Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης & Υπολογιστων

LLMs for Biased Tabular Datasets

A Diploma Thesis

by

Ioannis Rekkas

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2024

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής & Υπολογιστών
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

LLMs for Biased Tabular Datasets

A Diploma Thesis

by

Ioannis Rekkas

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 16η Ιουλίου, 2024.

.....	.....	.....
Γεώργιος Στάμου	Αθανάσιος Βουλόδημος	Παρασκευή Τζούβελη
Καθηγητής Ε.Μ.Π.	Επ. Καθηγητής Ε.Μ.Π.	Ε.ΔΙ.Π. Ε.Μ.Π.

Αθήνα, Ιούλιος 2024

.....

Ιωαννης Ρεκκας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Τα δεδομένα σε μορφή πίνακα συνήθως αντιμετωπίζονται στην μηχανική μάθηση με τη χρήση μοντέλων δέντρων ή δασών αποφάσεων. Όμως είναι γνωστό ότι τα μοντέλα αυτά είναι αρκετά επιρρεπή στην εκμάθηση τυχών μεροληψιών που μπορεί να περιέχουν τα εκάστοτε σύνολα δεδομένων. Πρόσφατα η άνοδος των μεγάλων γλωσσικών μοντέλων έχει αναδείξει τρόπους χρήσης τους και για δεδομένα μορφής πίνακα μέσω σειριοποίησης των δειγμάτων τους σε κείμενο. Έτσι σε αυτήν την εργασία ερευνάται η ικανότητα των μεγάλων γλωσσικών μοντέλων να ξεπερνούν μεροληψίες σε σύνολα δεδομένων μορφής πίνακα χρησιμοποιώντας πρότερη γνώση που απέκτησαν κατά το στάδιο εκπαίδευσής τους καθώς και την κατανόηση της σημασιολογίας των τιμών των κατηγορηματικών χαρακτηριστικών των δεδομένων. Τα αποτελέσματα υποδεικνύουν ότι τα μεγάλα γλωσσικά μοντέλα τα πηγαίνουν το ίδιο καλά ή καλύτερα από τις μεθόδους που αποτελούν την τελευταία λέξη της τεχνολογίας για δεδομένα σε μορφή πίνακα υπό το καθεστώς μεροληψίας.

Λέξεις-κλειδιά — Δεδομένα σε Μορφή Πίνακα, Μεροληψία, Μεγάλα Γλωσσικά Μοντέλα, Δέντρα Αποφάσεων, Σειριοποίηση, Πρότερη Γνώση, Σημασιολογία Κατηγωρηματικών Χαρακτηριστικών

Abstract

Tabular Data are usually predicted by decision tree or decision forest models in machine learning. However it is known that such models are quite susceptible to learning data bias potentially present in datasets. Recently, the rise of Large Language Models has lead to discovery of ways of utilizing LLMs for tabular data predictions by serializing data samples to text. Thus in this thesis the capability of Large Language Models to overcome tabular data bias via utilization of prior knowledge gathered during training and via understanding of categorical feature semantics is examined. Results indicate that LLMs can do as well as or better than state-of-the-art methods for tabular data under bias conditions.

Keywords — Tabular Data, Bias, Large Language Models, Decision Trees, Serialization, Prior Knowledge, Categorical Feature Semantics

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή κύριο Στάμου για την στήριξη του στην ακαδιμαϊκή μου πορεία καθώς και τον καθηγητή κύριο Βουλόδημο για την ευκαιρία που μου έδωσαν να εργαστώ στο εργαστήριο τους για την διπλωματική μου εργσία. Επιπλέον, από το εργαστήριο θέλω να ευχαριστήσω τους υποψήφιους διδάκτορες Γιώργο Φιλανδριανό και Μαρία Λυμπεραίου για την συνεργασία και καθοδήγηση που μου παρείχαν για αυτήν την εργασία. Επίσης από το εργαστήριο, ευχαριστώ την κυρία Παρασκευή Τζούβελη για την τεχνική υποστήριξη που μου παρείχε για την πρόσβαση χρήσημων υπολογιστών πόρων χωρίς τους οποίους η εργασία αυτή θα ήταν αδύνατη.

Γιάννος Ρέκκας,

Ιούλιος 2024

Contents

List of Figures

List of Tables

Chapter 1

Εκτεταμένη Περίληψη στα Ελληνικά

Contents

1.1 Εισαγωγή

1.1.1 Μεγάλα Γλωσσικά Μοντέλα

Ενάμιση χρόνο μετά την δημοσίευση του ChatGPT έχει υπάρξει μεγάλο ενδιαφέρον από όλους τους τομείς της κοινωνίας για την περιοχή της τεχνητής νοημοσύνης που έχει ονομαστεί επεξεργασία φυσικής γλώσσας. Το κύριο εργαλείο αυτής πλέον, επηρεασμένο από τις προσταγές της βαθιάς μάθησης οι οποίες επιτάσσουν την χρήση όλο και μεγαλύτερων μοντέλων για την αύξηση των επιδόσεων, είναι τα μεγάλα γλωσσικά μοντέλα. Πιο συγκεκριμένα, εκπληκτικά καλές επιδώσεις παρατηρούνται από τα μεγάλα γλωσσικά μοντέλα που υλοποιούνται με την δομή του λεγόμενου «μετασχηματιστή».

Σε ένα αφαιρετικό επίπεδο περιγραφής, μετασχηματιστής είναι μία συγκεκριμένη δομή της περιοχής της βαθιάς μάθησης, εκπαιδεύσιμη, που δέχεται ως είσοδο μία αλληλουχία λεκτικών μονάδων (οι οποίες συνήθως είναι είτε λέξεις είτε συνθετικά λέξεων) και μετά από κάποιες αλγεβρικές πράξεις την μετασχηματίζει σε μία άλλη. Βασικό συστατικό του είναι ο μηχανισμός προσοχής που μαθαίνει να τονίζει τις σημασιολογικά σημαντικές λεκτικές μονάδες μίας πρότασης ώστε να χρησιμοποιηθούν για την παραγωγή της ακολουθίας εξόδου. Αρχικά είχε χρησιμοποιηθεί για μετάφραση [20], αλλά σύντομα αποδείχθηκε πολύ χρήσιμος για μοντελοποίηση γλώσσας, δηλαδή πρόβλεψη της επόμενης λεκτικής μονάδας σε μια αλληλουχία, που είναι και ο τρόπος λειτουργίας των μεγάλων γλωσσικών μοντέλων.

Το ChatGPT χρησιμοποιεί τον μετασχηματιστή GPT (Generative Pretraining Transformer) το όνομά του οποίου προέρχεται από τα αρχικά της έκφρασης «Γεννητικός Προ-εκπαιδευμένος Μετασχηματιστής» στα αγγλικά. Γεννητικός (κάποιες φορές αναφέρεται στα ελληνικά και ως παραγωγικός) δηλώνει ότι ως μοντέλο δεν κατηγοριοποιεί απλά τα δεδομένα εισόδου αλλά τα χρησιμοποιεί για να παράξει (γεννήσει) άλλα δεδομένα. Προ εκπαιδευμένος, ότι έχει εκπαιδευτεί σε προηγούμενο χρόνο, κάτι σύνηθες για μεγάλα μοντέλα όπου τα κόστη εκπαίδευσης είναι ακριβά. Οι πιο καινούριες εκδοχές του όπως ο GPT-4 βαθμολογούνται ανάμεσα στους κορυφαίους ανθρώπους για διάφορες γραπτές εξετάσεις [17].

Πέραν του GPT υπάρχουν και άλλοι μετασχηματιστές που χρησιμοποιούνται για εφαρμογές ψηφιακού βοηθού (Bing Copilot, Gemini, Claude) ενώ σε αντίθεση με τους προηγούμενους υπάρχουν και αρκετοί που παρέχονται και ξεχωριστά για την ανάπτυξη εφαρμογών οποιουδήποτε τύπου (Llama, Vicuna, Mistral, T0). Όλα τα προαναφερθέντα μοντέλα έχουν υψηλές απαιτήσεις σε μνήμη γραφικών που φέρνει το κόστος αγοράς υπολογιστών ικανών για την εκτέλεσή τους σε μερικές χιλιάδες ευρώ. Επιπλέον, μερικές φορές οι υπαρκτές κάρτες γραφικών δεν παρέχουν την απαιτούμενη μνήμη και αναγκαστικά πρέπει να χρησημοποιηθούν συστοιχίες υπολογιστών όπως κέντρα δεδομένων και υπερυπολογιστές, γεγονός που αυξάνει δραματικά την πολυπλοκότητα (και τα κόστη) της διαδικασίας.

Μια σημαντική ιδιότητα που κατέχουν τα μεγάλα γλωσσικά μοντέλα πέραν της αντίληψης φυσικής γλώσσας είναι η αποκαλούμενη «πρότερη γνώση». Μετά την εκπαίδευσή τους μπορούν να απαντήσουν ορθά σε πολλές ερωτήσεις γνώσεων χωρίς να χρειαστεί να ψάξουν κάπου. Αυτό όμως δεν συμβαίνει πάντα και όταν κάνουν λάθος συνήθως η απάντησή τους, αν διαβαστεί σε φυσική γλώσσα, δεν δείχει ίχνη αμφιβολίας, ενώ όταν διορθωθούν από κάποιον τότε αλλάζουν γνώμη. Για αυτόν τον λόγο όταν συμβαίνει αυτό λέγεται ότι «το γλωσσικό μοντέλο έχει παραισθήσεις».

1.1.2 Δεδομένα σε μορφή Πίνακα

Σε πολλές εφαρμογές τεχνητής νοημοσύνης παρέχεται ένα σύνολο δεδομένων το οποίο μπορεί να εκφραστεί με την δομή ενός πίνακα και ζητείται να γίνει κάποια σχετική πρόβλεψη. Οι στήλες του πίνακα αντιστοιχούν σε μετρήσεις ή προτάσεις για τα στοιχεία του συνόλου δεδομένων και λέγονται χαρακτηριστικά, ενώ οι σειρές του πίνακα αντιστοιχούν σε κάθε στοιχείο. Οι μετρήσεις συνήθως ονομάζονται αριθμητικά δεδομένα ενώ οι προτάσεις κατηγορικά. Για παράδειγμα παρουσιάζεται ένα υποσύνολο με τρία στοιχεία από κάποιο υποθετικό σύνολο δεδομένων στον πίνακα 1.1.

Τα αριθμητικά δεδομένα εκ φύσεως έχουν μία δομή η οποία περιέχει κάποια σημασιολογικά χαρακτηριστικά που γίνονται εύκολα κατανοητά από μία μηχανή. Ως πραγματικοί αριθμοί έχουν διάταξη και επιπλέον μπορούν να συνδυαστούν με πράξεις όπως πρόσθεση, πολλαπλασιασμό αλλά και πιο σύνθετες πράξεις ώστε να εκφράσουν συνδυασμένες τιμές τους ή και λογικές συνθήκες αυτών.

Αυτό δεν ισχύει όμως για τα κατηγορικά δεδομένα τα οποία δεν έχουν κατ'ανάγκην κάποια αλγεβρική δομή.

Ονοματεπώνυμο	Κατοικία	Ύψος (μ.)	Χρώμα Ματιών
Λούσι Μακλέιν	Λος Άντζελες	1,65	καφέ
Κούπερ Χάουαρντ	Λος Άντζελες	1,78	καφέ
Ρόμπερτ Έντουιν Χάους	Λας Βέγκας	1,92	καφέ

Table 1.1: Παράδειγμα Δεδωμένων σε Πίνακα

Έτσι στις μέχρι πρόσφατα υπάρχουσες μεθόδους για την μοντελοποίηση δεδομένων πίνακα, οι αλγόριθμοι δεν είχαν την ικανότητα να λάβουν υπόψη κάποια πιθανή σημασιολογία των κατηγορικών δεδομένων. Η μόνη σημασιολογία που χρησημοποιούσαν για αυτά στις προβλέψεις τους ήταν αν διαφέρουν ή είναι ακριβώς ίδια.

1.1.3 Χρήση Μεγάλων Γλωσσικών Μοντέλων για Προβλέψεις σε δεδομένα Πίνακα

Με την άνοδο των μεγάλων γλωσσικών μοντέλων, η αξιοποίηση της σημασιολογίας και των κατηγορικών δεδομένων γίνεται πλέον εφικτή. Εισάγοντας μεγάλα γλωσσικά μοντέλα σε ένα σύστημα τεχνητής νοημοσύνης καθίσταται πλέον δυνατή η εκμάθησή της και συνεπώς η χρήση της για τις προβλέψεις. Επιπλέον, η επίδοση ενός τέτοιου συστήματος ενισχύεται και από την πρώτερη γνώση του μοντέλου που μπορεί να του δώσει έτοιμες απαντήσεις και για το τι σημαίνει κάποιος συνδυασμός των κατηγορικών δεδομένων.

Αυτό ερευνήθηκε στην περίπτωση των δεδομένα πίνακα σε μια σειρά από πρόσφατες δημοσιεύσεις. Σε αυτές τυπικά κάθε γραμμή του πίνακα σειριοποιούνταν σε κάποιου είδους πρόταση και δινόταν στο γλωσσικό μοντέλο από το οποίο στη συνέχια ζητούνταν να παραχθεί κάποια πρόβλεψη. Οι ερευνητές παρατήρησαν υπό κάποιες συνθήκες επίδοση καλύτερη σε σχέση με τις εώς τότε καλύτερες μεθόδους τεχνητής νοημοσύνης για αυτό το πρόβλημα, τα δένδρα αποφάσεων [14].

1.1.4 Ανθεκτικότητα σε Σύνολα δεδομένων που έχουν Μεροληψία

Ένα από τα γνωστά προβλήμματα των μεθόδων δενδρών και δασών αποφάσεων στην τεχνητή νοημοσύνη είναι ότι είναι αρκετά επιρρεπή στο να μαθαίνουν μεροληψίες του συνόλου δεδομένων εκπαίδευσης όταν αυτό είναι μη αντιπροσωπευτικό. Αυτό έχει ως αποτέλεσμα μετά την εκπαίδευσή τους να έχουν χαμηλή επίδοση σε πραγματικά δεδομένα και να κάνουν μη ακριβείς προβλέψεις ή ακόμη και παραπλανητικές [8, 11, 6, 7, 10]. Παρά αυτήν την αδυναμία τους είχαν παραμείνει ως μία από τις καλύτερες σε επίδοση μεθόδους για τα δεδομένα πίνακα μέχρι τις προηγούμενες δημοσιεύσεις.

Η εξής ιδέα γεννάται από τα προηγούμενα: Αφού τα γλωσσικά μοντέλα έχουν την ικανότητα να προσφέρουν κατανόηση της σημασιολογίας κατηγορικών δεδομένων καθώς και σχετικές πρότερες γνώσεις, θα μπορούσε ένα σύστημα τεχνητής νοημοσύνης να τα αξιοποιήσει ώστε να μπορέσει να ανακάψει στην περίπτωση που το σύνολο δεδομένων στο οποίο εκπαιδεύεται είναι μη αντιπροσωπευτικό, δηλαδή περιέχει αυτό που στα αγγλικά ονομάζεται *bias* (μεροληψία);

Αυτό είναι το ερώτημα που επιχειρείται να απαντηθεί σε αυτήν την διπλωματική εργασία. Χρησημοποιούνται μεγάλα γλωσσικά μοντέλα σε σύνολα δεδομένων μορφής πίνακα στα οποία έχουν πρόστεθεί διάφορα είδη μεροληψίας και συγκρίνονται με άλλα μοντέλα σχεδιασμένα για δεδομένα πίνακα.

1.2 Θεωρητικό Μέρος

1.2.1 Δημιουργία Μεροληψίας σε Σύνολα δεδομένων

Για την προσθήκη μεροληψίας στα διαθέσιμα σύνολα δεδομένων χρησιμοποιήθηκαν πέντε μέθοδοι.

- **Μεροληψία ετικέτας (label bias)**. Σε αυτό το είδος μεροληψίας όλα τα δείγματα με μία συγκεκριμένη τιμή ετικέτας αφαιρούνται από το σύνολο δεδομένων.
- **Μεροληψία χαρακτηριστικού (feature bias)**. Σε αυτό το είδος μεροληψίας όλα τα δείγματα με μία συγκεκριμένη τιμή χαρακτηριστικού αφαιρούνται από το σύνολο δεδομένων.

- **Μεροληψία συνδυασμού (combined bias)**. Σε αυτό το είδος μεροληψίας όλα τα δείγματα που εμφανίζουν έναν συγκεκριμένο συνδυασμό χαρακτηριστικού - ετικέτας αφαιρούνται από το σύνολο δεδομένων.
- **Μεροληψία ζεύγους χαρακτηριστικών (fpair bias)**. Σε αυτό το είδος μεροληψίας όλα τα δείγματα που εμφανίζουν ένα συγκεκριμένο ζεύγους τιμών σε δύο συγκεκριμένα διαφορετικά χαρακτηριστικά αφαιρούνται από το σύνολο δεδομένων.
- **Διπλή μεροληψία χαρακτηριστικού (double feature bias)**. Αυτό το είδος μεροληψίας είναι ισοδύναμο με την διπλή εφαρμογή της μεθόδου για προσθήκη μεροληψίας χαρακτηριστικού.

Από αυτές τις μεροληψίες πιο ρεαλιστικές θεωρούνται η μεροληψία συνδυασμού και η μεροληψία ζεύγους χαρακτηριστικών καθώς πιστεύεται ότι συνήθως στα σύνολα δεδομένων όταν λείπουν στοιχεία, αυτά λείπουν κατά ομάδες (π.χ. ζευγών χαρακτηριστικών), κάποιες φορές ίσως και λόγω κάποιου σφάλματος στην μέθοδο ή τον μηχανισμό συλλογής δεδομένων. Οι υπόλοιπες μεροληψίες παραμένουν όμως χρήσιμες για την κατανόηση του πώς ένα μοντέλο συμπεριφέρεται υπό διάφορες συνθήκες μεροληψίας.

