Schätzen von Wertebereichen mit ChatGPT

Petros Tsialis
Hochschule Aalen
Fakultät Elektornik und Informatik
Aalen, Deutschland
petros.tsialis@studmail.htw-aalen.de
Betreuer: Prof. Dr. Martin Heckmann

Abstract*—Ist es möglich mithilfe von ChatGPT Wertebereiche eines Features zu bestimmen, so das unerfahrene *Data Science Nutzer dies benutzen können, als eine Art Stütze in ihrer Arbeit? Diese Arbeit beschäftigt sich mit der Thematik des schätzen von Wertebereichen mit Hilfe des LLMS ChatGPT-3.5-turbo. Um herauszufinden, wie gut und ob das Funktioniert wurde, ein Experiment erstellt, das die oben Gestelle Frage beantworten soll.

Keywords—LLM, ChatGPT, Data pre-processing, Value ranges

I. Motivation

Generative Pre-trained Transformer (ChatGPT), ist ein Chatbot, der Methoden aus der Künstlichen Intelligenz benutzt, um mit Nutzern zu interagieren. Er benutzt modernste Ansätze aus dem Machine Learning- und Natural language processing Bereich, um Nutzern passende Antworten auf jede Frage geben zu können [1]. Dabei versucht ChatGPT so natürlich, wie möglich zu klingen, um dem Nutzer ein angenehmes Erlebnis zu ermöglichen. ChatGPT kann Aufgaben erledigen, wie:

- Texte schreiben
- Texte analysieren
- Texte umschreiben
- Texte komplettieren
- Texte klassifizieren
- Übersetzen
- Programmieren
- Ideen sammeln über verschiedene Domains

Das schafft es dadurch, dass es Milliarden von Texten, wie Bücher, wissenschaftliche Arbeiten und Internetwebseiten in seinem Training analysiert und daraus gelernt hat [1].

Ein Use Case, welches in letzter Zeit immer in Verbindung mit ChatGPT gebracht wurde, ist Data preprocessing [2]. Immer mehr Data Scientist setzten sich mit der Möglichkeit auseinandergesetzt, Large Language Models (LLMs), wie ChatGPT, zu verwenden, um übliche Aufgaben in einem Data preprocessing Ablauf zu automatisieren [14]. Dabei gibt es bereits Forschung für Data cleaning, Augmentation, Imputing, Analysing, und Transformation [2,3]. Ein Aufgabenbereich, der bisher weitestgehend noch nicht untersucht wurde, ist das Abschätzen der Wertebereiche von Featuren [2].

Die Wertebereiche eines Features zu wissen, kann Nutzern viele Einsichten in den Datensatz geben. Sie kann ein hervorragendes Maß sein, wenn eine intuitive Statistik benötigen wird, die angibt, wie stark die Daten verteilt sind [4]. Zudem kann es unerfahrenen Nutzern, die noch keine Data Science Erfahrung haben, helfen Outlier oder invalide Werte im Datensatz zu erkennen [4]. Dafür gibt es bereits Methodiken, welche Statistische Methoden benutzen. Das Problem an dieser Herangehensweise ist, dass diese Methodiken die Domain, in der sich der Datensatz befinden nicht berücksichtigen. Was dazu führen kann, dass Schätzungen eines Wertebereiches Falsch sind. Es können Invalide Wertebereiche entstehen, wie z.B. das die Höhe eines Materials einen negativer Wert sein kann. Deshalb sollen LLMs eingesetzt werden, um die Schätzungen der Wertebereiche zu übernehmen. Denn diese haben durch ihr weites Spektrum an Trainingsdaten, bereits tiefes Domainwissen in einigen Fachbereichen.

II. Technologie

Large Language models (LLMs) ist die Verwendung verschiedener statistischer und probabilistischer Techniken zur Bestimmung der Wahrscheinlichkeit, dass eine bestimmte Wortfolge in einem Satz vorkommt. Sprachmodelle analysieren Texte in einem Trainingsablauf, um diese Wortvorhersagen machen zu können. Sie werden in Anwendungen zur Verarbeitung natürlicher Sprache (NLP) eingesetzt, insbesondere in solchen, die Text als Ausgabe erzeugen. Einige dieser Anwendungen sind das Übersetzen und die Beantwortung von Fragen [5,6].

Sprachmodelle bestimmen die Wortwahrscheinlichkeit durch die Analyse von Textdaten. Sie interpretieren diese Daten, indem sie sie durch einen Algorithmus leiten, der Regeln für den Kontext in der natürlichen Sprache aufstellt. Anschließend wendet das Modell diese Regeln bei Sprachaufgaben an, um neue Sätze genau vorherzusagen oder zu produzieren. Das Modell lernt im Wesentlichen die Merkmale und Eigenschaften der Grundsprache und verwendet diese Merkmale, um neue Sätze zu verstehen [6].

Moderne LMs, die auf neuronalen Netzen basieren, verwenden sehr große Modellarchitekturen (z.B. 175 Milliarden Parameter [1,2,5]) und trainieren auf riesigen Datensätzen (z.B. fast ein Terabyte englischer Text. Diese Skalierung erhöht die Fähigkeit von LMs, flüssige natürliche Sprache zu erzeugen und ermöglicht es ihnen auch, auf eine Vielzahl anderer Aufgaben angewendet zu werden, selbst ohne Aktualisierung ihrer Parameter oder Änderung der Modellarchitektur [5,6].

