο εκπαιδεύεται για $EPOCHS=10$ εποχές. Η επιλογή 10 εποχών βασίζεται σε δοκιμές, όπου διαπιστώθηκε ότι το μοντέλο συγκλίνει επαρκώς σε αυτό το διάστημα. Σε μεγαλύτερο αριθμό εποχών, το μοντέλο ενδέχεται να αρχίσει να προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης. Κατά την εκπαίδευση, η απώλεια υπολογίζεται για κάθε παρτίδα (batch), και στη συνέχεια ενημερώνονται τα βάρη του μοντέλου με backpropagation. Μετά από κάθε εποχή, το μοντέλο αξιολογείται στο σύνολο επικύρωσης (val_dataset), καταγράφοντας την απώλεια επικύρωσης.

Το μοντέλο αξιολογείται στο σύνολο ελέγχου (test_dataset) και καταγράφονται οι προβλέψεις (y_pred) και οι πραγματικές τιμές (y_true). Για την αποτίμηση της απόδοσης, χρησιμοποιείται το Classification Report της Scikit-learn, το οποίο παρέχει μετρικές όπως Precision, Recall και F1-score. Η επιλογή ορίου 0.5 ($outputs.squeeze() > 0.5$) για την ταξινόμηση βασίζεται σε δοκιμές, όπου διαπιστώθηκε ότι αυτή η τιμή προσφέρει καλύτερη ισορροπία μεταξύ False Positives και False Negatives.

Τέλος, δημιουργείται ένα διάγραμμα που δείχνει την εξέλιξη της απώλειας εκπαίδευσης και επικύρωσης στις 10 εποχές. Αυτό επιτρέπει την αξιολόγηση της προόδου του μοντέλου και την ανίχνευση πιθανών προβλημάτων όπως overfitting.



Το γράφημα δείχνει ότι το μοντέλο μαθαίνει με επιτυχία από τα δεδομένα εκπαίδευσης, καθώς η απώλεια εκπαίδευσης μειώνεται σταθερά σε κάθε εποχή. Αυτό σημαίνει ότι το μοντέλο βελτιώνει συνεχώς την ικανότητά του να αναγνωρίζει μοτίβα στα δεδομένα και να προσαρμόζει τα βάρη του με ακρίβεια. Επιπλέον, το γεγονός ότι η απώλεια επικύρωσης ξεκινά σε χαμηλό επίπεδο και παραμένει σταθερή για τις πρώτες εποχές δείχνει ότι το μοντέλο είναι ικανό να γενικεύει καλά στα αρχικά στάδια της εκπαίδευσης.

Το μοντέλο καταφέρνει να αξιοποιήσει τις πληροφορίες από τα δεδομένα εκπαίδευσης και εμφανίζει μια συνεπή βελτίωση στην ικανότητά του να κάνει προβλέψεις. Η χρήση διπλής κατεύθυνσης LSTM (BiLSTM) φαίνεται να βοηθά σημαντικά στην κατανόηση των σχέσεων μεταξύ των λέξεων στα κείμενα, προσφέροντας μια πλούσια αναπαράσταση των δεδομένων. Το learning rate των 0.0005 επιτρέπει σταδιακές και σταθερές προσαρμογές στα βάρη, χωρίς απότομες αλλαγές, διατηρώντας την εκπαίδευση αποτελεσματική. Επίσης, η χρήση του Adam optimizer επιτρέπει στο μοντέλο να συγκλίνει πιο γρήγορα και αποτελεσματικά.

Γενικά, η εκπαίδευση του μοντέλου δείχνει ιδιαίτερα θετικά αποτελέσματα, καθώς μαθαίνει σταδιακά και βελτιώνει την ικανότητά του να αναλύει και να ταξινομεί κείμενα. Το γεγονός ότι η απώλεια εκπαίδευσης μειώνεται τόσο ομαλά είναι ένδειξη ενός καλά ρυθμισμένου συστήματος μάθησης, όπου το μοντέλο βελτιώνεται με κάθε εποχή.