1.2.2 Μηχανισμός Προβλέψεων με Χρήση Μεγάλου Γλωσσικού Μοντέλου

Ο μηχανισμός μεγάλου γλωσσικού μοντέλου που χρησιμοποιήθηκε παρουσιάζει ομοιότητες με τον μηχανισμό στο [14]. Τα δεδομένα πίνακα σειριοποιούνται σε κείμενο και εισάγονται στο μεγάλο γλωσσικό μοντέλο το οποίο παράγει την πρόβλεψη. Επιπλέον μπορεί να χρησιμοποιηθεί υποβοήθηση ολίγων παραδειγμάτων (few-shot prompting) ή να γίνει εκπαίδευση με λίγα δείγματα (few-shot training) ή και τα δύο μαζί. Η εκπαίδευση γίνεται με μέθοδο fine-tuning για μικρό αριθμό εποχών. Το μεγάλο γλωσσικό μοντέλο είτε βγάζει ως έξοδο τον αριθμό της προβλεπόμενης κατηγορίας του δείγματος (χρησιμοποιώντας την κατηγοριοποίηση ακολουθίας - sequence classification της βιβλιοθήκης huggingface [22]) είτε το όνομα της κλάσης (χρησιμοποιώντας την μετατροπή ακολουθίας σε ακολουθία - sequence to sequence του huggingface). Αυτές οι δύο μέθοδοι όπως αναφέρεται στα αποτελέσματα παράγουν διαφορετικές μετρικές παρόλο που επιτελούν σχεδόν την ίδια διαδικασία.

1.2.3 Υποβοήθηση Ολίγων Παραδειγμάτων και Εκπαίδευσης με Λίγα Δείγματα

Παρόλο που οι δύο αυτές τεχνικές μοιάζουν, χαρακτηρίζονται από την εξής διαφορά: στην υποβοήθηση (prompting) το μοντέλο δεν εκπαιδεύεται, αλλά παρατηρείται βελτίωση των προβλέψεών του με την αύξηση του αριθμού παραδειγμάτων που δίνονται στην είσοδό του στο στάδιο της πρόβλεψης (inference) [2]. Η εκπαίδευση με λίγα δείγματα σημαίνει ότι το μοντέλο εκπαιδεύεται σε κάθε δείγμα. Όταν χρησιμοποιούνται και οι δύο τεχνικές ταυτόχρονα, τότε το μοντέλο εκπαιδεύεται σε λίγα δείγματα παρουσία λίγων παραδειγμάτων (δηλαδή και training και propmting).

1.3 Πειραματικό Μέρος

1.3.1 Πρακτική Παραγωγή Συνόλων δεδομένων Πίνακα με Μεροληψία

Για τα πειράματα συγκεντρώθηκαν τα σύνολα δεδομένων που χρησιμοποιήθηκαν και στο [14]. Αυτά αποτελούν μία ποικιλόμορφη επιλογή δεδομένων μορφής πίνακα που καλύπτουν πολλαπλές πραγματικές περιπτώσεις χρήσεις, όπως ιατρικά δεδομένα, δεδομένα εισοδήματος και λοιπά. Σε αυτά προστέθηκαν οι μεροληψίες που περιεγράφηκαν στο θεωρητικό μέρος.

Για τον έλεγχο της ποσότητας μεροληψίας στα πειράματα ακολουθήθηκε η εξής διαδικασία.

1. Από κάθε σύνολο δεδομένων και για κάθε είδος μεροληψίας παράχθηκαν δέκα παραλλαγές επιλέγοντας κάθε φορά κάποιο τυχαίο χαρακτηριστικό των δεδομένων το οποίο θα λάβει την μεροληψία (ή ζεύγους χαρακτηριστικών ανάλογα του είδους μεροληψίας).
2. Οι δέκα παραλλαγές ταξινομήθηκαν κατά αύξοντα αριθμό δειγμάτων που παρέμειναν στο σύνολο δεδομένων.
3. Η πρώτη (με τα λιγότερα δείγματα) αποτέλεσε την εκδοχή του συνόλου δεδομένων με υψηλή μεροληψία ενώ η πέμπτη αποτέλεσε την εκδοχή του συνόλου με μέτρια μεροληψία.

4. Τα μοντέλα αξιολογήθηκαν και στις δύο ξεχωριστά (καθώς και στην εκδοχή χωρίς μεροληψία σε κάποια πειράματα).

1.3.2 Προετοιμασία δεδομένων για τα Μεγάλα Γλωσσικά Μοντέλα

Όπως εξηγήθηκε στο θεωρητικό μέρος, για να εισαχθούν τα δεδομένα πίνακα σε μεγάλα γλωσσικά μοντέλα πρέπει να σειριοποιηθεί η κάθε γραμμή του πίνακα σε ένα κείμενο προς πρόβλεψη. Η σειριοποίηση που επιλέχθηκε είναι η απλή σειριοποίηση προτύπου κειμένου (text template) η οποία στην [14] έδωσε τα καλύτερα αποτελέσματα. Συνοπτικά, σύμφωνα με αυτήν ο τρόπος σειριοποήσης ακολουθεί το πρότυπο «Το *όνομα στήλης* είναι *τιμή στήλης*.» για κάθε στύλη ενώ για την ετικέτα: «Η ετικέτα είναι: *τιμή ετικέτας*» προτού ενωθούν όλες οι προτάσεις σε ένα κείμενο. Η τιμή της ετικέτας παραλείπεται στο σύνολο αξιολόγησης όπου θα την προβλέψει το μοντέλο και είναι παρούσα στο σύνολο εκπαίδευσης (τα οποία είναι ξένα μεταξύ τους). Η μέθοδος αυτή στα πειράματα εφαρμόστηκε στα αγγλικά που είναι και η γλώσσα των συνόλων δεδομένων που χρησημοποιήθηκαν. Τα αγγλικά έχουν και το πλεονέκτημα ότι δεν χρειάζεται να αλλάζει το άρθρο της κάθε στήλης ανάλογα του φύλου του ονόματός της.

Επιπλέον, στα πειράματα που χρησιμοποιήθηκε η τεχνική υποβοήθησης ολίγων παραδείγματων (few-shot prompting). Αυτό έγινε με την επιλογή τυχαίων δειγμάτων από το σύνολο εκπαίδευσης τα οποία σειριοποιήθηκαν με την παραπάνω μέθοδο (συμπεριλαμβανομένης της ετικέτας τους) και γράφηκαν πριν από το δείγμα αξιολόγησης (χωρίς την ετικέτα του) στην είσοδο του μοντέλου.

1.3.3 Μοντέλα

Τα μοντέλα που χρησημοποιήθηκαν συνολικά στα πειράματα ήταν τα εξής.

Μεγάλα Γλωσσικά Μοντέλα

- Το μεγάλο γλωσσικό μοντέλο mt0-base, το οποίο είναι 570 εκατομυρίων παραμέτρων [16].
- Το μεγάλο γλωσσικό μοντέλο T0_3B, το οποίο είναι 3 δισεκατομυρίων παραμέτρων [19].
- Το μεγάλο γλωσσικό μοντέλο T0pp, το οποίο είναι 11 δισεκατομυρίων παραμέτρων και αποτελεί βελτίωση του γνωστού μοντέλου T0 [19]

Από αυτά, ως βασικό χρησημοποιήθηκε το mt0 καθώς ήταν το πιο μικρό, κάτι που διευκόλυνε σημαντικά την εκτέλεση του μαζί με όλα τα σύνολα δεδομένων. Πιστεύεται ότι, παραδείγματος χάρην, στις υπολογιστικές υποδομές που παρέχονται από τον ελληνικό υπερυπολογηστή ARIS για μηχανική μάθηση, το mt0-base αποτελέι το μεγαλύτερο υπαρκτό γλωσσικό μοντέλο που μπορεί να τρέξει.

Επιπλέον, ως αντιπρόσωποι των άλλων υπερσύγχρονων μεθόδων για δεδομένα μορφής πίνακα χρησημοποιήθηκαν τα εξής μοντέλα:

- Το μοντέλο XGBoost, το οποίο είναι ένα από τα καλύτερα μοντέλα δένδρων αποφάσεων για δεδομένα πίνακα [3].
- Το TabPFN, το οποίο είναι ένα πρόσφατο μοντέλο για μικρά σύνολα δεδομένων πίνακα που επίσης χρησημοποιεί την δομή του μετασχηματιστή [15].

Σε όσα πειράματα έγινε εκπαίδευση μεγάλων γλωσσικών μοντέλων, για αυτήν χρησημοποιήθηκε η βιβλιοθήκη peft με προσαρμογή χαμηλής τάξης (LoRA) ώστε η εκπαίδευση να είναι αποδοτική και για μοντέλα πολλών παραμέτρων.

Όλα τα μοντέλα αξιολογήθηκαν ως προς την ακρίβεια των προβλέψεων τους στην καλύτερη εποχή τους.

1.3.4 Περιγραφές Πειραμάτων

Για αρχή, περιγράφεται τι έγινε στα πειράματα. Τα αποτελέσματα παρουσιάζονται στην επόμενη ενότητα.

Πείραμα up to car

Σε αυτό το πείραμα χρησημοποιήθηκε το μεγάλο γλωσσικό μοντέλο mt0 και τα XGBoost και TabPFN. Εκπαιδεύτηκε για ταξινόμηση ακολουθιών και τα σύνολα δεδομένων που χρησημοποιήθηκαν ήταν τα blood, diabetes, heart και car. Για την εκπαιδεύση και την πρόβλεψη χρησημοποιήθηκε υποβοήθηση ενός παραδείγματος (1-shot prompting) για 15 εποχές. Δοκιμάστηκαν όλα τα είδη μεροληψίας καθώς και τα αρχικά σύνολα δεδομένων.

Πείραμα 1024

Σε αυτό το πείραμα χρησημοποιήθηκε πάλι το μεγάλο γλωσσικό μοντέλο mt0 καθώς και τα XGBoost και TabPFN. Εκπαιδεύτηκε για ταξινόμηση ακολουθιών και τα σύνολα δεδομένων που χρησημοποιήθηκαν ήταν τα income, bank, jungle και calhousing. Για την εκπαιδεύση και την πρόβλεψη χρησημοποιήθηκε υποβοήθηση ενός παραδείγματος (1-shot prompting) για 15 εποχές. Δοκιμάστηκαν όλα τα είδη μεροληψίας καθώς και τα αρχικά σύνολα δεδομένων. Σε αντίθεση με πριν, τώρα χρησημοποιήθηκαν 1024 δείγματα από τα σύνολα δεδομένων. Το TabPFN είναι σχεδιασμένο για να λειτουργεί με μέχρι 1024 δείγματα. Επιπλέον, τα σύνολα δεδομένων του προηγούμενου πειράματος έχουν λιγότερα από 1024 δείγματα οπότε θα έβγαζαν τα ίδια αποτελέσματα.

Πείραμα Few Shot Training

Σε αυτό το πείραμα χρησημοποιήθηκε πάλι το μεγάλο γλωσσικό μοντέλο mt0 καθώς και τα XGBoost και TabPFN. Εκπαιδεύτηκε για ταξινόμηση ακολουθιών και τα σύνολα δεδομένων που χρησημοποιήθηκαν ήταν τα bank, jungle, calhousing, car, heart, diabetes, blood. Τώρα χρησιμοποιήθηκαν 20 εποχές και δοκιμάστηκαν μόνο οι μεροληψίες ζεύγους χαρακτηρηστικών (fpair) και συνδυασμού (combined) οι οποίες και είναι οι πιο ρεαλιστικές για πραγματικά δεδομένα αλλά και υπήρχαν ενδείξεις ότι το μεγάλο γλωσσικό μοντέλο τα πήγαινε καλύτερα. Για όλα τα μοντέλα δοκιμάστηκε εκπαίδευση με λίγα παραδείγματα (few-shot training). Οι αριθμοί δειγμάτων που χρησημοποιήθηκαν ήταν 1, 2, 4, 8, 16, 32, 64, 128, 256, 512 και 1024 όπως ακριβώς έγινε και στο [14].

Πείραμα Sequence to Sequence

Σε αυτό το πείραμα χρησημοποιήθηκε το μεγάλο γλωσσικό μοντέλο mt0 αλλά σε λειτουργία μετατροπής ακολουθίας σε ακολουθία (sequence to sequence). Εκπαιδεύτηκε για ταξινόμηση ακολουθιών και τα σύνολα δεδομένων που χρησημοποιήθηκαν ήταν τα bank, jungle, calhousing, car, heart, diabetes, blood. Χρησημοποιήθηκαν 20 εποχές και δοκιμάστηκαν οι μεροληψίες ζεύγους χαρακτηρηστικών (fpair) και συνδυασμού (combined). Δοκιμάστηκε εκπαίδευση με λίγα παραδείγματα (few-shot training). Οι αριθμοί δειγμάτων που χρησημοποιήθηκαν ήταν 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512 και 1024. Οι συνθήκες είναι παρόμοιες με το fst για να συγκριθούν η μετατροπή ακολουθίας σε ακολουθία (sequence to sequence) με την κατηγοριοποίηση ακολουθίας που χρησημοποιούταν μεχρι τώρα. Η ακρίβεια της πρόβλεψης μετρήθηκε ελέγχοντας για κάθε δείγμα αν η προβλεπόμενη λέξη είναι ακριβώς ίδια με το όνομα της ετικέτας. Επιπλέον, η χρήση μετατροπής ακολουθίας σε ακολουθία (sequence to sequence) δίνει την δυνατότητα αξιολόγησης των απευθείας προβλέψεων χωρίς υποβοήθηση (0-shot).

Πείραμα fsp

Σε αυτό το πείραμα, χρησημοποιήθηκε πάλι το μεγάλο γλωσσικό μοντέλο mt0 σε λειτουργία μετατροπής ακολουθίας σε ακολουθία (sequence to sequence) μαζί με τα T0_3B και T0pp. Όλα τα μοντέλα εκπαιδεύτηκαν για ταξινόμηση ακολουθιών και τα σύνολα δεδομένων που χρησημοποιήθηκαν ήταν τα calhousing, car, heart, diabetes, blood. Χρησημοποιήθηκαν 20 εποχές και δοκιμάστηκαν οι μεροληψίες ζεύγους χαρακτηρηστικών (fpair) και συνδυασμού (combined). Τώρα δοκιμάστηκε υποβοήθηση ολίγων παραδειγμάτων (few-shot prompting). Οι αριθμοί δειγμάτων που χρησιμοποιήθηκαν ήταν 1, 2, 4, 8, 16 και 32 δίοτι η υλοποίηση είχε υψηλές απαιτήσεις μνήμης. Η ακρίβεια της πρόβλεψης μετρήθηκε ελέγχοντας για κάθε δείγμα αν η προβλεπόμενη λέξη είναι ακριβώς ίδια με το όνομα της ετικέτας. Το πείραμα αυτό σκοπεύει στην αξιοποίηση της πρότερης γνώσης των γλωσσικών μοντέλων καθώς δεν έχει καθόλου εκπαίδευση. Για αυτό είναι σημαντικό να δοκιμαστούν και όσο το δυνατόν μεγαλύτερα γλωσσικά μοντέλα.

1.4 Αποτελέσματα

1.4.1 Αποτελέσματα Πειραμάτων

Πείραμα up to car

Τα αποτελέσματα φαίνονται στον πίνακα 4.1. Σύμφωνα με αυτά, το μεγάλο γλωσσικό μοντέλο mt0 είναι πολύ κοντά στις επιδόσεις των άλλων μοντέλων στην πλειοψηφία των περιπτώσεων μεροληψίας. Πιο συγκεκριμένα, κατα μέσο όρο ξεπερνά καθαρά τα άλλα μοντέλα στις περιπτώσεις μέτριας μεροληψίας ετικέτας και υψηλής μεροληψίας χαρακτηριστικού. Αξιοσημείωτα, στην διπλή μεροληψία χαρακτηριστικού τα πήγε αρκετά χειρότερα και από τα δυό άλλα μοντέλα τόσο στην μέτρια όσο και στην υψηλή περίπτωση. Στις υπόλοιπες βρίσκεται συνήθως μεταξύ των δύο μοντέλων και κοντά στα αποτελέσματά τους.

Πείραμα 1024

Τα αποτελέσματα για το συγκεκριμένο πείραμα βρίσκονται στον πίνακα 4.2. Σύμφωνα με αυτά, πάλι το μεγάλο γλωσσικό μοντέλο mt0 είναι κοντά στις επιδόσεις των άλλων δύο. Κατά μέσο όρο παρουσιάζει σημαντική διαφορά στην περίπτωση υψηλής μεροληψίας συνδυασμού ($> 6\%$ από το κοντινότερο) ενώ πάλι τα πάει χειρότερα τόσο στην υψηλή όσο και στην μέτρια διπλή μεροληψία χαρακτηριστικού. Στις υπόλοιπες περιπτώσεις το μεγάλο γλωσσικό μοντέλο πάλι συνήθως βρίσκεται κοντά στα άλλα δύο. Οι παρατηρήσεις αυτές είναι συνεπείς και με τις προηγούμενες, καθώς στην ουσία τρέχει στο περίπου το ίδιο πείραμα αλλά σε διαφορετικά δεδομένα.

Λόγω των παρατηρήσεων των δύο παραπάνω πειραμάτων, τα επόμενα πειράματα επικεντρώθηκαν στις μεροληψίες συνδυασμού και ζεύγους χαρακτηριστικού.

Πείραμα Few Shot Training

Τα αποτελέσματα βρίσκονται στον πίνακα 4.3 και επίσης παρατείθονται γραφικές αναπραστάσεις των κατά μέσο όρο επιδόσεων των μοντέλων στα γραφήματα 4.4.1, 4.4.2, 4.4.3 και 4.4.4.

Στις περιπτώσεις μέτριας μεροληψίας τα μοντέλα παρουσιάζουν παρόμοια συμπεριφορά. Πιο συγκεκριμένα, φαίνεται ότι το μεγάλο γλωσσικό μοντέλο mt0 τα πάει καλύτερα από τις άλλες μεθόδους για μικρό αριθμό δειγμάτων εκπαίδευσης (≤ 8), ενώ για μεγαλύτερο αριθμό δειγμάτων οι 3 μέθοδοι φαίνονται να συγκλίνουν στο ίδιο σημείο.