A. Tokenization

Als Menschen nehmen wir Text als eine Sammlung von Wörtern wahr. Sätze sind Abfolgen von Wörtern [6]. Dokumente sind Abfolgen von Kapiteln, Abschnitten und Absätzen. Für Computer ist Text jedoch lediglich eine Abfolge von Strings [6]. Um Maschinen ebenfalls die Fähigkeit zu geben ganze Kontexte zu verstehen, muss ein Modell erstellt werden, welches Texte erfasst und daraus lernt. Dieses Modell muss jedes Wort oder Zeichen eines nach dem anderen verarbeiten und nur am Ende eine Ausgabe generieren, wenn der ganze Text eingelesen wurde. Um solch ein Modell zu kreieren kann man die Technologie der reccurend neuronal Networks (rnn) verwenden. Welche dazu entwickelt wurden, um mit Daten umzugehen, die sequenziell in das Neuronale Netzwerk eingespeist werden. Sie besitzen außerdem die Fähigkeit sich Informationen „Merken“ zu können. Das Problem an rnns ist, dass diese oft am Ende eines Text vergessen, was am Anfang stand. Denn Ihre Fähigkeit sich Sachen „Merken“ zu können, nicht so ausgebaut ist wie beim Menschen.
[6].

Im Jahr 2017 veröffentlichten Vaswani et al. eine Arbeit mit dem Titel "Attention is All You Need", in dem ein Transformer-Modell entwickeln wurde [7]. Es basiert auf dem Aufmerksamkeitsmechanismus. Im Gegensatz zu reccurent neuronal networks ermöglicht der Aufmerksamkeitsmechanismus, den gesamten Satz (oder sogar den Absatz) auf einmal zu sehen und nicht nur ein Wort. Dadurch kann das Transformer-Modell den Kontext eines Wortes besser verstehen. Viele moderne Sprachverarbeitungsmodelle basieren auf Transformern [6,7].

Um mithilfe eines Transformer-Modell einen ganzen Text verarbeiten zu können muss dieser zuerst so verändert werden das, dass Modell damit umgehen kann. Dabei werden Sequenzen der Wörter in Tokens umgewandelt. Diese werden durch einen Encoder in Embeddings umgewandelt. Embeddings sind eine Darstellungsform der Tokens in Form von Vektoren. Als nächstes wandelt der Encoder diese Embeddings in eine Kontextvektor um [7].

Der Kontextvektor gilt als Grundbaustein des Transformer-Modells. Durch das Benutzen des Kontextvektors, generiert der Decoder des Transformer-Modells eine Ausgabe anhand von Regeln. Es kann z.B. der originale Input als eine Art Regel dem Decoder gegeben werden, mithilfe dessen wird es diese Regeln analysieren und ein Wort kreieren, welches zum Kontext der Regeln passt. Dieser Prozess wird so lange ausgeführt, bis die gewünschte Ausgabe erstellt wurde [6,7].

Dieser Prozess wir auch autoregressive generation genannt und ist die Art und Weise wie Transformer-Modelle es schaffen, mithilfe der Kontextvektoren sehr lange und komplizierte Eingabe zu verarbeiten und gegebene Ausgaben zu erzeugen.

B. Warum können Transformer einen Text Vorhersagen

In dem Beitrag "Unreasonable Effectiveness of Recurrent Neural Networks" [8] zeigte Andrej Karpathy, dass reccurent neuronal networks das nächste Wort eines Textes recht gut vorhersagen können. Das liegt nicht nur daran, dass es in der menschlichen Sprache Regeln gibt (z. B. die Grammatik), die die Verwendung von Wörtern an verschiedenen Stellen eines Satzes einschränken, sondern auch daran, dass es in der Sprache Redundanz gibt [8].

Durch Claude Shannons einflussreicher Arbeit "Prediction and Entropy of Printed English" hat die englische Sprache eine Entropie von 2,1 Bit pro Buchstaben, obwohl sie 27 Buchstaben (einschließlich Leerzeichen) hat [9]. Würden die Buchstaben nach dem Zufallsprinzip verwendet, läge die Entropie bei 4,8 Bits, so dass es einfacher wäre, vorherzusagen, was in einem Text in menschlicher Sprache als nächstes kommt. Modelle des maschinellen Lernens und insbesondere Transformer-Modelle sind in der Lage dieses Konzept zu adaptieren und dadurch vorhersagen zu machen [6,9].

Durch Wiederholung dieses Prozesses kann ein Transformatormodell einen kompletten Absatz Wort für Wort generieren. Was aber ist Grammatik aus der Sicht eines Transformationsmodells [6]? Im Wesentlichen beschreibt die Grammatik, wie Wörter in der Sprache verwendet werden, indem sie in verschiedene Wortarten eingeteilt werden und eine bestimmte Reihenfolge innerhalb eines Satzes erfordern. Trotzdem ist es schwierig, alle Regeln der Grammatik aufzuzählen. In Wirklichkeit speichert das Transformer-Modell diese Regeln nicht explizit, sondern eignet sie sich implizit durch Beispiele an. Es ist möglich, dass das Modell über die Grammatikregeln hinaus lernt und sich auf die in den Beispielen dargestellten Ideen erstreckt, aber das Transformatormodell muss groß genug sein, um das zu schaffen [6].