Classification Report:			
	precision	recall	f1-score
Negative (0)	0.554755	0.70880	0.622388
Positive (1)	0.596855	0.43112	0.500627
macro avg	0.575805	0.56996	0.561507
weighted avg	0.575805	0.56996	0.561507

Το classification report δείχνει ότι το μοντέλο έχει μια αρκετά ισορροπημένη απόδοση μεταξύ των δύο κλάσεων, με την ακρίβεια (precision) και την ανάκληση (recall) να είναι συγκρίσιμες τόσο για τις θετικές όσο και για τις αρνητικές κριτικές. Συγκεκριμένα, η precision για τις αρνητικές κριτικές (55.4%) είναι ελαφρώς χαμηλότερη από αυτή των θετικών (59.6%), γεγονός που δείχνει ότι το μοντέλο μπορεί να αναγνωρίσει με αρκετή ακρίβεια ποια κριτική είναι πραγματικά θετική ή αρνητική. Από την άλλη, η ανάκληση για τις αρνητικές κριτικές (70.8%) είναι υψηλότερη σε σχέση με τις θετικές (43.1%), κάτι που σημαίνει ότι το μοντέλο καταφέρνει να εντοπίσει την πλειονότητα των αρνητικών κριτικών, αλλά μπορεί να χάνει μερικές θετικές. Το macro avg και weighted avg F1-score (~56.1%) δείχνουν ότι το μοντέλο έχει μια συνεπή συνολική απόδοση, προσφέροντας μια καλή ισορροπία μεταξύ precision και recall. Αυτό δείχνει ότι το μοντέλο μπορεί να κάνει σταθερές προβλέψεις ανεξαρτήτως της κατανομής των δεδομένων. Παρόλο που υπάρχει περιθώριο βελτίωσης, τα αποτελέσματα δείχνουν ότι το μοντέλο είναι ήδη λειτουργικό και μπορεί να βελτιωθεί περαιτέρω με fine-tuning και τεχνικές regularization για ακόμα μεγαλύτερη ακρίβεια.

Σύγκριση Α και Γ :

Ο Naïve Bayes από το Α' μέρος παρουσιάζει σαφώς καλύτερες επιδόσεις σε σχέση με το Stacked BiRNN του Γ' μέρους, καθώς επιτυγχάνει Test Macro F1-score 80.3%, το οποίο είναι σημαντικά υψηλότερο από το ~56% του BiRNN. Η Precision για τις θετικές κριτικές (85.17%) είναι επίσης ανώτερη από την 59.6% του BiRNN, δείχνοντας ότι ο Naïve Bayes αναγνωρίζει σωστά περισσότερες θετικές κριτικές χωρίς να κάνει πολλά λάθη. Επιπλέον, η Recall για τις θετικές κριτικές (73.60%) είναι υψηλότερη από την 43.1% του BiRNN, κάτι που σημαίνει ότι το μοντέλο του Α' μέρους καταφέρνει να ανιχνεύσει πολύ περισσότερα θετικά δείγματα. Το F1-score για την Κλάση 1 (78.96%) είναι επίσης βελτιωμένο σε σχέση με το BiRNN, επιτυγχάνοντας καλύτερη ισορροπία μεταξύ precision και recall. Συνολικά, ο Naïve Bayes φαίνεται να λειτουργεί εξαιρετικά, παρά την απλότητά του, ενώ το BiRNN παρουσιάζει χαμηλότερες επιδόσεις, πιθανότατα λόγω υπερπροσαρμογής ή ανεπαρκούς ρύθμισης των υπερπαραμέτρων του.