Στις περιπτώσεις υψηλής μεροληψίας παρατηρούμε διαφορετική συμπεριφορά από τα μοντέλα. Στην μεροληψία συνδυασμού το XGBoost οριακά παρουσιάζει βελτιώση με την αύξηση του αριθμού των δειγμάτων. Το TabPFN ανέρχεται γρήγορα, αλλά δεν δείχνει να βελτιώνεται από ένα σημείο και μετά. Το μεγάλο γλωσσικό μοντέλο mt0 μέχρι τα 4 δείγματα είναι σημαντικά καλύτερο από τα άλλα (πιθανόν λόγω πρώτερης γνώσης), στην συνέχια όμως το ξεπερνάνε σημαντικά. Προς το τέλος, το mt0 αρχίζει μία σταθερή άνοδο και ξεπερνά τελικά και τα άλλα δύο μοντέλα. Αυτό ίσως ήταν αναμενόμενο γιατί εκ φύσεως η μεροληψία συνδυασμού επηρεάζει πολύ μοντέλα δένδρων αποφάσεων, και ίσως χρειάζεται πρότερη γνώση για να ξεπεραστεί (καθώς χωρίς πρότερη γνώση ένα μοντέλο παραβλέπει μία πιθανή αιτία κατηγοριοποίησης σε κάποια ετικέτα).

Αντίθετα, στην υψηλή μεροληψία ζεύγους χαρακτηρηστικών τα τρία μοντέλα εμφανίζουν παρόμοιες τιμές στην ακρίβειά τους. Ίσως αυτό συμβαίνει διότι αυτός ο τύπος μεροληψίας δεν επηρεάζει τις ετικέτες άμεσα, και έτσι έχει μικρότερη επίδραση στις προβλέψεις.

Πείραμα Sequence to Sequence

Τα αποτελέσματα του συγκεκριμένου πειράματος παρουσιάζονται στον πίνακα 4.4. Δίνονται οι γραφικές παραστάσεις της κατά μέσου όρου ακρίβειας του μεγάλου γλωσσικού μοντέλου με χρήση παραγωγής ακολουθίας σε ακολουθία σε σύγκριση με τις ακρίβειες των XGBoost και TabPFN από το προηγούμενο πείραμα (αφού για αυτά είναι το ίδιο πράγμα). Αυτές είναι στις εικόνες 4.5.1, 4.5.2, 4.5.3 και 4.5.4.

Για αρχή, πλέον έχει νόημα και η πρόβλεψη μηδενικού αριθμού δειγμάτων (0-shot). Αυτή όμως σε κάθε περίπτωση είναι εξαιρετικά χαμηλή και το μεγάλο γλωσσικό μοντέλο παράγει λέξεις οι οποίες διαφέρουν από το όνομα της ετικέτας (ίσως αναμενόμενο γιατί το μοντέλο mt0 δεν είναι επαρκώς μεγάλο ώστε να επιλύσει το παρόν ζήτημα). Παρατηρείται βέβεια δραματική βελτίωση με εκπαίδευση μόνο ενός δείγματος (1-shot).

Το μεγάλο γλωσσικό μοντέλο πάλι τα πάει όσο καλά και τα υπόλοιπα. Σε σχέση με πριν, τώρα φαίνεται να αποδίδει λίγο χειρότερα, ειδικά σε λίγα δείγματα. Αυτό ίσως να ήταν αναμενόμενο καθώς πριν εκπαιδεύοταν στο να βγάζει απλά τον αριθμό της κατηγορίας του δείγματος, ενώ τώρα καλείται να παράξει ακριβώς το όνομα της κατηγορίας, κάτι το οποίο είναι ίσως λίγο πιο δύσκολο, και ίσως απαιτεί περισσότερη εκπαίδευση για να πετύχει υψηλότερη ακρίβεια.

Αξιοσημείωτο είναι το γεγονός ότι πλέον το μεγάλο γλωσσικό μοντέλο στα 1024 δείγματα ξεπερνά τα άλλα δύο σχεδόν σε όλες τις περιπτώσεις, κάτι που δεν συνέβαινε πριν. Ακόμα πιο εμφανές γίνεται αυτό στην περίπτωση υψηλής μεροληψίας συνδυασμού, όπου από τα 128 έως και τα 1024 δείγματα βρίσκεται πάνω από τα άλλα δύο, και σε σχέση με το TabPFN, το δεύτερο καλύτερο, περισσότερο από 5% στα 512 και 4% στα 1024. Επίσης ξεπερνά και τα άλλα δύο μοντέλα στα 4 δείγματα. Οι λόγοι που αποδίδει τόσο καλά στην μεροληψία συνδυασμού αναπτύχθηκαν προηγουμένως, και αυτό ίσως είναι το πιο εμφανές παράδειγμα στο οποίο το μεγάλο γλωσσικό μοντέλο έχει τη δυνατότητα να ανακάμψει από την μεροληψία των δεδομένων περισσότερο σε σχέση με τα άλλα δύο.

Πείραμα fsp

Τα αποτελέσματα του πειράματος παρουσιάζονται, στον πίνακα 4.5. Για αρχή τα δεδομένα δεν είναι πλήρη, καθώς κατλεστη αδύνατο να τρέξουν κάποια πειράματα λόγω περιορισμών μνήμης της κάρτας γραφικών. Παρόλα αυτά, τα συγκεκριμένα αποτελέσματα αρκούν για την εξαγωγή κάποιων συμπερασμάτων. Όπως φαίνεται, όσο μεγαλύτερο είναι το γλωσσικό μοντέλο, τόσο καλύτερη ακρίβεια πετυχαίνει υπό τις ίδιες συνθήκες. Όπως ήταν αναμενόμενο, ύπαρχει δραματική αύξηση στην πρόβλεψη μηδενικής υποβοήθησης (zero-shot prompting). Σε γενικές γραμμές, με την άυξηση των παραδειγμάτων υποβοήθησης (prompt) παρατηρείται αύξηση της επίδοσης. Αξιοσημείωτο είναι το γεγονός ότι το μοντέλο των 11 δισεκατομυρίων παραμέτρων σε κάποιες περιπτώσει φαίνεται να μειώνει την ακρίβειά του στην μεροληψία συνδυασμού, τόσο στην υψηλή όσο και στην μέτρια. Ίσως αυτο συμβαίνει γιατί τελικά αν είναι επαρκώς μεγάλα, με prompting τα μεγάλα γλωσσικά μοντέλα γίνονται επιρρεπή σε μεροληψία συνδυασμού. Παρόλαυτά οι επιδόσεις ακόμα και των μεγαλύτερων γλωσσικών μοντέλων μόνο με prompting δεν καταφέρνουν να φτάσουν τις προηγούμενες που είχαν εκπαίδευση λίγων δειγμάτων και μάλιστα βρίσκονται και χαμηλότερα από τις «κλασσικές» μεθόδους.

1.4.2 Επιπλέον Πορίσματα

Ακόμη και το μεγάλο γλωσσικό μοντέλο ενδέχεται να μάθει την μεροληψία των δεδομένων. Αυτό ίσως είναι αναμενόμενο. Ως αποτέλεσμα των μετρήσεων που λήφθηκαν μπορούν να παρουσιαστούν διάφορες γραφικές παραστάσεις που το επιδεικνύουν. Δύο στις οποίες αυτό φαίνεται καλά είνα οι εικόνες 4.7.1 και 4.7.2. Και οι δύο περιπτώσεις προέρχονται από πειράματα υψηλής μεροληψίας όπου και ίσως φαίνεται καλύτερα αυτό το φαινόμενο. Σε αυτές το μεγάλο γλωσσικό μοντέλο ενώ αρχικά αυξάνει την ακρίβειά του με τον αριθμό των εποχών, φτάνει σε ένα οξύ μέγιστο για μία εποχή και μετά η ακρίβειά του καταρρέει είτε γρήγορα ή πιο αργά χωρίς να μπορέσει ποτέ να επανέλθει καθώς πλέον μαθαίνει το σύνολο εκπαίδευσης το οποίο δεν είναι αντιπροσωπευτικό.

Το μεγάλο γλωσσικό μοντέλο σε λειτουργία παραγωγής ακολουθίας από ακολουθία (sequence2sequence) μαθαίνει καλύτερα με περισσότερα δείγματα απ' ό,τι σε λειτουργεία κατηγοριοποίησης ακολουθίας (sequence classification) όπου αποδίδει καλύτερα με χρήση few-shot. Όπως αναφαίρθηκε και προηγουμένως, αυτό ίσως οφείλεται στο ότι για την σωστή πρόβλεψη του στην κατηγοριοποίηση χρειάζεται απλά να επιλέξει τον αριθμό της κατηγορίας ενώ για σωστή προβλεψη στην παραγωγή ακολουθίας, χρειάζεται να παράξει ακριβώς σωστά το όνομα της κατηγορίας. Έτσι, ίσως είναι πιο δύσκολο το δεύτερο σενάριο, υπό την έννοια ότι χρειάζεται περισσότερα δείγματα για να πετύχει. Αξιοσημείωτο είναι ότι στην παραγωγή ακολουθίας, υπό κάποιες περιπτώσεις μεροληψίας τα πηγαίνει σημαντικά καλύτερα απότι με κατηγοριοποίση ακολουθίας, ενώ για μικρό αριθμό δειγμάτων συνήθως η κατηγοριοποίηση ακολουθίας τα πάει καλύτερα στην πλειοψηφεία των περιπτώσεων.

Η εκπαίδευση με λίγα παραδείγματα (few-shot training) επιτυγχάνει καλύτερα αποτελέσματα από ένα σημείο και μετά σε σχέση με την υποβοήθηση ολίγων παραδειγμάτων (few-shot prompting). Η χρήση και των δύο μαζί δίνει εν τέλει καλύτερα αποτελέσματα από καθένα ξεχωριστά.

1.5 Συμπεράσματα

Το γλωσσικό μοντέλο ακολουθεί τις μεθόδους που αποτελούν την τελευταία λέξη της τεχνολογίας ή και κάποιες φορές τις ξεπερνά, τόσο σε περιπτώσεις υψηλής όσο και μεσαίας παρουσίας μεροληψίας στο σύνολο εκπαίδευσης. Αναμενόμενα, στην πλειοψηφία των περιπτώσεων ξεπερνά το XGBoost που βασίζεται σε δέντρα αποφάσεων λόγω των μειονεκτημάτων τους που αναφέρθηκαν προηγουμένως. Ακόμη, το TabPFN που επίσης βασίζεται στην δομή του μετασχηματιστή συχνά ξεπερνά το XGBoost.

Στην διπλή μεροληψία χαρακτηριστικού το μεγάλο γλωσσικό μοντέλο τα πάει εμφανώς χειρότερα από τα άλλα δύο μοντέλα κατά μέσο όρο σε όλα τα πειράματα.

Το μεγάλο γλωσσικό μοντέλο φαίνεται να μπορεί να ξεπεράσει την μεροληψία των δεδομένων περισσότερο από τα άλλα δύο στις περιπτώσεις μεροληψίας συνδυασμού και ζεύγους χαρακτηριστικού.

Chapter 2

Introduction

Contents

2.1 TabLLM

In recent advancements of machine learning, the integration of large language models (LLMs) into various domains has shown significant promise. One such domain is the processing and classification of tabular data, which has traditionally been the realm of specific algorithms such as decision trees and gradient boosting machines. The analysis of tabular data, which remains one of the most pervasive forms of data in both industrial and research contexts, has traditionally relied on classical machine learning models such as decision trees, random forests, and gradient boosting machines. These models are adept at handling structured data with fixed schemas but often fall short in leveraging the rich representational capabilities of modern neural architectures. The advent of transformers, particularly in the realm of natural language processing (NLP), has revolutionized the way sequential and contextual data are processed, leading to significant performance improvements in a variety of tasks. Extending this transformative power to tabular data is the primary motivation behind the development of TabLLM (Tabular Large Language Model). The TabLLM framework emerges as a pioneering method to bridge the gap between tabular data and LLMs, offering a comprehensive solution to serialize and fine-tune these models for improved performance in few-shot learning scenarios [14].

2.1.1 Background and Motivation

Tabular data is characterized by its structured format, consisting of rows and columns, where each column represents a distinct feature and each row an individual record. This format is ubiquitous in databases, spreadsheets, and many other data repositories. However, the fixed schema and the often heterogeneous nature of the data types (numerical, categorical, etc.) pose unique challenges for direct application of deep learning models designed for unstructured data. Classical models like decision trees and gradient boosting have been highly successful due to their simplicity and interpretability, but they lack the capacity to model complex feature interactions without extensive feature engineering [21].

In contrast, transformers have shown remarkable capability in modeling sequential data through self-attention mechanisms, capturing long-range dependencies and contextual relationships. This has led to their dominance in NLP tasks, from language translation to text generation. The success of transformers in these domains suggests potential benefits for tabular data if adapted appropriately. TabLLM aims to bridge this gap by leveraging transformer architectures to handle the intricacies of tabular data, thus providing a unified framework for both predictive and generative tasks [9].

2.1.2 TabLLM Key Contributions

TabLLM proposes a comprehensive framework that adapts the transformer architecture to the specific needs of tabular data analysis. By integrating mechanisms to handle both numerical and categorical data efficiently, it offers a versatile tool capable of tackling a wide range of tabular tasks.

The core innovation lies in the adaptation of self-attention mechanisms to capture inter-column dependencies. Unlike classical models that treat each feature independently, TabLLM leverages self-attention to understand and model interactions between different features, leading to potentially richer and more nuanced representations.

Inspired by the success of pre-training in NLP, TabLLM employs a two-stage training process. Initially, the model undergoes unsupervised pre-training on a large corpus of tabular data to learn generic feature representations. This is followed by supervised fine-tuning on specific tasks, allowing the model to specialize and achieve higher performance on those tasks [21, 5].

Preliminary evaluations have demonstrated that TabLLM not only matches but often surpasses the performance of traditional models and other neural network-based approaches on benchmark datasets [12, 18, 13]. Its ability to handle missing data, imbalanced classes, and varied data types makes it a robust choice for real-world applications [14].

2.1.3 Way of Operation

The core innovation of TabLLM lies in its ability to serialize tabular data into natural language formats that LLMs can effectively process. This involves converting table entries into coherent textual representations

while possibly embedding task-specific cues. The framework explores various serialization techniques, such as text-template serialization, feature combination, and even LaTeX-based serialization. Each technique is aimed at optimizing the LLM's understanding and processing of tabular information.

For instance, text-template serialization, which noted the highest performance according to the researchers, transforms table columns into descriptive sentences, creating a natural flow of information that LLMs can easily comprehend. This method not only enhances the model's interpretability but also respects the token limits of LLMs, ensuring efficient data handling. Additionally, the framework incorporates feature combination techniques to reflect the interrelationships between different data features more naturally, further improving the model's predictive capabilities.

TabLLM also employs advanced fine-tuning methods, such as T-few, which is a parameter-efficient technique allowing the model to adapt to varying amounts of data with minimal computational overhead. This fine-tuning is crucial for achieving high performance across different datasets and shot levels, making TabLLM particularly effective in few-shot learning contexts.

Experimental results have demonstrated that TabLLM significantly outperforms traditional models like XGBoost, especially in zero-shot and few-shot settings. For example, LaTeX serialization has shown remarkable performance in zero-shot scenarios, highlighting the potential of structured data formats in enhancing LLM capabilities.

The implementation code is available on GitHub.

2.2 Our Work

In this context, our research builds upon the methodologies presented in the TabLLM paper. We construct a similar framework that utilizes text-template serialization and parameter-efficient fine-tuning. Instead of employing T-few, we leverage Low-Rank Adaptation (LoRA), a technique known for its efficiency in fine-tuning large language models (LLMs), which is particularly useful in environments with limited computational resources. Our primary objective is to evaluate the impact of dataset bias and to investigate whether LLMs can effectively mitigate this bias, in contrast to traditional methods, which are notorious for their inability to do so.

To this end, we primarily compare the performance of LLMs with traditional (non-LLM) methods such as XGBoost and the recent TabPFN. We mostly test smif enabled, swiping will not clamp at the neighboring workspaces but continue to the further ones.aller LLMs to assess their capability in handling these tasks, aiming to determine whether they can achieve comparable performance to larger models, especially in the context of mitigating dataset bias, while benefiting from reduced computational requirements. However, we also include tests on larger LLMs to provide a comprehensive evaluation of performance across different model sizes.

2.3 Methods Comparing Against: TabPFN

TabPFN is a novel machine learning method specifically designed to address small tabular classification problems efficiently and effectively. Introduced by researchers, including those from the AutoML community, TabPFN leverages the power of Transformers, a neural network architecture primarily known for its success in natural language processing, to perform rapid and accurate tabular data classification.

TabPFN is not just another machine learning model; it is a meta-learned algorithm. This means that it has been trained to learn from a wide variety of datasets and generalize this learning to new datasets quickly. Unlike traditional models that require extensive training on each new dataset, TabPFN performs inference in a single forward pass, making it extremely fast.

One of the standout features of TabPFN is its ability to approximate Bayesian inference. Bayesian methods are highly regarded for their ability to handle uncertainty and incorporate prior knowledge into the model. TabPFN incorporates a prior that emphasizes simplicity and causal relationships, leading to more interpretable and robust predictions.

The training process for TabPFN is conducted offline. This involves generating millions of synthetic datasets based on a predefined prior and training the model to predict outcomes for these datasets. This prior is based on structural causal models (SCMs) and Bayesian neural networks (BNNs), which encode relationships and patterns found in real-world tabular data. This extensive pre-training enables TabPFN to make accurate predictions on new datasets without additional training [1, 4].

TabPFN is designed to handle small datasets efficiently. It has been tested on datasets with up to 1,000 training examples, 100 features, and 10 classes. The results show that TabPFN outperforms traditional methods like boosted trees and is competitive with state-of-the-art AutoML systems. Furthermore, it achieves this with significant speed advantages, providing up to a 230x speedup compared to traditional methods and even higher when utilizing GPUs [15].

A significant advantage of TabPFN is that it requires no hyperparameter tuning. This is a critical feature for practitioners who need quick and reliable models without the overhead of extensive hyperparameter searches, which are often computationally expensive and time-consuming [15].