C. *Aufbau eines Large Language Models*

Ein großes Sprachmodell ist ein Transformationsmodell in großem Maßstab [6]. Es ist so groß, dass es normalerweise nicht auf einem einzigen Computer ausgeführt werden kann. Daher ist es ein Dienst, der über eine API oder eine Webschnittstelle bereitgestellt wird [6]. Wie zu erwarten, hat ein solch großes Modell aus einer riesigen Textmenge gelernt, bevor es sich die Muster und Strukturen der Sprache merken kann.

So wurde beispielsweise das GPT-3-Modell, das dem ChatGPT-Dienst zugrunde liegt, anhand riesiger Mengen von Textdaten aus dem Internet trainiert. Dazu gehören Bücher, Artikel, Websites und verschiedene andere Quellen [6]. Während des Trainingsprozesses lernt das Modell die statistischen Beziehungen zwischen Wörtern, Phrasen und Sätzen, so dass es kohärente und kontextuell relevante Antworten auf eine Aufforderung oder Anfrage generieren kann [6].

Aus dieser riesigen Menge an Text kann das GPT-3-Modell mehrere Sprachen verstehen und verfügt über Wissen zu verschiedenen Themen [6]. Aus diesem Grund kann es Texte in verschiedenen Stilen produzieren.

D. *GPT-3 API*

APIs (Application Programming Interface) ist eine Methodik, die es zwei Software-Komponenten ermöglicht über verschiedene Definitionen und Protokollen miteinander zu kommunizieren [10]. In diesen Definitionen und Protokollen ist festgelegt, wie auf Informationen zugegriffen werden darf. Die beiden Software-Komponenten, sind jeweils immer einmal eine Software-Komponente, welche eine Information zur Verfügung stellt und eine Software-Komponenten, welche diese Information extern abrufen möchte [10].

Das Model, welches in dieser Arbeit benutzt wird, ist das von Openai kreierte LLM Chat-GPT3.5-turbo. Welches über die von Openai zur Verfügung gestellte API aufgerufen wurde. Die API hat eine eigene Python library, die es ermöglicht ein Python Objekt zu erzeugen, welches alle benötigten Funktionen besitzt um das Chat-GPT-3.5-turbo Model zu benutzen.

E. *Regex*

Reguläre Ausdrücke, kurz Regex oder Regexp genannt, Ist eine leistungsfähige Methode, um nach Strings zu suchen oder diese zu manipulieren, insbesondere bei der Verarbeitung von ganzen Textdateien. Eine Zeile Regex kann leicht mehrere Dutzend Zeilen Programmiercode ersetzen [11].

Regex wird von allen Skriptsprachen (wie Perl, Python, PHP und JavaScript) sowie von allgemeinen Programmiersprachen wie Java und sogar von Textverarbeitungsprogrammen wie Word zum Durchsuchen von Texten unterstützt [11].

III. Experimentaufbau

Das Ziel dieser Arbeit ist es herauszufinden, wie gut LLMs mögliche Wertebereiche eines Features schätzen können. Alter kann z.B. nicht einen Wert unter 0 haben aber auch nicht über ca. 130. Um dies zu erreichen, wurden drei Experiment Stufen vorbereitet. In diesen drei Stufen wurde dem LLM wenig bis viele Informationen rund um den Datensatz gegeben. Ein Teilziel ist es herauszufinden, wie viel wissen ein LLM braucht, um herauszufinden in welcher Domain es sich befindet und ab welchem Informationsstand es gute Intervalle schätzen kann. Im ersten Experiment wird dem LLM nur der Name des Features gegeben. Beim zweiten Experiment wird dem LLM der Name des Features gegeben, sowie alle anderen Feature Namen aus dem Datensatz. Im letzten Experiment wird dem LLM der Name des Features, der Name alle anderen Features und eine Beschreibung des Datensatzes gegeben. Alle Experimente werden zweimal ausgeführt. Einmal für den Datensatz mit dem bekannten Featuren und einmal für den Datensatz mit dem unbekannten Featuren.

Zur Durchführung der beiden Durchläufe wurden drei verschiedene Prompts geschrieben, welche für jede Experimentstufe individuell angepasst wurden. Damit die Experimente in den beiden Durchläufen vergleichbar sind wurden die Prompts während der Durchführung Phase nicht verändert. Dies sollte dazu beitragen konstante Ergebnisse zu erzeugen, die vergleichbar sind.

Durch die Prompts wird versucht den Antworten zusätzlich eine konstante Form zu geben: „Minimum: X Maximum: Y“. Dadurch das ChatGPT aber eine Zufalls Komponente eingebaut hat, weichen die Antworten stark von der Gewünschten Form ab. Obwohl in den Prompts steht, dass die Antworten so kurz wie möglich sein sollen, gibt ChatGPT oft lange Antworten. In denen er die Antwort umschreibt, so dass es für den Benutzer

wie ein Menschlicher Chatpartner klingt oder die Antwort nicht und man bekommt einen langen Text in dem dies drinsteht. Die Datensätze mit den Bekannten und den Unbekannten Daten, haben die gleichen Prompts bekommen. Damit aus beiden Datensätzen die Ergebnisse vergleichbar sind wurden die Prompts so gewählt, dass diese in jeder Experimentstufe nur die nötigsten Informationen hinzugefügt werden.