Ο Random Forest από το Α' μέρος υπερέχει σε σχέση με το Stacked BiRNN του Γ' μέρους, προσφέροντας καλύτερη ισορροπία μεταξύ precision και recall. Συγκεκριμένα, το Test Macro F1-score του Random Forest (67.9%) είναι σημαντικά υψηλότερο από το 56% του BiRNN, υποδηλώνοντας ότι το μοντέλο μπορεί να κατηγοριοποιεί καλύτερα και τις δύο κλάσεις χωρίς να χάνει πολλά θετικά ή αρνητικά δείγματα. Η Precision για τις θετικές κριτικές (64.49%) είναι καλύτερη από την 59.6% του BiRNN, δείχνοντας ότι το Random Forest κάνει λιγότερα λάθη όταν χαρακτηρίζει μια κριτική ως θετική. Παράλληλα, η Recall για τις θετικές κριτικές (76.89%) είναι ανώτερη από το 43.1% του BiRNN, κάτι που σημαίνει ότι το Random Forest αναγνωρίζει περισσότερες θετικές κριτικές, σε αντίθεση με το BiRNN, το οποίο χάνει πολλά θετικά δείγματα. Αυτό πιθανώς οφείλεται στο γεγονός ότι το Random Forest είναι λιγότερο επιρρεπές σε overfitting σε σχέση με το BiRNN, το οποίο απαιτεί περισσότερα δεδομένα για να γενικεύσει σωστά. Συνολικά, το Random Forest είναι ανώτερο από το BiRNN, αφού καταφέρνει να ισορροπήσει καλύτερα ακρίβεια και ανάκληση, προσφέροντας πιο αξιόπιστες προβλέψεις χωρίς να θυσιάζει κάποια από τις δύο μετρικές.

Ο AdaBoost από το Α' μέρος υπερέχει του Stacked BiRNN από το Γ' μέρος σε πολλές μετρικές, αλλά το BiRNN είναι καλύτερο στο recall και F1-score για τις αρνητικές κριτικές. Συγκεκριμένα, το AdaBoost έχει υψηλότερη Precision για τις αρνητικές κριτικές (72.86%) έναντι 55.47% του BiRNN, δείχνοντας ότι κάνει λιγότερα λάθη όταν ταξινομεί μια κριτική ως αρνητική. Ωστόσο, το Recall για τις αρνητικές κριτικές (54.18%) είναι χαμηλότερο από το 70.88% του BiRNN, κάτι που σημαίνει ότι το BiRNN καταφέρνει να εντοπίσει περισσότερες αρνητικές κριτικές, έστω και με χαμηλότερη ακρίβεια. Αντίστοιχα, το F1-score για τις αρνητικές κριτικές (62.15%) στο AdaBoost είναι χαμηλότερο από το 62.24% του BiRNN, κάτι που δείχνει ότι το BiRNN έχει ελαφρώς καλύτερη ισορροπία precision-recall στις αρνητικές κριτικές. Για τις θετικές κριτικές, το AdaBoost έχει Precision 61.83% έναντι 59.69% του BiRNN, ενώ το Recall (77.68%) είναι πολύ υψηλότερο από το 43.11% του BiRNN, αποδεικνύοντας ότι το AdaBoost αναγνωρίζει πολύ περισσότερες θετικές κριτικές. Το F1-score για τις θετικές κριτικές (68.86%) είναι επίσης ανώτερο από το 50.06% του BiRNN, επιβεβαιώνοντας ότι το AdaBoost διατηρεί καλύτερη ισορροπία μεταξύ precision και recall στις θετικές κριτικές. Συνολικά, το AdaBoost υπερέχει στη συνολική ισορροπία precision και recall, με υψηλότερο macro precision (67.35%) και macro F1-score (65.50%) έναντι του BiRNN. Ωστόσο, το BiRNN έχει πλεονέκτημα στην ανάκληση και το F1-score των αρνητικών κριτικών, γεγονός που σημαίνει ότι καταφέρνει να εντοπίσει περισσότερες αρνητικές κριτικές, έστω και με χαμηλότερη ακρίβεια.