2.4 Zero and Few-Shot Prompting

2.4.1 Background and Significance

In recent years, the field of natural language processing (NLP) has witnessed remarkable advancements, primarily driven by the development of large-scale pre-trained language models such as GPT-3 by OpenAI, BERT by Google, and T5 by Google Research. These models have demonstrated unprecedented capabilities in understanding and generating human-like text across a multitude of tasks, ranging from text completion and translation to question answering and summarization. Central to the performance of these models is their ability to leverage vast amounts of textual data during the pre-training phase, enabling them to acquire a nuanced understanding of language structure, semantics, and context.

2.4.2 The Concept of Prompting

Prompting is a technique used to guide pre-trained language models to perform specific tasks by providing them with carefully designed input sequences or "prompts." Unlike traditional supervised learning approaches that require extensive task-specific labeled data for fine-tuning, prompting leverages the inherent knowledge encoded within pre-trained models, allowing them to generate appropriate responses based on minimal or no additional training data. This paradigm shift has profound implications for the accessibility, efficiency, and scalability of NLP applications.

2.4.3 Zero-Shot Prompting

Zero-shot prompting refers to the capability of a language model to perform a task without any explicit task-specific training or fine-tuning. In a zero-shot setting, the model relies entirely on the general knowledge acquired during pre-training to interpret the prompt and generate a relevant response. For instance, a model can be prompted with "Translate the following English sentence to French: 'Hello, how are you?"' and produce the correct translation despite not being explicitly trained on translation tasks. This approach is particularly advantageous in scenarios where labeled data is scarce or unavailable, offering a versatile solution for a wide array of applications.

The potential of zero-shot prompting lies in its ability to generalize across tasks and domains, making it a powerful tool for addressing novel or infrequent queries. However, its effectiveness is heavily dependent on the quality and specificity of the prompts, as well as the comprehensiveness of the pre-trained model's knowledge base. Researchers and practitioners are continuously exploring innovative prompting strategies to enhance the accuracy and reliability of zero-shot responses.

2.4.4 Few-Shot Prompting

Few-shot prompting builds upon the zero-shot paradigm by providing the model with a small number of task-specific examples within the prompt. This technique bridges the gap between zero-shot and fully supervised

learning, offering a middle ground that leverages the strengths of both approaches. In few-shot prompting, the model is presented with a prompt containing a handful of example input-output pairs, followed by a query for which the model must generate a response.

For example, a few-shot prompt for a sentiment analysis task might look like this:

```
Review: "The movie was fantastic, I loved it."
Sentiment: Positive

Review: "The film was boring and too long."
Sentiment: Negative

Review: "The plot was engaging, but the acting was mediocre."
Sentiment:
```

The model, having seen the examples, is expected to infer the sentiment of the third review based on the patterns demonstrated in the provided examples. Few-shot prompting significantly enhances the model's performance by giving it a clearer understanding of the task requirements and expected outputs, thus reducing ambiguity and improving accuracy.

2.4.5 Challenges

While zero and few-shot prompting offer exciting opportunities for leveraging pre-trained language models, they also present several challenges. Designing effective prompts is a non-trivial task that requires a deep understanding of both the model's capabilities and the specific nuances of the task at hand. Suboptimal prompts can lead to poor performance, highlighting the need for systematic methods to optimize prompt construction.

Moreover, the interpretability of model responses in zero and few-shot settings remains an area of active research. Understanding how and why a model arrives at a particular output is crucial for building trust and reliability in NLP systems, especially in high-stakes applications such as healthcare, finance, and legal domains.

Zero and few-shot prompting represent transformative approaches in the realm of natural language processing, offering innovative solutions to the challenges of task-specific training and data scarcity. By harnessing the latent knowledge within pre-trained language models, these techniques enable the development of flexible, efficient, and scalable NLP systems.

Chapter 3

Proposal

Contents

Now, the model built shall be presented along with ideas on bias.

3.1 Contributions

3.1.1 Intentionally Creating Bias in Datasets

In order to evaluate the effects of bias in the models under examination biased datasets are required. However, because most datasets are created in ways that intentionally avoid bias and because it is hard to detect bias in an existing dataset a different approach is needed if we wish to come into possession of biased datasets. Also, perhaps most importantly, the test set would need to be unbiased so that the evaluation can show how bias affects the model. Thus, the only way of obtaining biased datasets is to, starting with an assumed unbiased dataset, alter it in ways that induce bias. Since the TabLLM researchers already created a good collection of tabular datasets from various domains with various features [14], we would like to utilize it as the original unbiased datasets to avoid having to do the same thing from the scratch. For this purpose five kinds of bias were invented which will be explained now.

Label Bias

Label bias is the simplest form of bias, albeit the bias created is quite artificial. A dataset is altered with label bias when all the data with exhibiting one specific value in their label are dropped. This kind of bias is thought that it can easily confuse decision tree methods which rely entirely on observing a label to be able to predict it. In contrast, large languge models are thought to be less affected by it because they can statistically predict the label as text.

Feature Bias

Feature Bias is the translation of label bias to features of the dataset. This bias is believed to be very realistic because when collecting data it often happens that an entire group of similar data points is accidentally ignored, goes missing or simply doesn't exist in the first place. Feature Bias is present when from a dataset all elements exhibiting a specific value in a specific feature are dropped. That way certain common situations can be represented, like for instance the fact that healthy people don't do as many medical exams and the data are skewed.

Combined Bias

Combined Bias is encountered when a specific combination of the value of a feature and a label is dropped from the dataset. It is a combination of the above two kinds of bias and decision tree methods should be particularly susceptible to it for similar reasons to label bias. Additionally, because it is also akin to feature bias, it should also be more realistic than label bias, however it still remains artificial.

Fpair Bias

Fpair bias (feature pair bias) is the translation of combined bias to only concern features. It is also an evolution of simple feature bias in which instead of only one feature deciding if a sample is to be dropped, a pair of two features determines that instead. To be more precise, the samples dropped are the ones exhibiting a specific pair of values in two distinct features. Naturally, less samples are usually dropped compared to feature bias therefore making the dataset "less biased" (more on that later) and at the same time, this kind of bias, more realistic.

Double Feature Bias

This kind of bias is equivalent to applying feature bias twice (and therefore for two distinct features). It is also dual to Fpair bias in a way, because now instead of selecting two values from distinct features and dropping their logical conjunction (as was the case in fpair), their logical disjunction is dropped. Viewed set-theoretically, this means dropping the union of two sets, each containing samples with one value in a feature, whereas in fpair the intersection of these sets were dropped. Consequently it is a stronger bias compared to both fpair and feature bias because it drops more samples (and also because it's literally equivalent to applying a previous bias twice) and it might be lie more towards artificial rather than realistic.

3.1.2 Controlling the Amount of Bias

Additionally, there is a need for a mechanism to control the amount of bias added to a dataset. For instance, the choice of one feature value to drop might lead to few data points being dropped or many depending on how predominant that value is among the (unbiased) dataset. It is reasonable to assume that starting from an unbiased dataset, dropping fewer samples gives a dataset with less bias than when dropping a lot of samples.

Accounting for the above observation, the following bias control mechanism is proposed. Starting with an unbiased dataset and a kind of bias to add as described above, create a number of different datasets with that kind of bias and order them by the number of samples remaining in them. Thus, a measure of bias is created for each biased dataset according to its order and by selecting an order, one can control how much bias is incurred in the original dataset. For our experiments 10 biased variations were generated and the median with respect to number of samples along with the one with the least samples were selected to represent environments with mid and high bias respectively. The high bias variant may not be practical but it should help with giving intuition as to how models behave under bias and the mid bias can likely function as a baseline for a substantial real bias.

To the best of our knowledge there is no work related to anything presented in the two previous subsections, i.e. adding bias to a dataset and selecting the intensity of that bias.

3.1.3 Why Language Modeling May Be More Apt for Tabular Data Classification

A known property of decision tree and forest methods is that while they can make use of the semantics of numerical features of the dataset, they cannot do so for the categorical features. In fact, the only semantics they use with regard to categorical features is whether they are the exact same or not. This is in contrast to numerical features, from which algebraic properties are easily deduced by these methods and more complicated predicates can be made, such as ones including the notions of larger or lesser and algebraic combinations like products, sums, divisions, scaling, etc.

Language modeling can come into play here by offering a way to learn the algebraic properties of categorical features, which hitherto are considered simply as strings of text and therefore their properties elude machine learning methods. First of all, via language modeling these categorical features can be translated into an algebraic vector space (using embeddings). Additional semantics can be discovered via the use of large language models and in particular via use of their prior knowledge which can enhance the simple vector space embedding. Therefore, a machine learning model for tabular data which makes use of language modeling and even more so, large language models, is in a better position to interpret the samples and understand them compared to simple decision trees or forests and should be able to make more accurate predictions. This was experimentally demonstrated in some recent works like [14].

3.1.4 Why Language Modeling May Be Able to Overcome Tabular Data Bias

Similarly as above, large language models may have the ability to overcome data bias. They have been pretrained and have managed to accumulate some knowledge about language semantics in general. If they were pretrained on an unbiased dataset then they can utilize this knowledge to give accurate predictions even when the tabular dataset they are inferring has bias.

3.2 Model

The model used for our experiments was inspired by the one used in the TabLLM paper [14]. It was not possible to utilize the actual TabLLM framework as it relies on the T-few framework which has not been updated in three years and was written in a version of python that is quite outdated by now, to the point where its dependencies no longer work (and therefore T-few has now passed on from the world of actual ideals to the world of ideals soon to be consigned to eternal oblivion, where it rightfully belongs alongside its conceivers). Therefore a new, similarly functioning framework was written from scratch for the purposes of the experiments and will be described here.

3.2.1 Tabular Serialization

The first step of the inference process is the serialization of tabular data samples to text that can be inputted in the large language model. To this end, the text template serialization method was used, in part because out of all serialization methods used in the TabLLM paper, this was found to yield the highest accuracy [14].

In short, it works in the following way. A template for the dataset is used which is of the form "The *column name* is *column value*." for columns expressing features. It is filled with the column values of the sample being transformed for each feature column. These are concatenated into a small paragraph and are followed by a task specific prompt like "The label is" [14]. In the case where the sample is to be used not for prediction but rather for training, the information of the label can also be included last with a sentence like "The label is *label value*". Furthermore, the last format is additionally used for few-shot prompting (which will be mentioned later).

3.2.2 Few-Shot Prompting

Optionally, the few-shot prompting technique can be used. It works according to what was described in the introduction chapter, that is by adding a few examples with their labels, taken from the train set, to the sample which is being predicted from the test set. These shots are added before the text of the sample. Few-shot prompting is known to be able to increase the accuracy of large language model predictions for various tasks. If few-shot prompting is not used, then the model is run in zero-shot mode for each sample.

3.2.3 Few-Shot Training

Also optionally, the few-shot training technique can be used. It works by training in a few samples of the train set instead of the whole train set before evaluating predictions across the whole test set. This was used in the TabLLM paper [14]. Naturally, it can be used along few-shot prompting and as we will see in the results of the experiments, using both likely gives the best results.

3.2.4 Fine-Tuning

As stated before the original TabLLM framework used T-few for fine-tunning [14]. For our model, LoRA was used instead. In more detail, the LoRA implementation of the peft library was used with a small LoRA layer. This might result in slightly lower prediction accuracy than T-few but it offers greatly improved training time, and entirely avoids the troubles of setting up T-few. All training mentioned before is in fact done via fine-tuning to reduce train time and costs.

3.2.5 LLMs

The main ingredient in our model is the actual large language model. It becomes evident with the model description so far that the model is designed in such a way to be able to use any large language model. For our experiments three models were used, each for different reasons. All of them are related in some way to the T0 model which was also used by the original TabLLM [14].

The first large language model employed was mt0-base, a 570M parameters model [16]. It offers many advantages. For one, it is the largest model that can fit in the GPUs that are offered by cloud services such as kaggle and also in the GPUs of the supercomputer Aris. As such, it makes running experiments with it significantly easier than with larger models and so more complicated experiments may be run.

The next large language model used was T0_3B, a 3B parameters model [19]. It is a 3B version of the T0 model and offers a balance between performance and model size.

The last large language model used was T0pp, an 11B parameters model [19]. It is an improvement of the T0p model which itself is an improvement of the T0 model. This is the largest model tested.

3.2.6 Model Output

Finally, there has to be a way to convert the output of the large language model, which is a token or a sequence of tokens, into a prediction for the label. There were two easily implementable methods to do this

which were both tested in specific experiments with the purpose of comparing their performance under bias.

The first one is to use a small classification head on top of the model output, which will be trained to classify the model's output into a number representing one of the possible labels of the tabular dataset. This method is automatically employed by huggingface's transformers library in its `*ForSequenceClassification` classes which were used in our implementation.

The other way to convert LLM output to a label is to take the output tokens, translate them into text and compare this text with the label. There are many ways to compare text, especially with the intention of creating an error function, however the one used was simply to consider incorrect all output which was not identical to the label's text. TabLLM used a similar mechanism [14]. Thus the large language model would have to output just the textual representation of the predicted label, a task that seems harder than outputting something which will go through a classification head. However it may lead to the model learning the task better or learning to use language and its semantics to make prediction better. One interesting property of this technique is that it can be used for zero-shot predictions as there is no classification head to train. More on all of that in the experiments section which follows.

Chapter 4

Experiments

Contents

In this chapter the experiments run will be described and their results presented.

4.1 General

For the purpose of testing the ability of large language models to resist data bias several experiments were conducted. In most experiments the goal was to compare the LLM with a "traditional" method for tabular data, like decision trees for instance, and see if the LLM performs better. To this aim, the metric chosen for the evaluation of the models was accuracy and the models were evaluated on several biased datasets.

The biased datasets used for the experiments were produced by using the methods to add bias to a dataset that were outlined in the previous chapter, on the datasets used in the TabLLM paper. These datasets represent a varied choice of different real world datasets from different domains like healthcare, insurance, banking, etc [14]. They also vary in size with approximately half of them being small (less than 2000 samples) and the other half large (over 9000).

The non-LLM methods tested were XGBoost and TabPFN. XGBoost is the state of the art decision tree model as of today and is a good representative of the decision tree methods [3]. TabPFN is a more recent creation and is quite suited for the tasks of few-shot learning (without bias) as it is designed for small datasets [15]. Furthermore, it utilizes the transformer architecture which brings it closer to LLMs as a model.

Unless otherwise stated, a huggingface sequence classification class was used for converting the LLM's output to a label prediction.

More details for each experiment along with presentation of its results will follow separately.

4.2 Experiment up-to-car

4.2.1 Description

The first experiment carried out, it involved testing the smaller mt0 model on the datasets blood, diabetes, heart and car. Diligent readers may notice that these are the smaller datasets in our disposal. Of these datasets the model was tested in all bias variations previously discussed, high and mid, as well as at their original, non-biased versions. One-shot prompting was employed in all cases in addition to fifteen epochs of training, among which, the best was selected to represent the large language model's performance. Along with mt0 the XGBoost and TabPFN models were also evaluated in the same tasks to compare how well these methods work against biased datasets. The detailed results can be seen below in table 4.1 along with averages on the kind of bias:

Dataset	LLM (mt0)	XGBoost	TabPFN
biased_datasets/car_no_bias/car	97.40%	99.42%	98.27%
biased_datasets/car_label_bias/car	93.35%	96.24%	95.38%
biased_datasets/car_feature_bias/car	91.33%	95.95%	95.66%
biased_datasets/car_combined_bias/car	92.49%	96.24%	98.27%
biased_datasets/car_fpair_bias/car	92.20%	98.55%	97.40%
biased_datasets/car_doublef_bias/car	91.62%	97.11%	96.82%
biased_datasets/heart_no_bias/heart	85.33%	86.41%	87.50%
biased_datasets/heart_label_bias/heart	54.89%	54.89%	
biased_datasets/heart_feature_bias/heart	86.96%	86.41%	86.41%
biased_datasets/heart_combined_bias/heart	88.59%	88.59%	89.67%
biased_datasets/heart_fpair_bias/heart	86.96%	88.59%	89.67%
biased_datasets/heart_doublef_bias/heart	56.52%	83.15%	84.78%
biased_datasets/diabetes_no_bias/diabetes	74.03%	79.87%	77.92%
biased_datasets/diabetes_label_bias/diabetes	36.36%	1.42%	36.36%
biased_datasets/diabetes_feature_bias/diabetes	74.68%	70.13%	80.52%
biased_datasets/diabetes_combined_bias/diabetes	74.68%	72.08%	77.27%
biased_datasets/diabetes_fpair_bias/diabetes	75.97%	74.03%	77.92%
biased_datasets/diabetes_doublef_bias/diabetes	62.34%	70.78%	73.38%
biased_datasets/blood_no_bias/blood	74.00%	75.33%	76.67%
biased_datasets/blood_label_bias/blood	79.33%	79.33%	
biased_datasets/blood_feature_bias/blood	82.00%	72.67%	82.00%
biased_datasets/blood_combined_bias/blood	78.67%	78.00%	82.67%
biased_datasets/blood_fpair_bias/blood	78.00%	81.33%	81.33%
biased_datasets/blood_doublef_bias/blood	71.33%	72.00%	75.33%
high-bias-datasets/car-feature-bias/car	93.06%	99.13%	85.26%
high-bias-datasets/car-combined-bias/car	84.39%	88.15%	91.91%
high-bias-datasets/car-fpair-bias/car	90.46%	97.40%	93.93%
high-bias-datasets/car-doublef-bias/car	73.12%	80.64%	80.35%
high-bias-datasets/heart-feature-bias/heart	78.26%	75.00%	79.89%
high-bias-datasets/heart-combined-bias/heart	54.35%	55.98%	55.43%
high-bias-datasets/heart-fpair-bias/heart	85.87%	87.50%	86.41%
high-bias-datasets/heart-doublef-bias/heart	55.43%	79.35%	80.98%
high-bias-datasets/diabetes-feature-bias/diabetes	73.38%	72.73%	73.38%
high-bias-datasets/diabetes-combined-bias/diabetes	75.97%	75.97%	79.22%
high-bias-datasets/diabetes-fpair-bias/diabetes	77.92%	73.38%	77.92%
high-bias-datasets/diabetes-doublef-bias/diabetes	64.29%	72.73%	74.03%
high-bias-datasets/blood-feature-bias/blood	79.33%	76.67%	82.67%
high-bias-datasets/blood-combined-bias/blood	80.00%	61.33%	72.67%
high-bias-datasets/blood-fpair-bias/blood	77.33%	74.00%	78.00%
high-bias-datasets/blood-doublef-bias/blood	76.00%	62.00%	78.00%
Average no bias	82.69%	85.26%	85.09%
Average label bias	65.99%	57.97%	65.87%
Average feature bias (mid)	83.74%	81.29%	86.15%
Average combined bias (mid)	83.60%	83.73%	86.97%
Average fpair bias (mid)	83.28%	85.63%	86.58%
Average doublef bias (mid)	70.45%	80.76%	82.58%
Average feature bias (high)	81.01%	80.88%	80.30%
Average combined bias (high)	73.68%	70.36%	74.81%
Average fpair bias (high)	82.90%	83.07%	84.07%
Average doublef bias (high)	67.21%	73.68%	78.34%

Table 4.1: Model accuracy for each dataset, small datasets

4.2.2 Observations

First of all, as expected on average all models perform better when no bias is present in the dataset than when there is any kind of bias and they also perform better when the bias is in mid intensity compared to when it is in high. Additionally label bias and doublef bias, both considered more of an artificial kind of bias than one naturally occurring as explained previously, seem to impact all models the hardest and lower their accuracy more than the other kinds of bias.