- „Try to set a range of values for the feature: [name_of_the_features] that comes from this dataset. Even if you don't know just guess something. Only give the minimum and maximum in the answer. The answer should be as short as possible."

- "Here is a list of all features of a dataset: [List of all the other Featurenames in the Dataset]. Try to set a range of values for the feature: [name_of_the_features] that comes from this dataset. Even if you dont know just guess something. Only give the minimum and maximum in the answer. The answer should be as short as possible."

- "Try to set a range of values for the feature: [name_of_the_features] that comes from a dataset. Keep the answer as short as possible, just show the min and the max. Here is the Dataset description: [Description of the Dataset]. And here is a list of all Features of this Dataset: [List of all the other featurenames in the Dataset]. "

Es wurde ein Python Script geschrieben, welches durch die Openai API ein Objekt erstellt, mit dem es möglich ist, mit dem ChatGPT-Modell zu interagieren. Dieses Script wird insgesamt dreimal für jeweils den bekannten und unbekannten Datensatz ausgeführt. In jeder Ausführung wird dem Model eine der oben genannten Prompts + ein Featurename + die jeweiligen anderen Informationen pro Experimentstufe zugewiesen. Das Modell nimmt diese Prompts + weitere Informationen an, verarbeitet diese und sendet dem Objekt eine Antwort zurück. In dieser Antwort sind die gewünschten geschätzten Wertebereiche enthalten.

Um aus dem Text die benötigten Informationen zu extrahieren, wie oben bereits genannt gibt das Model oft nicht relevanten Text mit, wurde hier mit der Natural language Processing Methode des Regular Expression Matching (Regex) gearbeitet. Mit Regex ist es möglich aus einem Text eine beliebige Zeichenkette, mit selbstdefinierten Syntaxregeln, zu finden. Durch Regex ist es möglich die Wertebereiche aus den Antworten zu extrahieren.

Die Antworten aus dem LLM werden dann jedem Feature von dem sie Stammen zugeordnet und gespeichert. So dass in einer Reihe des Datensatzes der Wertebereich aus dem Datensatz steht und daneben die Ergebnisse aus den drei Experimenten.

Nach dem Extrahieren der Wertebereiche aus den Antworten, wird für jedes Feature der Jaccard Index (JI) berechnet [12,23]. JI ist eine Messtechnik, die es erlaubt verschiedene Mengen, in diesem Fall auch Intervalle (siehe Kapitel Messtechnik), miteinander zu vergleichen. Beim berechnendes JI wird als „Grund Truth" Wertebereich, der Wertebereich aus den Datensätzen verwendet. Diese wird mit dem aus den Antworten extrahierten Wertebereichen verglichen. Insgesamt hat jedes Feature, für jedes Experiment, einen eigenen Jaccard Index Wert. Für jedes Experiment wird der Durchschnittliche JI [12,23] berechnet. Diese werden anschließend im Kapitel Ergebnisse miteinander verglichen.

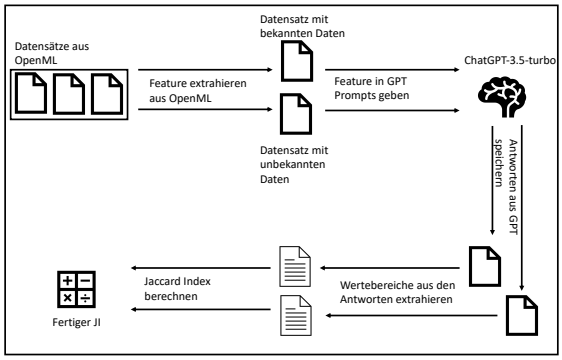

Figure 1 Experimentaufbau

IV. Daten

Für die Durchführung der Experimente sind Tabulare Daten nötig welche numerische Werte enthalten. Um diesen Bedarf an Daten zu decken, wurde mithilfe der OpenML API ein Skript geschrieben welches, die Datensätze von OpenML durchforstet, geeignete Datensätze findet und die nötigen Informationen aus den Datensätzen extrahiert. Für die Experimente wurden Folgende Informationen aus den Datensätzen extrahiert: Der Name von jedem Feature welches nur numerische Werte enthält. Diese sind nötig für das Abfragen der LLMs. Dazu für jedes extrahierte Feature

der minimale und maximale Wert, diese Werte aus dem Datensatz bilden die „Ground Truth“. Außerdem den Namen und die ID des Datensatzes, um Duplikate zu vermeiden.

Dadurch das ChatGPT mit Webseiten aus dem Internet trainiert wurde, darunter auch OpenML, ist die Wahrscheinlichkeit groß das ChatGPT die abgefragten Datensätze/Feature bereits kennt.

Deswegen werden zwei Datensätze kreiert, ein Datensatz welcher Datensätze enthält die ChatGPT kennt und ein Datensatz welcher Datensätze enthält die ChatGPT nicht kennt. Beide Datensätze werden bei den Experimenten benutzt, es soll verglichen werden wie ChatGPT abschneidet im Vergleich zu Datensätzen, die es bereits kennt.