Συγκριση Β και Γ

Το Scikit-learn Naïve Bayes υπερτερεί του BiRNN σε όλες τις μετρικές, τόσο για τις αρνητικές όσο και για τις θετικές κριτικές. Συγκεκριμένα, η precision για την Κλάση 0 (71.95%) είναι υψηλότερη από την 57.6% του BiRNN, ενώ και η recall για την Κλάση 0 (87.14%) ξεπερνά κατά πολύ το 47.19% του BiRNN, δείχνοντας ότι το Naïve Bayes ανιχνεύει περισσότερες αρνητικές κριτικές χωρίς να κάνει τόσα λάθη. Αντίστοιχα, για τις θετικές κριτικές, η precision του Naïve Bayes (83.7%) είναι πολύ ανώτερη από την 55.3% του BiRNN, ενώ και η recall για την Κλάση 1 (66.04%) είναι υψηλότερη από το 43.1% του BiRNN, αποδεικνύοντας ότι το Naïve Bayes εντοπίζει περισσότερες θετικές κριτικές. Το συνολικό Macro F1-score του Naïve Bayes (76.32%) είναι πολύ ανώτερο από το 55.87% του BiRNN, επιβεβαιώνοντας ότι προσφέρει μια πολύ πιο αξιόπιστη ισορροπία μεταξύ precision και recall, τόσο για τις θετικές όσο και για τις αρνητικές κριτικές. Συνολικά, το Scikit-learn Naïve Bayes αποδίδει καλύτερα από το BiRNN σε όλες τις κατηγορίες, προσφέροντας υψηλότερη ακρίβεια, recall και συνολική ποιότητα ταξινόμησης.

Ο Scikit-learn Random Forest υπερέχει του Stacked BiRNN σε όλες σχεδόν τις μετρικές. Η Precision για τις αρνητικές κριτικές (80.81%) είναι σημαντικά υψηλότερη από την 55.47% του BiRNN, ενώ το Recall (65.27%) είναι χαμηλότερο από το 70.88% του BiRNN, δείχνοντας ότι το BiRNN ανιχνεύει περισσότερες αρνητικές κριτικές αλλά με χαμηλότερη ακρίβεια. Στις θετικές κριτικές, το Random Forest έχει Precision 70.87% έναντι 59.69% του BiRNN, ενώ η Recall (84.50%) είναι πολύ ανώτερη από το 43.11% του BiRNN, αποδεικνύοντας ότι το Random Forest εντοπίζει περισσότερες θετικές κριτικές με μεγαλύτερη ακρίβεια. Το Macro F1-score (74.65%) είναι πολύ υψηλότερο από το 56.15% του BiRNN, επιβεβαιώνοντας την ανώτερη συνολική απόδοση του Random Forest, το οποίο είναι σαφώς πιο αξιόπιστος ταξινομητής.

Ο Sklearn AdaBoost από το Β' μέρος υπερέχει του Stacked BiRNN από το Γ' μέρος σε όλες σχεδόν τις μετρικές, εκτός από το recall για τις αρνητικές κριτικές, όπου το BiRNN έχει καλύτερη απόδοση. Συγκεκριμένα, η Precision για τις αρνητικές κριτικές στο AdaBoost (81.87%) είναι πολύ ανώτερη από την 55.47% του BiRNN, αποδεικνύοντας ότι το AdaBoost κάνει λιγότερα λάθη όταν ταξινομεί μια κριτική ως αρνητική. Ωστόσο, το recall για τις αρνητικές κριτικές (60.87%) στο AdaBoost είναι χαμηλότερο από το 70.88% του BiRNN, κάτι που σημαίνει ότι το BiRNN αναγνωρίζει περισσότερες αρνητικές κριτικές, αν και με χαμηλότερη ακρίβεια.

Για τις θετικές κριτικές, το AdaBoost υπερέχει ξεκάθαρα, με Precision 69.47% έναντι 59.69% του BiRNN και Recall 87.29% έναντι 43.11%, δείχνοντας ότι αναγνωρίζει πολύ περισσότερες θετικές κριτικές με μεγαλύτερη ακρίβεια. Το F1-score για τις θετικές κριτικές (77.37%) είναι επίσης ανώτερο από το 50.06% του BiRNN, επιβεβαιώνοντας ότι το AdaBoost διατηρεί καλύτερη ισορροπία μεταξύ precision και recall.

Συνολικά, το Macro F1-score του AdaBoost (73.59%) είναι σημαντικά υψηλότερο από το 56.15% του BiRNN, δείχνοντας την ανώτερη συνολική απόδοσή του. Παρόλο που το BiRNN έχει καλύτερο recall στις αρνητικές κριτικές, το AdaBoost υπερέχει σε όλες τις υπόλοιπες μετρικές, προσφέροντας καλύτερη ισορροπία και μεγαλύτερη ακρίβεια στη συνολική ταξινόμηση.