Remarkably, in the case of no bias and on average the large language model performs close to the other two methods though this was to be expected because of the TabLLM paper [14]. Furthermore, in the case of label bias, on average, the large language model surpasses both other methods although TabPFN only by a margin. From the above two points it is evident that indeed large language models do better than other methods when the dataset is biased. However, as stated before, since label bias is considered more of an artificial kind of bias, excluding the next experiment which is similar to this one except it concerns the larger datasets, focus will shift towards the other kinds of bias and in particular to combined and fpair bias.

With regards to feature bias, the large language model on average only outperforms XGBoost when the bias is in mid intensity but in high intensity it manages to surpass both other methods. Interestingly, applying feature bias twice (that is, using doublef bias) the large language model is shattered as it perfoms under 10% lower than TabPFN and 5 to 10% lower than XGBoost in both mid and high bias intensity. This has not been explained somehow.

In the more realistic kinds of bias, that is in combined bias and fpair bias, the large language model appears to be on par with the other two methods although it performs slightly worse. If a larger model was tested, it is likely that it would perform even better and possibly outperform the other two methods.

4.3 Experiment 1024

4.3.1 Description

This experiment is more of a continuation of the previous one with a few changes to accustom it to the larger datasets. Most prominently, all models were trained with 1024 samples of each train set for various reasons such as it sped up training that would have otherwise taken absurd amounts of time to just a few minutes. The main reason this was done however has already been mentioned and observant readers may have already picked up on it. It is that the TabPFN model is designed and only works for small datasets with up to 1024 samples. The datasets used in the previous experiment all had less than 1024 samples in their train set and the ones used in this one have more. Thus the separation of these two experiments was due to more of a technical reason than any other.

Excluding this important detail, all other parameters of this experiment are the same as previously. That is, one-shot prompting was employed in all cases in addition to fifteen epochs of training for the large language model and it was compared wit h the other two methods. The results are shown below, in table 4.2:

Dataset	LLM (mt0)	XGBoost	TabPFN
biased_datasets/income_no_bias/income	83.59%	83.89%	83.69%
biased_datasets/income_label_bias/income	78.52%	78.52%	
biased_datasets/income_feature_bias/income	77.54%	81.64%	82.03%
biased_datasets/income_combined_bias/income	78.22%	78.13%	78.32%
biased_datasets/income_fpair_bias/income	84.18%	83.11%	84.18%
biased_datasets/income_doublef_bias/income	83.01%	81.84%	82.13%
biased_datasets/bank_no_bias/bank	89.65%	89.16%	88.87%
biased_datasets/bank_label_bias/bank	88.67%	88.67%	
biased_datasets/bank_feature_bias/bank	88.57%	88.96%	89.06%
biased_datasets/bank_combined_bias/bank	88.67%	88.28%	89.55%
biased_datasets/bank_fpair_bias/bank	89.84%	87.30%	88.77%
biased_datasets/bank_doublef_bias/bank	89.16%	88.28%	89.06%
biased_datasets/jungle_no_bias/jungle	74.61%	61.28%	82.71%
biased_datasets/jungle_label_bias/jungle	52.25%	52.25%	
biased_datasets/jungle_feature_bias/jungle	70.80%	85.16%	84.28%
biased_datasets/jungle_combined_bias/jungle	50.68%	79.98%	83.98%
biased_datasets/jungle_fpair_bias/jungle	72.17%	84.86%	82.91%
biased_datasets/jungle_doublef_bias/jungle	73.54%	82.62%	82.13%
biased_datasets/calhousing_no_bias/calhousing	78.32%	85.16%	85.74%
biased_datasets/calhousing_label_bias/calhousing	50.29%	50.29%	
biased_datasets/calhousing_feature_bias/calhousing	75.68%	85.84%	87.11%
biased_datasets/calhousing_combined_bias/calhousing	75.78%	85.06%	84.57%
biased_datasets/calhousing_fpair_bias/calhousing	77.73%	85.16%	85.45%
biased_datasets/calhousing_doublef_bias/calhousing	74.90%	84.86%	85.94%
high-bias-datasets/income-feature-bias/income	82.81%	79.39%	78.03%
high-bias-datasets/income-combined-bias/income	36.33%	33.30%	33.89%
high-bias-datasets/income-fpair-bias/income	75.20%	81.84%	83.20%
high-bias-datasets/income-doublef-bias/income			
high-bias-datasets/bank-feature-bias/bank			
high-bias-datasets/bank-combined-bias/bank	88.96%	48.44%	48.34%
high-bias-datasets/bank-fpair-bias/bank	85.84%	86.62%	86.72%
high-bias-datasets/bank-doublef-bias/bank			
high-bias-datasets/jungle-feature-bias/jungle	77.44%	84.57%	84.77%
high-bias-datasets/jungle-combined-bias/jungle	72.56%	74.80%	75.68%
high-bias-datasets/jungle-fpair-bias/jungle	53.42%	85.74%	84.18%
high-bias-datasets/jungle-doublef-bias/jungle	76.37%	82.42%	80.66%
high-bias-datasets/calhousing-feature-bias/calhousing	76.95%	85.45%	88.28%
high-bias-datasets/calhousing-combined-bias/calhousing	73.63%	85.35%	86.43%
high-bias-datasets/calhousing-fpair-bias/calhousing	76.76%	85.84%	86.62%
high-bias-datasets/calhousing-doublef-bias/calhousing	75.88%	86.04%	86.04%
Average no bias	81.54%	79.87%	85.25%
Average label bias	67.43%	67.43%	
Average feature bias (mid)	78.15%	85.40%	85.62%
Average combined bias (mid)	73.34%	82.86%	84.11%
Average fpair bias (mid)	80.98%	85.11%	85.33%
Average doublef bias (mid)	80.15%	84.40%	84.81%
Average feature bias (high)	79.07%	83.14%	83.69%
Average combined bias (high)	67.87%	60.47%	61.08%
Average fpair bias (high)	72.80%	85.01%	85.18%
Average doublef bias (high)	76.12%	84.23%	83.35%

Table 4.2: Model accuracy for each dataset, 1024 samples from large datasets

4.3.2 Observations

The results are in line with the observations of the previous experiment for reasons that have already been explained.

4.4 Experiment Few Shot Training

4.4.1 Description

As this experiment's name suggests, it was aimed at investigating the effects of the number of shots in few-shot training of the models. This was an important factor in the TabLLM paper where the LLM's performance was found better compared to other methods for tabular data in zero-shot and few-shot training of models (the less shots the better) [14]. Thus this experiment should give insight into how bias affects the three kinds of models during their training. The kinds of bias used in the datasets were combined and fpair, both in mid and high intensity, as these are considered to be the most similar to naturally occurring bias as well as other reasons explained before.

In more technical details, the number of shots was exponentially stepped 1, 2, ..., 1024 just like in the TabLLM paper and this time 20 epochs were used with no prompting for the LLM. The LLM used is still mt0. The results may be found in table 4.3 below:

Dataset	Shots	LLM (mt0)	XGBoost	TabPFN
biased_datasets/blood_combined_bias/blood	1	78.67%	78.67%	
	2	44.67%	78.67%	66.67%
	4	78.67%	78.67%	72.00%
	8	78.67%	78.67%	67.33%
	16	78.67%	78.67%	67.33%
	32	78.67%	64.00%	78.67%
	64	78.67%	71.33%	78.67%
	128	78.67%	71.33%	78.67%
	256	78.67%	72.67%	83.33%
	512	78.67%	75.33%	82.67%
	1024	80.67%	78.00%	82.67%
biased_datasets/blood_fpair_bias/blood	1	78.00%	78.00%	
	2	78.00%	78.00%	
	4	78.00%	78.00%	
	8	78.00%	78.00%	78.00%
	16	78.00%	78.00%	78.00%
	32	78.00%	78.00%	78.00%
	64	78.00%	76.00%	78.00%
	128	78.00%	73.33%	79.33%
	256	78.00%	77.33%	79.33%
	512	80.67%	78.67%	79.33%
	1024	79.33%	81.33%	81.33%
biased_datasets/diabetes_combined_bias/diabetes	1	62.99%	62.99%	
	2	62.99%	62.99%	
	4	62.99%	62.99%	64.29%
	8	62.99%	62.99%	65.58%
	16	62.99%	63.64%	70.78%
	32	62.99%	71.43%	71.43%
	64	67.53%	74.03%	70.78%
	128	62.99%	70.13%	74.03%
	256	68.83%	74.03%	76.62%
	512	76.62%	77.27%	76.62%
	1024	73.38%	72.08%	77.27%
biased_datasets/diabetes_fpair_bias/diabetes	1	68.18%	68.18%	
	2	68.18%	68.18%	
	4	68.18%	68.18%	69.48%
	8	70.13%	60.39%	61.69%
	16	64.94%	53.90%	60.39%
	32	65.58%	63.64%	74.68%
	64	68.18%	70.78%	75.32%
	128	70.78%	76.62%	79.22%
	256	77.27%	72.73%	77.27%
	512	74.03%	71.43%	78.57%
	1024	76.62%	74.03%	77.92%
biased_datasets/heart_combined_bias/heart	1	54.35%	54.35%	
	2	63.04%	45.65%	82.07%
	4	85.33%	45.65%	76.09%
	8	87.50%	75.54%	82.07%
	16	84.78%	87.50%	85.87%
	32	86.96%	88.04%	84.24%
	64	87.50%	82.61%	84.24%

	128	87.50%	86.96%	90.76%
	256	90.22%	87.50%	89.67%
	512	90.76%	90.22%	89.67%
	1024	88.04%	88.59%	89.67%
biased_datasets/heart_fpair_bias/heart	1	59.78%	42.93%	
	2	57.07%	42.93%	66.30%
	4	81.52%	42.93%	78.26%
	8	79.89%	79.89%	78.26%
	16	57.07%	73.91%	77.72%
	32	80.98%	77.17%	79.89%
	64	83.70%	77.17%	77.17%
	128	88.59%	82.61%	80.43%
	256	91.30%	85.87%	88.04%
	512	89.13%	84.24%	86.41%
	1024	90.76%	88.59%	89.67%
biased_datasets/car_combined_bias/car	1	70.81%	70.81%	
	2	70.81%	70.81%	
	4	70.81%	70.81%	
	8	70.81%	70.81%	
	16	70.81%	70.81%	70.81%
	32	70.81%	79.48%	77.75%
	64	77.75%	82.37%	83.24%
	128	77.75%	82.37%	87.28%
	256	86.42%	87.28%	92.77%
	512	93.93%	93.35%	96.24%
	1024	97.69%	96.82%	95.66%
biased_datasets/car_fpair_bias/car	1	70.23%	70.23%	
	2	70.23%	70.23%	
	4	70.23%	70.23%	
	8	70.23%	70.23%	
	16	70.23%	70.23%	
	32	70.23%	80.06%	75.72%
	64	77.75%	76.59%	83.24%
	128	78.32%	82.95%	88.15%
	256	86.42%	88.44%	94.22%
	512	90.46%	93.64%	95.38%
	1024	97.11%	96.82%	96.82%
high-bias-datasets/blood-combined-bias/blood	1	80.00%	80.00%	
	2	80.00%	80.00%	34.00%
	4	78.67%	80.00%	68.00%
	8	80.00%	80.00%	74.00%
	16	80.00%	80.00%	80.00%
	32	80.00%	80.00%	80.00%
	64	80.00%	78.67%	80.00%
	128	80.00%	58.00%	68.67%
	256	80.00%	60.67%	73.33%
	512	80.00%	61.33%	72.67%
	1024	80.00%	61.33%	72.67%
high-bias-datasets/blood-fpair-bias/blood	1	77.33%	77.33%	
	2	77.33%	77.33%	
	4	77.33%	77.33%	
	8	77.33%	77.33%	77.33%
	16	77.33%	77.33%	75.33%

	32	77.33%	79.33%	76.00%
	64	77.33%	78.00%	77.33%
	128	77.33%	80.67%	77.33%
	256	79.33%	70.67%	76.00%
	512	77.33%	73.33%	76.67%
	1024	78.67%	74.00%	78.00%
high-bias-datasets/diabetes-combined-bias/diabetes	1	68.18%	68.18%	
	2	68.18%	68.18%	65.58%
	4	68.18%	68.18%	68.18%
	8	68.18%	68.18%	68.18%
	16	68.18%	73.38%	72.73%
	32	68.18%	74.03%	75.32%
	64	68.18%	81.82%	79.22%
	128	72.08%	78.57%	75.97%
	256	68.18%	74.68%	77.92%
	512	69.48%	74.03%	79.87%
	1024	81.82%	75.97%	79.22%
high-bias-datasets/diabetes-fpair-bias/diabetes	1	62.99%	62.99%	
	2	62.34%	62.99%	68.18%
	4	57.79%	62.99%	70.13%
	8	62.99%	62.99%	54.55%
	16	62.99%	67.53%	62.34%
	32	62.99%	66.23%	69.48%
	64	62.99%	67.53%	65.58%
	128	62.99%	72.08%	78.57%
	256	62.99%	75.97%	79.22%
	512	74.68%	72.73%	79.22%
	1024	75.97%	73.38%	77.92%
high-bias-datasets/heart-combined-bias/heart	1	53.80%	53.80%	
	2	76.63%	46.20%	73.91%
	4	46.20%	46.20%	58.70%
	8	46.20%	46.20%	75.54%
	16	48.37%	46.20%	70.65%
	32	46.20%	68.48%	65.76%
	64	46.20%	66.85%	62.50%
	128	55.43%	67.39%	60.33%
	256	54.35%	53.80%	56.52%
	512	56.52%	55.98%	55.43%
	1024	55.43%	55.98%	55.43%
high-bias-datasets/heart-fpair-bias/heart	1	55.98%	55.98%	
	2	55.98%	55.98%	
	4	55.98%	55.98%	55.43%
	8	55.98%	55.98%	64.67%
	16	55.98%	71.74%	58.70%
	32	83.15%	61.41%	73.37%
	64	83.15%	86.41%	75.00%
	128	83.70%	83.70%	82.07%
	256	88.59%	87.50%	86.96%
	512	88.59%	87.50%	86.41%
	1024	88.59%	87.50%	86.41%
high-bias-datasets/car-combined-bias/car	1	70.52%	70.52%	
	2	70.52%	70.52%	
	4	70.52%	70.52%	69.94%

	8	70.52%	70.52%	70.52%
	16	70.52%	67.63%	69.36%
	32	70.52%	61.85%	74.28%
	64	70.52%	74.57%	80.92%
	128	70.52%	81.79%	88.44%
	256	82.08%	78.61%	92.20%
	512	87.57%	88.73%	92.20%
	1024	88.44%	85.26%	91.33%
high-bias-datasets/car-fpair-bias/car	1	69.65%	69.65%	
	2	69.65%	69.65%	
	4	69.65%	69.65%	
	8	69.65%	69.65%	69.65%
	16	69.65%	64.16%	72.83%
	32	69.65%	68.50%	73.70%
	64	69.65%	77.46%	81.50%
	128	73.41%	80.64%	84.97%
	256	82.66%	85.84%	88.44%
	512	90.46%	89.02%	91.04%
	1024	91.04%	97.11%	93.93%
high-bias-datasets/calhousing-combined-bias/calhousing	1	50.68%	50.68%	
	2	50.58%	49.32%	53.29%
	4	49.27%	49.32%	50.05%
	8	50.70%	49.32%	68.24%
	16	49.32%	53.39%	59.42%
	32	52.57%	72.02%	73.72%
	64	53.49%	76.70%	78.75%
	128	59.42%	77.93%	82.75%
	256	73.43%	80.60%	83.65%
	512	74.73%	84.69%	84.40%
	1024	76.91%	84.62%	85.08%
high-bias-datasets/calhousing-fpair-bias/calhousing	1	48.98%	48.98%	
	2	48.98%	48.98%	
	4	48.98%	48.98%	49.98%
	8	48.98%	48.98%	55.21%
	16	48.98%	72.80%	75.19%
	32	51.45%	70.25%	72.65%
	64	54.09%	68.75%	75.36%
	128	62.16%	76.33%	83.48%
	256	63.52%	79.72%	83.28%
	512	75.94%	83.09%	85.15%
	1024	78.08%	85.05%	85.66%
high-bias-datasets/jungle-combined-bias/jungle	1	51.76%	51.76%	
	2	51.76%	51.76%	
	4	51.76%	51.76%	53.51%
	8	51.76%	51.76%	54.05%
	16	51.76%	61.18%	60.69%
	32	51.76%	65.50%	62.49%
	64	60.58%	70.09%	71.47%
	128	61.79%	71.51%	65.56%
	256	65.08%	69.53%	67.90%
	512	67.19%	71.73%	70.34%
	1024	64.85%	73.67%	73.27%
high-bias-datasets/jungle-fpair-bias/jungle	1	48.43%	48.43%	