Der Datensatz mit den bekannten Daten hat insgesamt 386 Feature die aus 29 unterschiedlichen Datensätzen stammen. Der Datensatz mit den unbekannten Daten hat insgesamt 463 Feature die aus 28 unterschiedlichen Datensätzen stammen. Aus Tabelle 1 kann die Form der Datensätze entnommen werden.

Table 1 Form der Datensätze

Name	Dataset ID	Min/ Max	Description	guessed Values
density	871	-12/10	Der Datens..	0/10
…	…	…	…	…

V. MESSTECHNIK

Um die Ergebnisse aus den verschiedenen Experimenten miteinander vergleichen zu können, muss eine Messtechnick gewählt werden, welche uns Auskunft darüber gibt wie gut die Experimente performt haben. Dadurch das wir Intervalle haben, wäre es mit herkömmlichen Messtechnicken zwar möglich einen Error zu berechnen, diesen müsste man aber separat machen für einmal den minimalen und einmal den Maximalen Wert. Das Endresultat wäre zwei Error Scores, die aber nicht vergleichbar wären. Aus diesem Grund wird die Jaccard Index [12,23]. Der Jaccard Index ist ein Ansatz, der es ermöglicht zwei Mengen miteinander zu vergleichen. Dabei bildet man die Schnittmenge (Abbildung. 1) und die Vereinigungsmenge (Abbildung. 2) zwischen den beiden Mengen [12,23]. Anschließend teilt man die Schnittmenge durch die Vereinigungsmenge [12,23]. Als Ergebnis bekommt man einen Wert zwischen 0 und 1, der widerspiegelt, wie ähnlich sich die beiden Mengen sind. 1 bedeutet die Mengen sind identisch. 0 bedeutet die Mengen schneiden sich gar nicht [12,23]. Werte dazwischen geben prozentual an wie ähnlich sich beide Mengen sind.

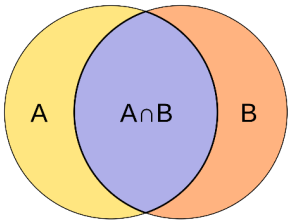

Figure 3 Schnittmenge

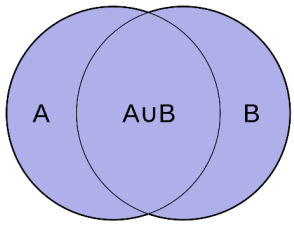

Figure 2 Vereinigungsmenge

Das Vereinigungsintervall bildet sich aus zwei reellen Intervallen und besteht aus der Vereinigung ihrer Mengen, die aus allen Elementen besteht, die zum ersten Intervall gehören, und aus allen Elementen, die zum zweiten Intervall gehören. Es wird die kleinste und größte grenze aus beiden Intervallen genommen (siehe Abbildung 3 und Formel 1) [12,23].

$$(X, Y) = (A, B) \cup (C, D) \quad (1)$$

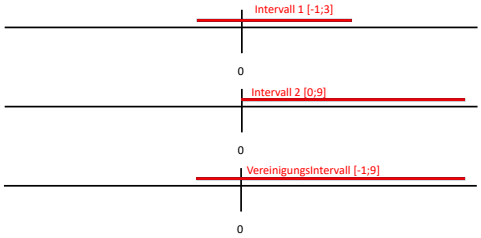

Figure 4 Example of Union Interval

Das Schnittmengenintervall bildet sich aus zwei reellen Intervallen und besteht sich aus der Schnittmenge, die Menge aller Elemente, die zu beiden Intervallen gehören. Falls es keine Schnittmenge gibt, bleibt das Intervall leer und ihm wird das Symbol: Ø zugewiesen [12,23] (siehe Abbildung 4 und Formel 2).

$$(X, Y) = (A, B) \cap (C, D) \quad (2)$$

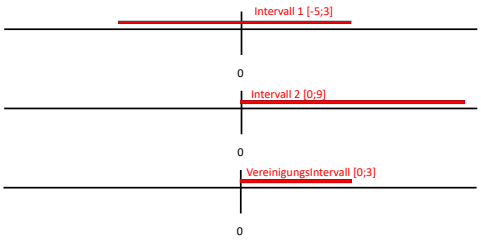

Figure 5 Example of Intersection Interval

Da es sich hierbei um Intervalle handelt kann nicht einfach die Vereinigungsmenge durch die Schnittmenge geteilt werden, denn in einem Intervall gibt es unendlich viele Zahlen. Aus diesem Grund wird die aus dem Vereinigungsintervall und dem Schnittintervall die Länge berechnet. Das erreicht man, indem man die rechte Grenze des Intervalls wird mit der Linken Grenze des Intervalls subtrahiert. Anschließend werden die Längen, welche aus den Schnitt- und Vereinigungsintervall gewonnen wurden, dividiert [12,23]. Das Ergebnis aus der Division ist der Jaccard Index.