	2	48.43%	48.43%	
	4	48.43%	48.43%	49.44%
	8	48.43%	48.43%	61.20%
	16	51.94%	68.50%	67.88%
	32	53.39%	67.30%	69.62%
	64	56.51%	73.77%	70.93%
	128	60.04%	77.49%	77.81%
	256	73.28%	79.75%	78.27%
	512	77.22%	81.30%	81.62%
	1024	70.84%	84.05%	83.56%
high-bias-datasets/bank-combined-bias/bank	1	88.85%	88.85%	
	2	88.85%	88.85%	
	4	88.85%	88.85%	88.85%
	8			
	16	88.85%	88.85%	80.49%
	32	89.73%	82.12%	70.57%
	64	89.73%	45.77%	52.39%
	128	89.73%	48.83%	48.89%
	256	47.77%	47.99%	49.45%
	512	47.25%	47.44%	49.39%
	1024	47.08%	47.56%	47.85%
high-bias-datasets/bank-fpair-bias/bank	1	88.63%	88.63%	
	2	88.63%	88.63%	
	4	88.63%	88.63%	88.61%
	8			
	16	88.63%	88.63%	88.63%
	32	88.63%	88.63%	88.60%
	64	88.63%	88.63%	88.60%
	128	88.88%	87.79%	89.19%
	256	88.63%	89.01%	88.99%
	512	88.39%	89.24%	89.16%
	1024	88.85%	89.12%	89.01%
Average Combined Bias (mid)	1	66.70%	66.70%	
	2	60.38%	64.53%	74.37%
	4	74.45%	64.53%	70.79%
	8	74.99%	72.00%	71.66%
	16	74.31%	75.15%	73.70%
	32	74.85%	75.74%	78.02%
	64	77.86%	77.58%	79.23%
	128	76.72%	77.70%	82.68%
	256	81.03%	80.37%	85.60%
	512	85.00%	84.04%	86.30%
	1024	84.94%	83.87%	86.32%
Average fpair Bias (mid)	1	69.05%	64.84%	
	2	68.37%	64.84%	66.30%
	4	74.48%	64.84%	73.87%
	8	74.56%	72.13%	72.65%
	16	67.56%	69.01%	72.04%
	32	73.70%	74.72%	77.07%
	64	76.91%	75.14%	78.43%
	128	78.92%	78.88%	81.78%
	256	83.25%	81.09%	84.72%
	512	83.57%	81.99%	84.92%

	1024	85.96%	85.19%	86.44%
Average Combined Bias (high)	1	68.13%	68.13%	
	2	73.83%	66.22%	57.83%
	4	65.89%	66.22%	66.20%
	8	66.22%	66.22%	72.06%
	16	66.77%	66.80%	73.19%
	32	66.22%	71.09%	73.84%
	64	66.22%	75.47%	75.66%
	128	69.51%	71.44%	73.35%
	256	71.15%	66.94%	74.99%
	512	73.39%	70.02%	75.04%
	1024	76.42%	69.64%	74.66%
Average fpair Bias (high)	1	66.49%	66.49%	
	2	66.33%	66.49%	68.18%
	4	65.19%	66.49%	62.78%
	8	66.49%	66.49%	66.55%
	16	66.49%	70.19%	67.30%
	32	73.28%	68.87%	73.14%
	64	73.28%	77.35%	74.86%
	128	74.36%	79.27%	80.74%
	256	78.39%	79.99%	82.65%
	512	82.76%	80.64%	83.34%
	1024	83.57%	83.00%	84.07%

Table 4.3: Model accuracy for each dataset and number of shots, few-shot training

4.4.2 Observations

To help present the observations of this experiment a graph has been made for each kind of bias and intensity. They present each models average accuracy across all datasets with the specific kind of bias and specific intensity.

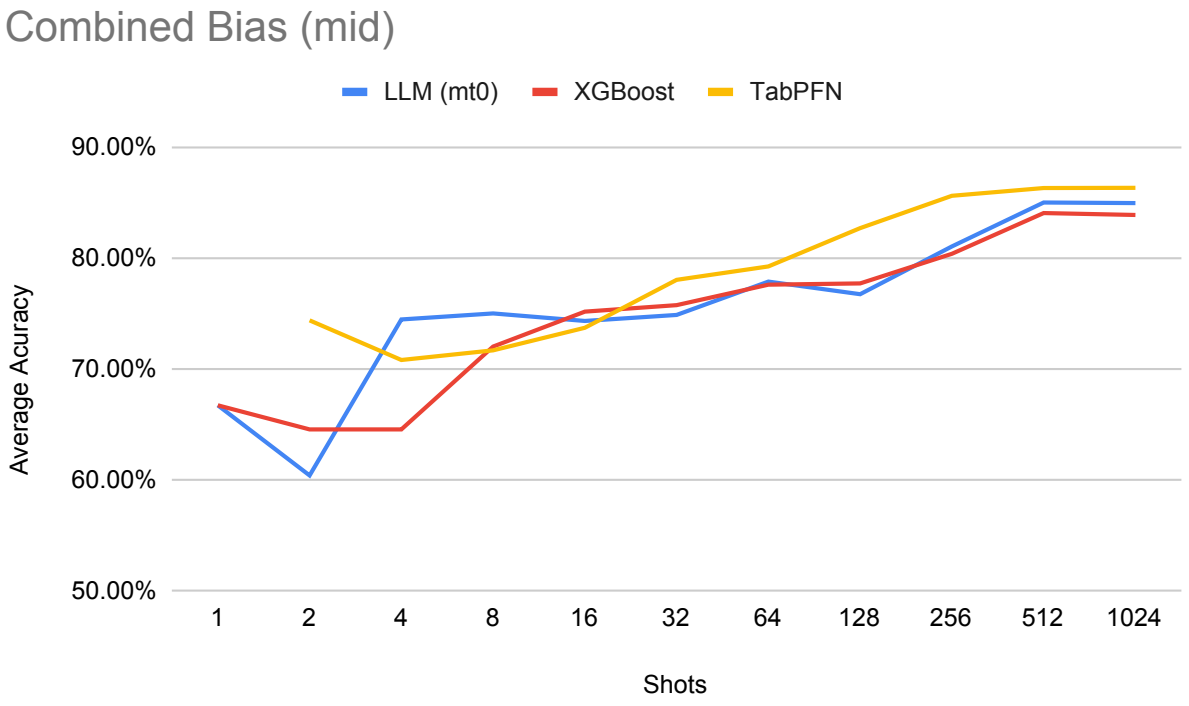

Figure 4.4.1: Few-shot learning accuracy, combined bias (mid)

Starting with combined bias in mid intensity, as seen in figure 4.4.1, it is evident that the large language model on average outperforms the other two methods in really few-shot situations but then closely follows XGBoost's performance ending with a slight advantage to it.

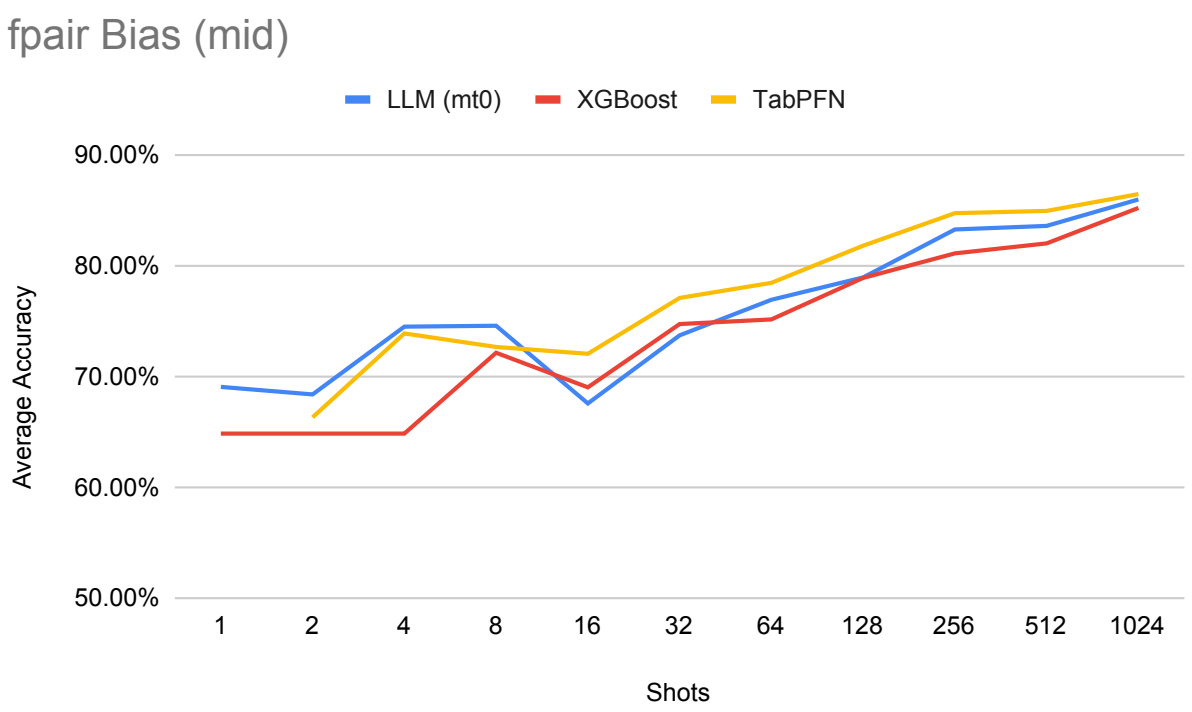

Figure 4.4.2: Few-shot learning accuracy, fpair bias (mid)

In mid fpair bias, as seen in figure 4.4.2, the large language model performs even better for really few-shots, this time outperforming the other two models in all cases were the number of shots is less than 8. Additionally, now it performs slightly better than XGBoost for larger numbers of shots, though it is still bettered by TabPFN.

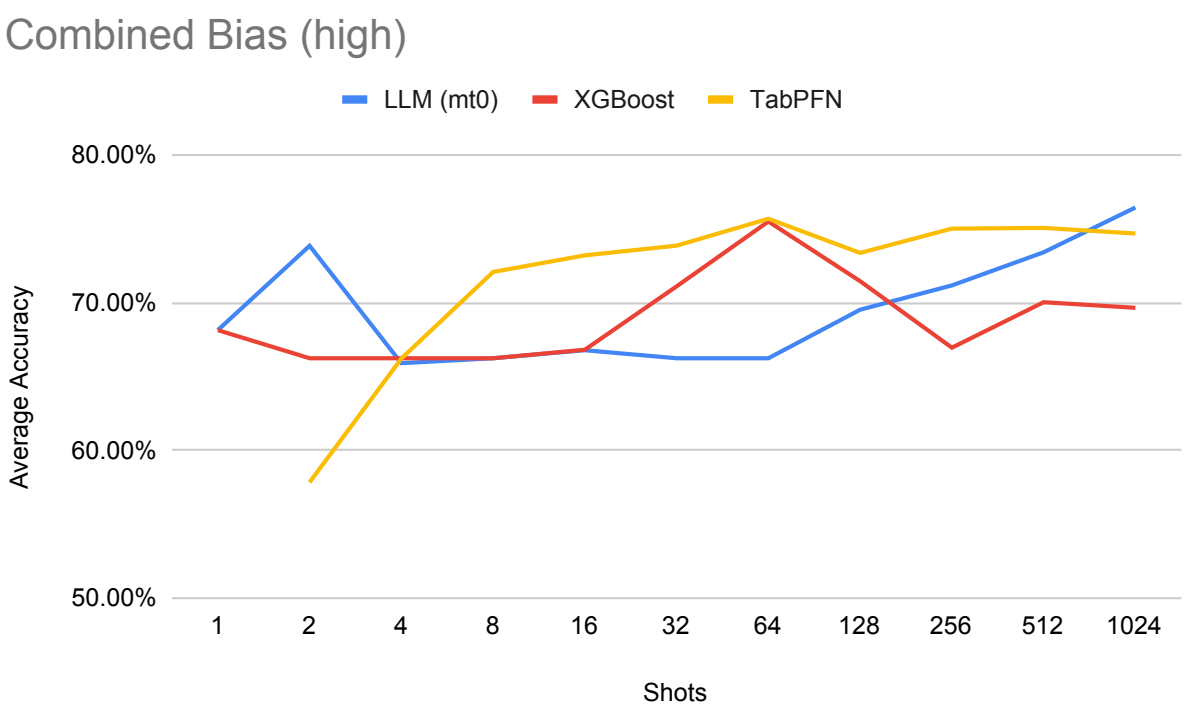

Figure 4.4.3: Few-shot learning accuracy, combined bias (high)

In the high bias version of combined bias, as seen in figure 4.4.3, the large language model still performs better for very few shots. However, this time XGBoost and TabPFN stop showing improvement after a number of shots (they peak at 64) while at the same time the large language model starts showing a steady increase in its accuracy, eventually surpassing the other two methods. This is probably due to the fact that the other two methods in this case were thrown off because of the high bias and it hints at the large language model being able to overcome data bias. This could be the case because combined bias affects the labels directly and so XGBoost and TabFPN could not cope as well as a large language model which can also use its prior knowledge.

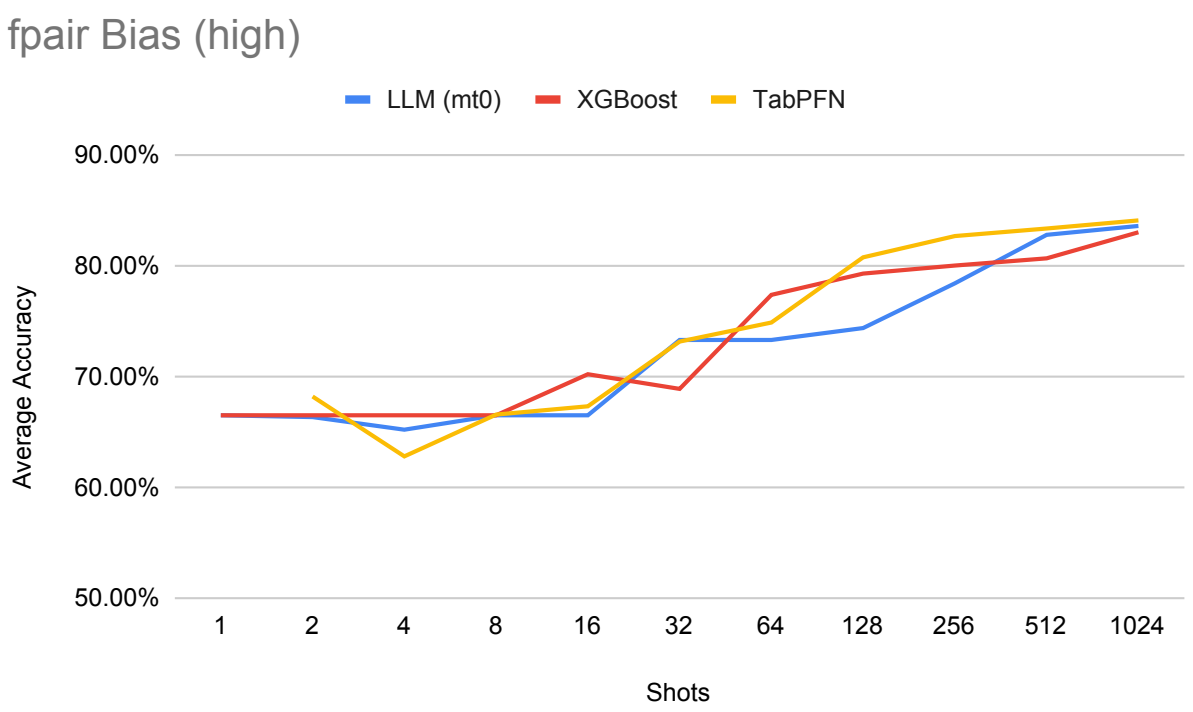

Figure 4.4.4: Few-shot learning accuracy, fpair bias (high)

In high fpair bias, as also seen in figure 4.4.4, all tested models seem to be doing equally well with at times either one being the best. It is possible that differences between models are not as visible here because, in contrast to combined bias, fpair bias does not directly affect labels.