VI. Probleme

Im Verlauf der Projektarbeit sind einige Probleme aufgetreten, welche die Ergebnisse direkt oder indirekt beeinträchtig haben. Beim Erstellen des Datensatzes ist das erste Problem aufgetreten. Die Wertebereiche, die aus dem Datensatz stammen spiegeln nicht den reellen Wertebereich der Domain aus dem der Datensatz stammt wider. Was dazu führt das die JI-Ergebnisse verzehrt sind. In einem der Datensätze könnte es sich um Kreditdaten handeln. In einem der Feature könnte das Alter benannt worden sein. Die im Datensatz enthaltene Minimal- und Maximalwerte könnten z.B. nur 23-54 sein. Falls ChatGPT dann 0-100 schätz, würde man einen schlechten JI wert erhalten. In den meisten Datensätzen sind die wahren Wertebereiche der Domains nicht enthalten. Sondern nur Teile der Wertebereiche.

Außerdem unterscheiden sich die Wertebereiche je nach Domain. Bei einem Kreditdatensatz kann Alter variieren zwischen 18-65, denn dies ist ein Alter, indem die Banken Kredite verteilen. Wenn man ChatGPT nur aber den featurename Alter gibt würde er 0-100 schätzen, was korrekt wäre, aber in dieser Domain Falsch. Deshalb sind die Ergebnisse aus Experiment 1 weitestgehend abhängig wie gut das Model abschätzen kann zu welcher Domain ein Feature Name gehört.

Ein weiters Problem sind die Antworten von ChatGPT. Das Model liefert unterschiedliche Antworten, bei der gleichen Eingabeprompt. Dadurch das es immer unterschiedliche Texte sind und die Wertebereiche immer unterschiedlich dargestellt werden. Gibt es Probleme beim automatischen Identifizieren und Extrahieren dieser Wertebereich. Es gibt zu viele Individual Fälle, um alle abzufangen. Dadurch mussten diese Händisch extrahiert werden, damit keine Informationen verloren gehen.

Die Beschreibung des Datensatzes enthält in einigen Fällen bereits die Wertebereiche der einzelnen Features. Was dazu führt das ChatGPT die Datensatz Beschreibungen zusammenfasst, analysiert und sich die Werteberreiche für jedes Feature rausnimmt. Das führt dazu das, dass Model nicht mehr selbst schätzt, sondern sich die Antwort aus dem Text zusammensucht. In vielen Fällen klappt das so gut, dass der JI auf einer 1 kommt, was dazu führt, dass der Durchschnittliche JI sich nach oben verzehrt. Dadurch das die Beschreibungen zu viele dazu noch lang und komplex sind. Ist es nicht möglich diese Händisch rauszufiltern.

Ein weiteres Problem sind Schätzungen in die Unendlichkeit. Das Model schätzt manche Features, positiv oder negativ unendlich. Was plausibel ist, aber Probleme in den Messungen bereiten. Falls ein Wert unendlich geschätzt wird, kommt in der Berechnung des JI ein Wert von null raus. Dadurch das die Union sich aus dem Beiden Maximal entferntesten Punkten beider Intervalle bildet, entsteht im Union Intervall, ein. Intervall ins unendliche. Was theoretisch korrekt sein könnte, denn es gibt Manche Werte die unendlich große/klein sein können. Praktisch führt es zum Problem das die „Grund Truth“ keiner solcher Werte besitzt dazu das es zu einem JI von 0 kommt.

Das größte Problem war der Mangel an geeigneten Daten. Es gibt haufenweise Datensätze mit guten Featurenamen, doch die meisten sind so alt, dass ChatGPT diese meisten bereits kennt. Es konnte mit Leichtigkeit ein Datensatz mit bekanntem Featuren erstellt werden. Doch das Erstellen des Datensatzes mit unbekannten Daten hat, durch das Händische suchen, lange gebraucht. Denn zwei Probleme sind häufig aufgekommen. Einmal das ChatGPT den Datensatz bereits kennt und dass die meisten Datensätze ungepflegt waren. Es wurden mehrere tausend Feature gefunden, diese waren aber für das Experiment kaum nutzbar. Die Feature hatten keine richtigen Namen, oft

nur Aufzählungen, sprachlich unterverständliche Experten orientierte Wörter, Nummerierungen oder interne Abkürzungen, welche in der Beschreibung nicht erläutert wurden. Das führt dazu das diese Datensätze mit ihrem Featuren unbrauchbar waren, weil ChatGPT diese Namen nicht schätzen kann.

VII. Ergebnisse

Die Ergebnisse aus den zwei versuchen, einmal bekannte und einmal unbekannte Datensätze, werden in diesem Kapitel näher betrachtet. Die Ergebnisse sind durch die nicht korrekte „Ground Truth“ etwas verzehrt, können uns aber ungefähr aufzeigen, wie gut das Model performt hat. Wie in Abbildung 5 und 6 zu sehen ist, sind die JI werte, beim Experiment 1 nur im 30-40 % Bereich. ChatGPT zeigt, dass wenn es nur den Feature Namen bekommt, keine genauen Angaben über die Wertebereiche machen kann. Die Ergebnisse mit 33% im Trainingsdatensatz und 38 % im Testdatensatz sind ähnlich und weißen keine Unterschiede auf. Der Unterschied von 5% kann hier durch unterschiedliche Faktoren entstehen. Ein Faktor wäre das die Namen in einem der beiden Datensätze besser gewählt wurde als beim anderen.

Beim zweiten Experiment erreichte der Datensatz mit den bekannten Feature 42% und der Datensatz mit den unbekannten Feature 44%. Die Ergebnisse unterscheiden sich kaum, zeigen aber wie im ersten Experiment einen minimalen unterschied auf.

Beim dritten Experiment erreichte der Datensatz mit den bekannten Feature 68% und der Datensatz mit den unbekannten Feature 54%. Die Experimente zeigen hier einen signifikanteren unterschied. Der bekannte Datensatz, schneidet in diesem Experiment besser ab als der Datensatz mit den unbekannten Daten. Der bekannte Datensatz zeigt mit 68% gute Schätzungen auf. Das könnte aber daran liegen, dass, das Model diese Daten bereits kennt. Dadurch das es sich hierbei um ein Black Box Model handelt, also ein Machine Learning Model, welches die erzeugten Lösungen nicht nachvollziehbar sind, kann nicht mit Sicherheit gesagt werden, ob das Model die Daten kennt und deshalb so akkurat antwortet oder ob es gut schätzt [15]. Beim unbekannten Datensatz sehen wir ein eher mäßig gutes Ergebnis, mit 54 %.

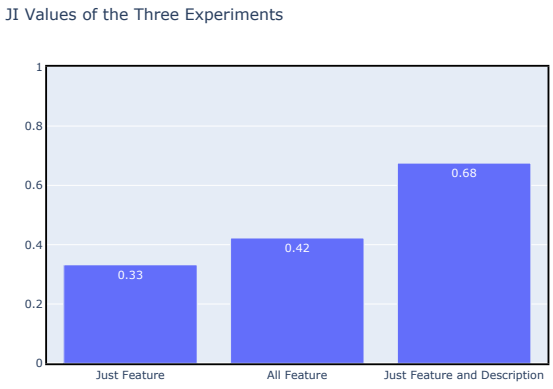

Figure 6 Ergebnisse für jede Experimentstufe aus dem Datensatz mit bekannten Featuren

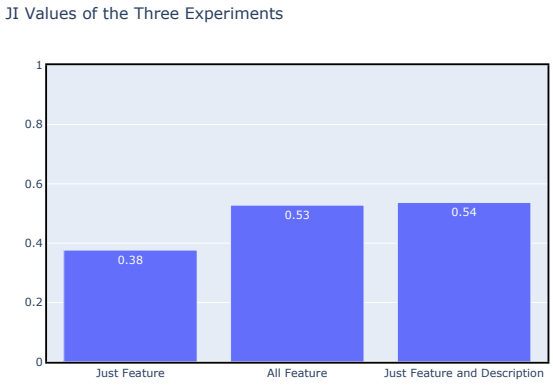

Figure 7 Ergebnisse für jede Experimentstufe aus dem Datensatz mit unbekannten Featuren

Aus Abbildung 3 und 4 ist zu entnehmen das Falls man ChatGPT einsetzen möchte, um die Wertebereiche abzuschätzen, es die besten Ergebnisse liefert, wenn man es mit genügend Informationen über den Datensatz versorgt. Die Experimentstufen 1 und 2 sind in fast allen Szenarien kaum Einsatzbar. Dagegen sieht Experimentstufe 3

Abbildung 7-10 zeigen die Verteilung der Wertebereiche, aus den ChatGPT antworten. In Abbildungen 7 und 8 sehen wie die die Verteilung der unteren Grenze der geschätzten Wertebereiche, für bekannte und unbekannte. Die Abbildungen 9 und 10 zeigen die obere Grenze der Wertebereiche, für bekannte und unbekannte.

Betrachtet man Abbildung 7 genauer an, sieht man das bei den unteren Grenzen, in jeder Experimentstufe die Verteilung der Werte nahe null liegt. Es gibt kaum

ausschweifende Werte. Das gleiche können wir in der Abbildung 8 sehen. Dort reihen sich auch fast alle Werte nahe null ein. Das bedeutet das Modell schätzt, falls es sich um ein Minimum handelt, sehr oft den Wert null. Es ist eine gute Schätzung, denn in diesen spezifischen Fällen ist die „Ground Truth“ oft Null. In dem bekannten Datensatz haben wir in der „Ground Truth“ über 174-mal den Wert Null. In dem Datensatz mit unbekannten Werten, 122-mal den Wert Null. Was aber zu der Frage führt, schätzt das Model tatsächlich jedes Features einzeln. Oder gibt es immer wieder mit null die gleiche Antwort. ChatGPT ist ein Blackbox Modell welches uns, vor allem Außenstehende keinen Einblick in seine Entscheidungsfindungsvorgang gibt. Deshalb können wir nicht genau sagen wie die Ergebnisse Zustandekommen.

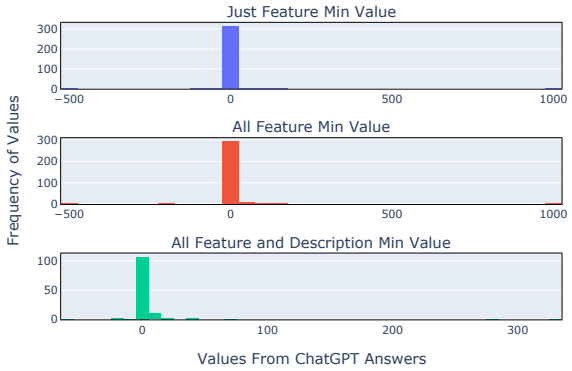

Figure 8 Verteilung der ChatGPT Antworten für den bekannten Datensatz. Abgebildet sind die Minimalen Werte.