4.5 Experiment Sequence to Sequence

4.5.1 Description

Up until now the large language model used a small trainable classification head to convert tokens into classes as described in section 3.2.6. The purpose of this experiment then, was to investigate the alternative method described there. A small detail is that now the model is also evaluated at zero-shot predictions. Previously, zero-shot did not make much sense since the classification head was not pretrained, but now that it doesn't exist the model can run zero-shot. It is no different from the previous one in any other way and all other experimental parameters are equal. As such, the reader will notice that on the table following shortly, the performance scores for XGBoost and TabPFN have not been filled. That is because they are the same as before. Their average performance was filled in to render comparisons easier. The results have been recorded in table 4.4 below:

Dataset	Shots	LLM (mt0)	XGBoost	TabPFN
biased_datasets/blood_combined_bias/blood	0	0.00%		
	1	78.67%		
	2	78.67%		
	4	78.67%		
	8	78.67%	same	same
	16	78.67%	with	with
	32	78.67%	few-shot	few-shot
	64	78.67%	training	training
	128	78.67%		
	256	78.67%		
	512	78.67%		
	1024	78.67%		
biased_datasets/blood_fpair_bias/blood	0	0.00%		
	1	78.00%		
	2	78.00%		
	4	78.00%		
	8	78.00%		
	16	78.00%		
	32	78.00%		
	64	78.00%		
	128	78.00%		
	256	78.00%		
	512	79.33%		
biased_datasets/diabetes_combined_bias/diabetes	1024	79.33%		
	0	12.34%		
	1	67.53%		
	2	63.64%		
	4	68.18%		
	8	64.94%		
	16	64.29%		
	32	67.53%		
	64	68.83%		
	128	66.88%		
	256	70.13%		
	512	75.32%		
	1024	75.97%		
biased_datasets/diabetes_fpair_bias/diabetes	0	7.79%		
	1	63.64%		
	2	68.18%		
	4	62.34%		
	8	53.90%		
	16	55.19%		
	32	63.64%		
	64	70.13%		
	128	72.08%		
	256	77.27%		
	512	77.27%		
	1024	81.17%		
biased_datasets/heart_combined_bias/heart	0	0.00%		
	1	54.35%		
	2	53.80%		

	4	61.96%
	8	71.20%
	16	75.54%
	32	80.43%
	64	84.78%
	128	89.67%
	256	89.67%
	512	90.76%
	1024	91.30%
biased_datasets/heart_fpair_bias/heart	0	0.00%
	1	36.41%
	2	48.91%
	4	52.17%
	8	53.26%
	16	62.50%
	32	81.52%
	64	82.07%
	128	85.33%
	256	89.67%
	512	90.76%
	1024	89.67%
biased_datasets/car_combined_bias/car	0	1.16%
	1	70.81%
	2	70.81%
	4	70.81%
	8	70.81%
	16	70.81%
	32	70.81%
	64	72.25%
	128	79.19%
	256	86.13%
	512	94.80%
	1024	98.27%
biased_datasets/car_fpair_bias/car	0	2.31%
	1	70.23%
	2	70.23%
	4	70.23%
	8	70.23%
	16	70.23%
	32	70.23%
	64	71.97%
	128	81.21%
	256	84.97%
	512	91.04%
	1024	97.69%
high-bias-datasets/blood-combined-bias/blood	0	0.00%
	1	80.00%
	2	80.00%
	4	80.00%
	8	80.00%
	16	80.00%
	32	80.00%
	64	80.00%
	128	80.00%

	256	80.00%
	512	80.00%
	1024	80.00%
high-bias-datasets/blood-fpair-bias/blood	0	0.00%
	1	77.33%
	2	77.33%
	4	77.33%
	8	77.33%
	16	77.33%
	32	77.33%
	64	77.33%
	128	77.33%
	256	77.33%
	512	78.00%
	1024	77.33%
high-bias-datasets/diabetes-combined-bias/diabetes	0	11.04%
	1	25.97%
	2	32.47%
	4	68.83%
	8	68.18%
	16	69.48%
	32	68.83%
	64	68.83%
	128	70.13%
	256	71.43%
	512	80.52%
	1024	81.17%
high-bias-datasets/diabetes-fpair-bias/diabetes	0	13.64%
	1	41.56%
	2	41.56%
	4	39.61%
	8	43.51%
	16	59.09%
	32	62.99%
	64	63.64%
	128	62.34%
	256	70.78%
	512	75.32%
	1024	74.68%
high-bias-datasets/heart-combined-bias/heart	0	0.00%
	1	51.09%
	2	55.98%
	4	52.17%
	8	51.09%
	16	53.26%
	32	52.17%
	64	67.39%
	128	78.26%
	256	79.35%
	512	70.11%
	1024	62.50%
high-bias-datasets/heart-fpair-bias/heart	0	0.00%
	1	55.98%

	2	55.98%		
	4	55.98%		
	8	72.83%		
	16	67.39%		
	32	83.15%		
	64	86.96%		
	128	87.50%		
	256	88.59%		
	512	88.59%		
	1024	88.04%		
high-bias-datasets/car-combined-bias/car	0	2.31%		
	1	70.52%		
	2	70.52%		
	4	70.52%		
	8	70.52%		
	16	70.52%		
	32	70.52%		
	64	70.52%		
	128	66.76%		
	256	76.59%		
	512	89.60%		
	1024	91.91%		
high-bias-datasets/car-fpair-bias/car	0	2.89%		
	1	69.65%		
	2	69.65%		
	4	69.65%		
	8	69.65%		
	16	69.65%		
	32	69.65%		
	64	70.52%		
	128	76.30%		
	256	82.95%		
	512	88.15%		
	1024	94.80%		
Average Combined Bias (mid)	0	3.37%		
	1	67.84%	66.70%	
	2	66.73%	64.53%	74.37%
	4	69.90%	64.53%	70.79%
	8	71.40%	72.00%	71.66%
	16	72.33%	75.15%	73.70%
	32	74.36%	75.74%	78.02%
	64	76.13%	77.58%	79.23%
	128	78.60%	77.70%	82.68%
	256	81.15%	80.37%	85.60%
	512	84.89%	84.04%	86.30%
	1024	86.05%	83.87%	86.32%
Average fpair Bias (mid)	0	2.53%		
	1	62.07%	64.84%	
	2	66.33%	64.84%	66.30%
	4	65.69%	64.84%	73.87%
	8	63.85%	72.13%	72.65%
	16	66.48%	69.01%	72.04%
	32	73.35%	74.72%	77.07%
	64	75.54%	75.14%	78.43%

	128	79.15%	78.88%	81.78%
	256	82.48%	81.09%	84.72%
	512	84.60%	81.99%	84.92%
	1024	86.97%	85.19%	86.44%
Average Combined Bias (high)	0	3.34%		
	1	56.90%	68.13%	
	2	59.74%	66.22%	57.83%
	4	67.88%	66.22%	66.20%
	8	67.45%	66.22%	72.06%
	16	68.32%	66.80%	73.19%
	32	67.88%	71.09%	73.84%
	64	71.69%	75.47%	75.66%
	128	73.79%	71.44%	73.35%
	256	76.84%	66.94%	74.99%
	512	80.06%	70.02%	75.04%
	1024	78.89%	69.64%	74.66%
Average fpair Bias (high)	0	4.13%		
	1	61.13%	66.49%	
	2	61.13%	66.49%	68.18%
	4	60.64%	66.49%	62.78%
	8	65.83%	66.49%	66.55%
	16	68.37%	70.19%	67.30%
	32	73.28%	68.87%	73.14%
	64	74.61%	77.35%	74.86%
	128	75.87%	79.27%	80.74%
	256	79.91%	79.99%	82.65%
	512	82.52%	80.64%	83.34%
	1024	83.71%	83.00%	84.07%

Table 4.4: Model accuracy for each dataset and number of shots, using sequence-to-sequence

4.5.2 Observations

Similarly to the previous experiment, to better illustrate the observations, a few graphs showing the models' average performance over the number of shots have been created. As one can see in all graphs, unfortunately zero-shot performance is incredibly low for this large language model. As will be show in the next experiment, this is because the mt-0 model used only has 570M parameters and the larger models will demonstrate an acceptable zero-shot performance. But this is for later

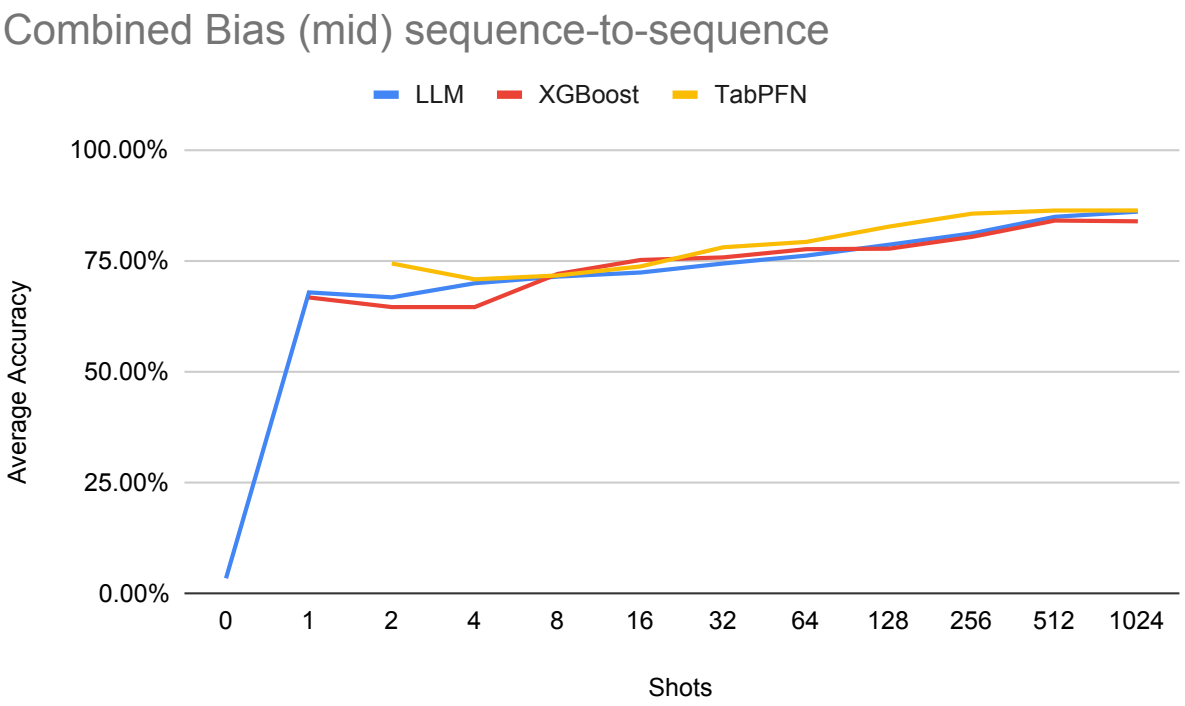

Figure 4.5.1: Few-shot learning accuracy, combined bias (mid) sequence-to-sequence

In the case of combined bias of mid intensity the large language model performs similar as before. A minor difference is that it performs slightly worse in really few shots like two and less but it performs slightly better with more shots, even reaching TabPFN at 1024.

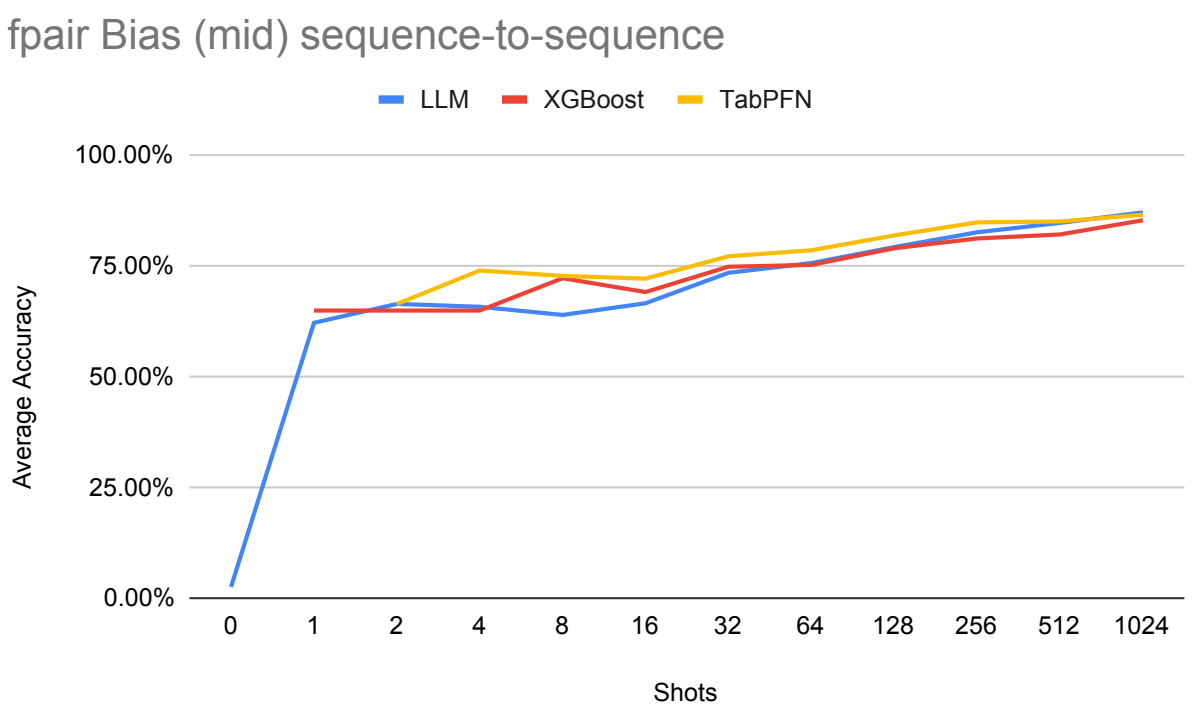

Figure 4.5.2: Few-shot learning accuracy, fpair bias (mid) sequence-to-sequence

In the case of fpair bias of mid intensity, the large language model initially doesn't perform that well on average in very few shots but as the number of shots increases it prfrorms better in comparison to the other two methods and eventually surpasses both XGBoost and TabPFN.

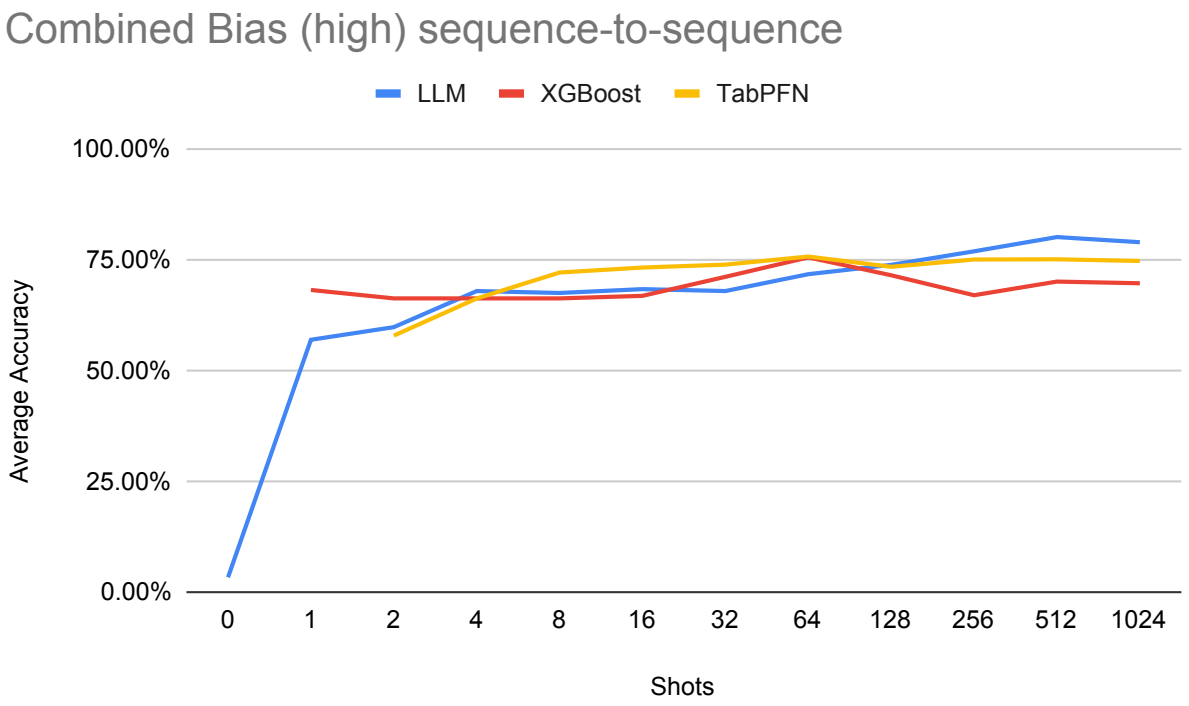

Figure 4.5.3: Few-shot learning accuracy, combined bias (high) sequence-to-sequence

In the case of high combined bias, the large language model exhibits the best performance out of all the experiments on average compared to XGBost and TabPFN. It performs better than them both at four shots and for each sample after 128 shots, overcoming data bias.

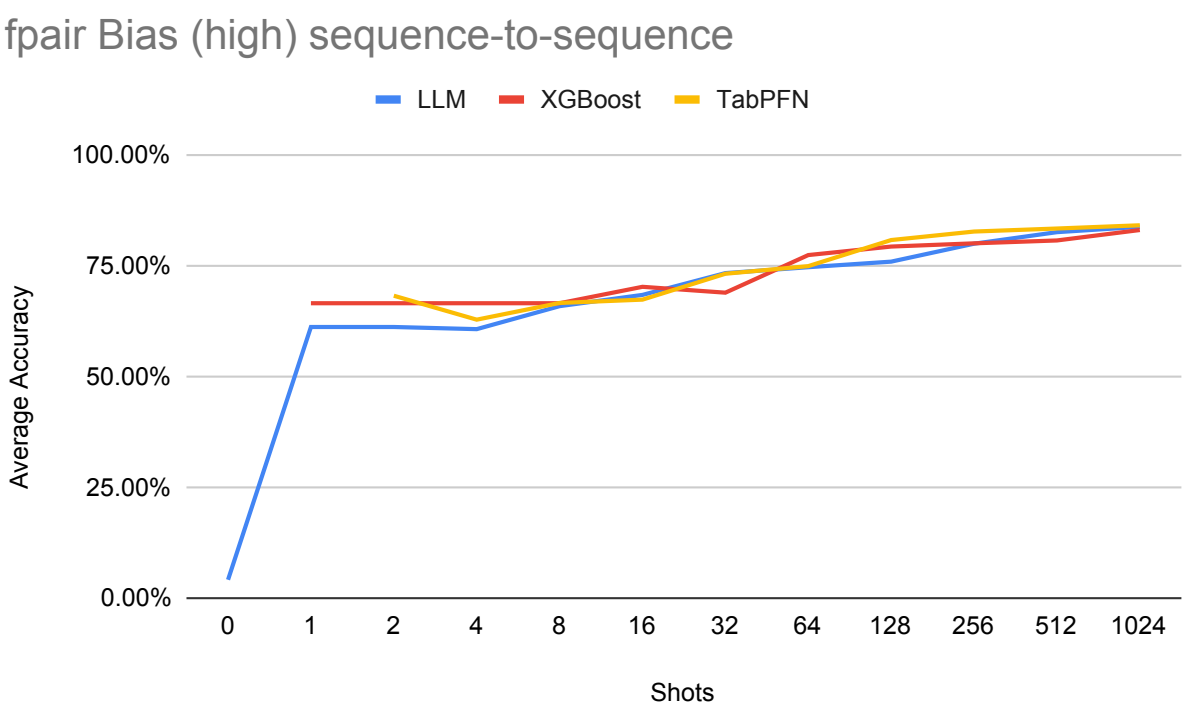

Figure 4.5.4: Few-shot learning accuracy, fpair bias (high) sequence-to-sequence

Finally, in the case of high fapir bias, all models perform quite simlarly.

Overall

Overall, the large language model performs better compared to XGBoost and TabPFN in combined bias and in the high intensity variants. This is also true for the previous experiment. This shows that large language models can overcome these biases in datasets. Additionally, it seems to be doing slightly better when not using a classification head but, instead, the other method described in subsection 3.2.6, especially with larger numbers of few shots.

4.6 Experiment Few Shot Prompting

4.6.1 Description

In this experiment, three different large language models were tested under bias, using only prompting (and no training). These models are mt0-base (570 million parameters), T0_3B (3 billion parameters) and T0pp (11 billion parameters). More on these models was written in a previous chapter. Again, different numbers of shots were tested to see the effects of bias as well as to compare with the results from training. Necessarily these models did not use a classification head which would need to be trained but the other method described in subsection 3.2.6. The results are presented in table 4.5 below:

Dataset	Shots	mt0	T0_3B	T0pp
biased_datasets/blood_combined_bias/blood	0	0.00%	78.67%	21.33%
	1	78.67%	78.67%	78.67%
	2	21.33%	78.67%	21.33%
	4	21.33%	78.67%	78.67%
	8	21.33%	78.67%	
	16	78.67%	21.33%	
	32	78.67%	21.33%	
biased_datasets/blood_fpair_bias/blood	0	0.00%	78.00%	22.00%
	1	78.00%	22.00%	22.00%
	2	78.00%	35.33%	78.00%
	4	22.00%	78.00%	22.00%
	8	78.00%	72.67%	
	16	22.00%	22.00%	
	32	78.00%	78.00%	
biased_datasets/diabetes_combined_bias/diabetes	0	12.34%	37.01%	62.99%
	1	62.34%	37.01%	37.01%
	2	37.01%	37.01%	62.99%
	4	62.34%	62.99%	
	8	37.66%	37.01%	
	16	37.01%	62.99%	
	32			
biased_datasets/diabetes_fpair_bias/diabetes	0	7.79%	31.82%	31.82%
	1		31.82%	31.82%
	2		31.82%	68.18%
	4		31.82%	
	8		68.18%	
	16		31.82%	
	32			
biased_datasets/heart_combined_bias/heart	0	0.00%	54.35%	54.35%
	1		54.35%	45.65%
	2		54.35%	
	4		45.65%	
	8		54.35%	
	16		54.35%	
	32			
biased_datasets/heart_fpair_bias/heart	0	0.00%	42.93%	57.07%
	1		42.93%	42.93%
	2		42.93%	
	4		42.93%	
	8		57.07%	
	16		42.93%	
	32			
biased_datasets/creditg_combined_bias/creditg	0		33.00%	67.00%
	1		67.00%	
	2		67.00%	
	4		32.50%	
	8		67.00%	
	16			
	32			
biased_datasets/creditg_fpair_bias/creditg	0		30.00%	31.50%

	1		31.50%	
	2		31.50%	
	4		68.50%	
	8		68.50%	
	16			
	32			
biased_datasets/car_combined_bias/car	0	1.16%	3.47%	3.76%
	1		21.97%	70.81%
	2		3.76%	3.76%
	4		3.76%	
	8		3.47%	
	16		3.76%	
	32		14.45%	
biased_datasets/car_fpair_bias/car	0		4.05%	4.62%
	1		34.39%	70.23%
	2		70.23%	21.10%
	4		4.05%	
	8		4.05%	
	16		8.09%	
	32		4.05%	
biased_datasets/calhousing_combined_bias/calhousing	0	2.31%	49.08%	
	1		50.92%	
	2		49.08%	
	4		49.08%	
	8			
	16			
	32			
high-bias-datasets/blood-combined-bias/blood	0	0.00%	20.00%	80.00%
	1		20.00%	80.00%
	2	20.00%	80.00%	20.00%
	4	20.00%	20.00%	80.00%
	8	20.00%	20.00%	
	16	72.67%	20.00%	
	32	80.00%	20.00%	
high-bias-datasets/blood-fpair-bias/blood	0	0.00%	77.33%	22.67%
	1	22.67%	77.33%	22.67%
	2	77.33%	22.67%	77.33%
	4	77.33%	22.67%	77.33%
	8	22.67%	22.67%	
	16	22.67%	40.67%	
	32	22.67%	22.67%	
high-bias-datasets/diabetes-combined-bias/diabetes	0	11.04%	68.18%	31.82%
	1	66.88%	31.82%	31.82%
	2	68.18%	31.82%	31.82%
	4	31.82%	31.82%	
	8	31.82%	68.18%	
	16	68.18%	68.18%	
	32			
high-bias-datasets/diabetes-fpair-bias/diabetes	0	13.64%	62.99%	37.01%
	1	49.35%	62.99%	37.01%
	2	37.01%	62.99%	62.99%
	4	62.99%	62.99%	

	8	37.01%	62.99%	
	16	37.01%	37.01%	
	32			
high-bias-datasets/heart-combined-bias/heart	0	0.00%	46.20%	58.15%
	1	47.28%	53.80%	46.20%
	2	46.20%	46.20%	
	4	53.80%	46.20%	
	8	46.20%	46.20%	
	16	46.20%	53.80%	
	32			
high-bias-datasets/heart-fpair-bias/heart	0	0.00%	55.98%	44.02%
	1	55.98%	48.91%	44.02%
	2	55.98%	55.98%	
	4	55.98%	44.02%	
	8	55.98%	55.98%	
	16	55.98%	55.98%	
	32			
high-bias-datasets/creditg-combined-bias/creditg	0		29.50%	70.50%
	1	29.50%	29.50%	
	2	29.50%	70.50%	
	4	58.00%	70.50%	
	8	29.50%	70.50%	
	16			
	32			
high-bias-datasets/creditg-fpair-bias/creditg	0		68.00%	32.00%
	1	68.00%	68.00%	
	2	68.00%	68.00%	
	4	68.00%	32.00%	
	8	68.00%	68.00%	
	16			
	32			
high-bias-datasets/car-combined-bias/car	0	2.31%	20.81%	70.52%
	1	4.62%	20.81%	4.62%
	2	4.62%	4.05%	20.81%
	4	19.65%	20.81%	
	8	4.05%	20.81%	
	16	4.05%	70.52%	
	32	43.93%	70.52%	
high-bias-datasets/car-fpair-bias/car	0	2.89%	5.20%	21.39%
	1	5.20%	5.20%	5.20%
	2	25.72%	3.76%	21.39%
	4	31.79%	11.56%	
	8	5.20%	3.76%	
	16	5.20%	69.65%	
	32	21.39%	3.76%	
high-bias-datasets/calhousing-combined-bias/calhousing	0		49.32%	
	1	49.32%	50.68%	
	2	49.32%	49.32%	
	4	50.68%	50.68%	
	8	50.63%		
	16			
	32			

Average Combined Bias (mid)	0	3.16%	42.60%	41.89%
	1		51.65%	58.04%
	2		48.31%	29.36%
	4		45.44%	
	8		48.10%	
	16		35.61%	
	32			
Average fpair Bias (mid)	0	2.60%	37.36%	29.40%
	1		32.53%	41.75%
	2		42.36%	55.76%
	4		45.06%	
	8		54.09%	
	16		26.21%	
	32			
Average Combined Bias (high)	0	3.34%	39.00%	62.20%
	1	39.52%	34.44%	40.66%
	2	36.30%	46.98%	24.21%
	4	38.99%	40.00%	
	8	30.36%	45.14%	
	16	47.77%	53.13%	
	32			
Average fpair Bias (high)	0	4.13%	53.90%	31.42%
	1	40.24%	52.49%	27.23%
	2	52.81%	42.68%	53.90%
	4	59.22%	34.65%	
	8	37.77%	42.68%	
	16	30.22%	50.83%	
	32			

Table 4.5: Model accuracy for each dataset and number of shots, few-shot prompting

4.6.2 Observations

The first observation is that the smallest model, mt0 of 570 million parameters, has nearly 0% accuracy in zero-shot mode, while the other two larger models perform much better. However, with just one shot, the performance of mt0 reaches the levels of the other two larger models.

Secondly, on average, the accuracy of each model seems to increase with the number of shots as expected but it also some times falls, particularly in combined bias for the larger models. This is undoubtedly the effect of bias.

Lastly all performances are severely lower compared to the previous performances we had obtained when using training instead of prompting under the same conditions. Even the larger models under promoting do not perform as well as mt0 when trained.

4.7 Additional Observations: LLM Eventually Learning Data Bias

As expected the large language model is not actually immune to data bias. As such two graphs were produced of the models test set performance per epoch under high intensity bias that show the model eventually learning the bias (and lowering its performance). Not all of the datasets had the same performance curve, these are just two examples were it was particularly noticeable. In both, the large language model reaches a peak performance and its accuracy drops after that having learned the data bias.

Figure 4.7.1: LLM eventually learning data bias 1

Based on these two performance curves alone, one might think that less than 10 epochs are required to get the best the large language model can give under these conditions of bias. However, as stated before these two are not necessarily representative of every case and there were curves where the model kept improving later on.

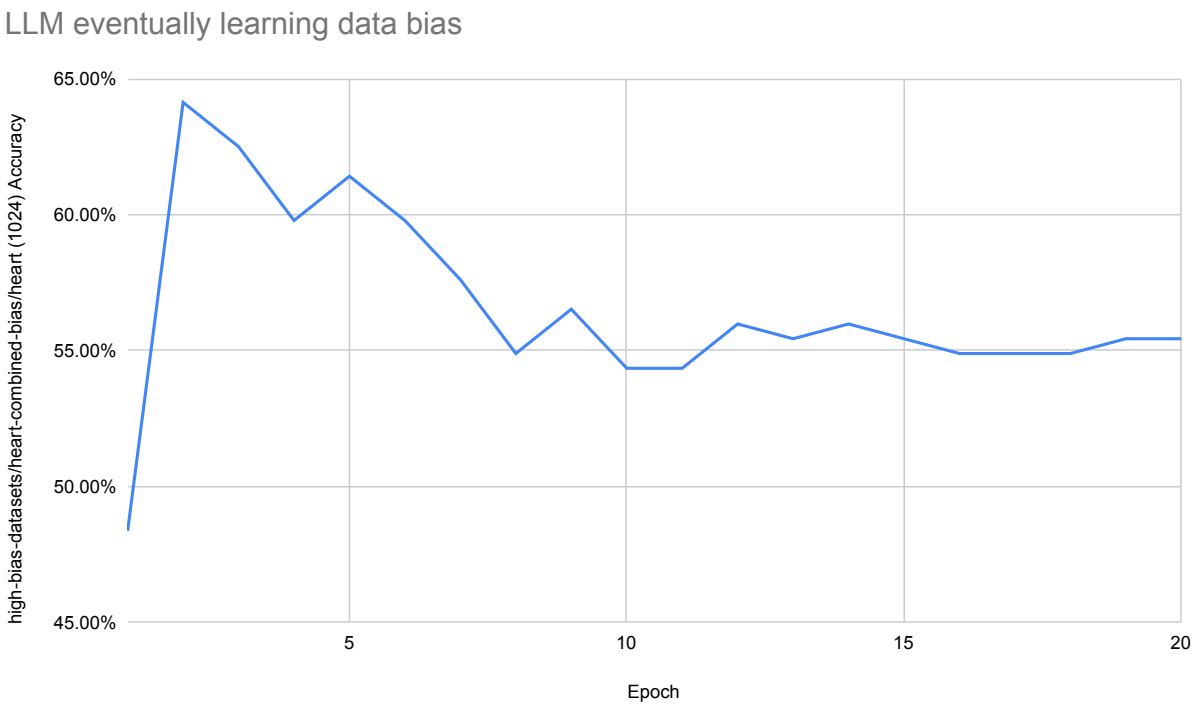

Figure 4.7.2: LLM eventually learning data bias 2

Chapter 5

Conclusion

Contents

5.1 Interpretations on the results

The results of the large language model's performance indicate that LLMs can indeed overcome bias when it is present in the dataset, for several reasons. First of all, a smaller model could perform about the same or even at times outperform state-of-the-art methods in various bias settings. Additionally, as observed, increasing the bias intensity, the model performs comparatively better to the other two state-of-the-art methods. Though the LLMs performance also drops the state-of-the-art methods lose more accuracy with the increase of bias. This shows a resistivity to bias on the part of the LLM. Lasstly, the larger the LLMs were the better performance they demonstrated against bias, likely due to to having more prior knowledge and better language modeling.

The reasons for the above are probably the ones initially conjectured. Firstly, prior knowledge which might help the LLM learn in a way that rejects bias or work around the bias. But also, the ability to utilise the semantics of a feature or label via either embedding or language modeling may play a part, especially given that for instance, XGBoost, being a decision tree model, cannot have access to such semantics and it under performs.

Another observation was that the large language model performed better in label bias, combined bias and feature pair bias. The reason for that is likely that these kinds of bias affect more the way decision trees and similar models learn the dataset than the way an LLM does. Indeed, all of them were expected to impact decision trees as they either drop a label directly (which the decision tree has no way of knowing if it doesn't appear in train set), or they destroy a correlation between a feature and a label (which is something decision trees but many other methods rely on) or on the correlation between features. This might also indicate that large language models do not rely on learning the correlations between a feature and a label but instead they should learn the correlation between the whole sequence of a features and the label as it normally processes sentences. If that is true then compared to LLMs other models might act more like N-grams. In fact this might also explain why TabPFN has such a good overall performance, many times resisting bias itself, it uses a transformer.

On the contrary, large language models were for some reason very negatively affected by double feature bias (i.e. applying feature bias twice) while the other methods weren't affected as much. The reason behind this isn't clear. Since as already explained this kind of bias drops the most samples, it is possible that the textual representations on the train set become too similar and so the large language model may have trouble telling them apart when their difference is only few tokens in specific positions, especially because it was a small LLM of only 570M parameters. As such it is possible the model learned to respond to the structure of the input with the most common label or some similar behavior.

A last observation was that training had better results than prompting and both training and prompting had the best results. This is likely not unique to bias conditions and seems reasonable. Training makes the large language model more fit for a task, while prompting helps it predict more correctly. Naturally, both together are better than one and changing the LLM's weights for a task is stronger than simply prompting it.

5.2 Recap

In this thesis, a tabular data large language model framework was created from scratch based on the recently proposed TabLLM framework. Additionally methods to artificially add bias to data were though of and implemented along with a mechanism to adjust the amount of bias added. The large language model framework was then tested on datasets which were augmented with various kinds of bias at mid and high intensity. Results indicate that large language models can overcome data bias better than classical methods under various circumstances, that larger language models can perform better showing greater resistivity to bias, and that large language models are not immune to data bias, which they also eventually learn.

5.3 Future Work

Future work could include the experimental testing with even larger language models than what was possible with the limited resources available to this project. Additionally, different kinds of bias could be explored,

especially ones derived from actual biased datasets. A more mathematically rigorous theory of bias could be devised by utilising ideas, results and operations from set theory. Finally, an investigation into why some kinds of bias affect LLMs less while others significantly more (like in the case of double feature bias) would be really valuable, especially for explainable AI.

Chapter 6

Bibliography

[1] AutoML.org. "TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second". In: (2022). URL:

[2] Brown, T. B. et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[3] Chen, T. and Guestrin, C. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. ACM, Aug. 2016. DOI: 10.1145/2939672.2939785. URL:

[4] Code, P. W. "TabPFN Explained". In: (2022). URL:

[5] Dervakos, E. et al. "Semantic Enrichment of Pretrained Embedding Output for Unsupervised IR." In: *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*. Vol. 2846. 2021.

[6] Dervakos, E. et al. "Choose your Data Wisely: A Framework for Semantic Counterfactuals". In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. Ed. by E. Elkind. Main Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023, pp. 382–390. DOI: 10.24963/ijcai.2023/43. URL:

[7] Dervakos, E. et al. "Choose your data wisely: A framework for semantic counterfactuals". In: *arXiv preprint arXiv:2305.17667* (2023).

[8] Dimitriou, A. et al. "Structure Your Data: Towards Semantic Graph Counterfactuals". In: *arXiv preprint arXiv:2403.06514* (2024).

[9] Fang, X. et al. *Large Language Models(LLMs) on Tabular Data: Prediction, Generation, and Understanding – A Survey*. 2024. arXiv: 2402.17944 [cs.CL].

[10] Filandrianos, G. et al. "Conceptual Edits as Counterfactual Explanations." In: *AAAI Spring Symposium: MAKE*. 2022.

[11] Filandrianos, G. et al. "Counterfactuals of Counterfactuals: a back-translation-inspired approach to analyse counterfactual editors". In: *arXiv preprint arXiv:2305.17055* (2023).

[12] Giadikiaroglou, P. et al. "Puzzle Solving using Reasoning of Large Language Models: A Survey". In: *arXiv preprint arXiv:2402.11291* (2024).

[13] Griogoriadou, N. et al. "AILS-NTUA at SemEval-2024 Task 6: Efficient model tuning for hallucination detection and analysis". In: *arXiv preprint arXiv:2404.01210* (2024).

[14] Hegselmann, S. et al. "Tabllm: Few-shot classification of tabular data with large language models". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 5549–5581.

[15] Hollmann, N. et al. "TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second". In: *The Eleventh International Conference on Learning Representations*. 2023. URL:

[16] Muennighoff, N. et al. "Crosslingual generalization through multitask finetuning". In: *arXiv preprint arXiv:2211.01786* (2022).

[17] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL].

[18] Panagiotopoulos, I. et al. "AILS-NTUA at SemEval-2024 Task 9: Cracking Brain Teasers: Transformer Models for Lateral Thinking Puzzles". In: *arXiv preprint arXiv:2404.01084* (2024).

[19] Sanh, V. et al. *Multitask Prompted Training Enables Zero-Shot Task Generalization*. 2021. arXiv: 2110.08207 [cs.LG].

[20] Vaswani, A. et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

[21] VLDB. "Large Language Models(LLMs) on Tabular Data: Prediction, Generation, and Understanding - A Survey". In: *VLDB* (2023). URL:

[22] Wolf, T. et al. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2020. arXiv: 1910.03771 [cs.CL].