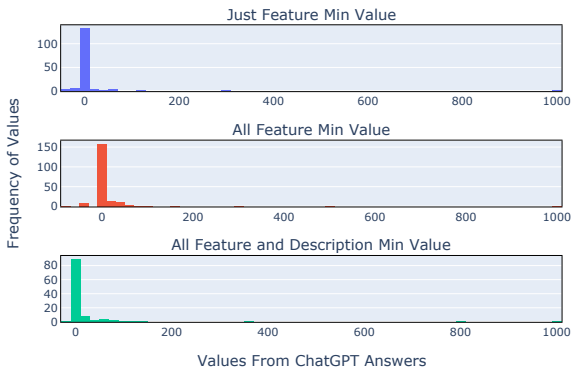

Figure 9 Verteilung der ChatGPT Antworten für den unbekannten Datensatz. Abgebildet sind die Maximalen Werte

Wenn man die Verteilungen der oberen Grenzen aus den Abbildungen 9 und 10 betrachtet, erkennt man im Vergleich zu den unteren Grenzen eine gleichmäßigere Verteilung. Es gibt zwar Ansammlungen bei den Werten 1,10 und 100 dennoch sind die Werte eher verteilt. Vor allem in der Experimentstufe 1 und 2 sind diese Ansammlungen zu erkennen. In den „Ground Truth“ werten, ist der Wert 1, 75-mal enthalten und der Wert 10 85-mal. Den Wert 100 gibt es nur 27-mal. Daraus ist zu schließen, das ChatGPT bei geringer Informationslage oft ganze Zahlen schätzt.

In der Experimentstufe 3 ist in beiden Abbildungen eine gleichmäßigere Verteilung zu sehen, was auch im höheren JI wert zu erkennen ist. Das Model kann mit mehr Informationen genauere Schätzungen vornehmen.

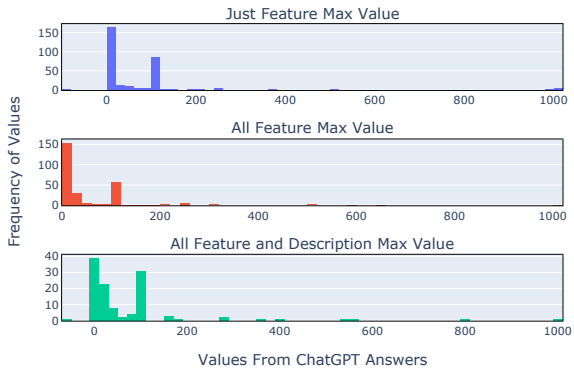

Figure 10 Verteilung der ChatGPT Antworten für den unbekannten Datensatz. Abgebildet sind die Minimalen Werte.

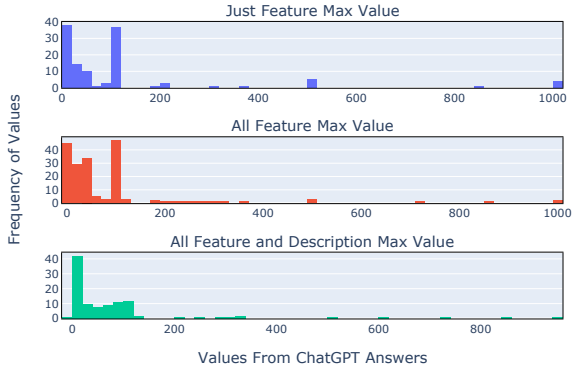

Figure 11 Verteilung der ChatGPT Antworten für den unbekannten Datensatz. abgebildet sind die Maximalen Werte

VIII. Weiterführung der Arbeit

Durch die limitierte Zeit konnte diese Arbeit nicht Ihren vollständigen Umfang erreichen. Dennoch gibt es weitere ungenutzte Möglichkeiten, um diese Projektarbeit weiterzuentwickeln. Neue Ansätze könnten untersucht, bisher unerforschte Aspekte beleuchtet und zusätzliche Quellen hinzugezogen werden. Die Integration weiterer verschiedener Aspekte und die Verwendung unterschiedlicher Methoden könnten zu neuen Erkenntnissen führen. Durch die Einbindung von aufkommenden Trends könnte diese Arbeit zukünftig bessere Ergebnisse hervorbringen als bisher. Es gibt drei Aspekte, die zum jetzigen Zeitpunkt diese Arbeit verbessern könnten:

- Der Ausbau der Datensätze. Wie im Kapitel Probleme besprochen, sind der Mangel an geeigneten Datensätze ein großes Problem. Vor allem gibt es einen immensen Mangel an unbekannten Datensätzen die geeignete Feature haben. Zwar gibt es eine Menge von Datensätzen, diese sind jedoch ungepflegt und die featurenamen sind eher Aufzählungen, interne Abkürzungen oder sprachlich unterverständliche Experten orientierte Wörter, mit denen das Model nichts anfangen kann. Dadurch dass es so schwer ist geeignete Feature zu finden war das Limit der Feature in dieser Arbeit 436. Zwar wurden mehrere 10 000 Features gefunden, diese haben aber nicht den oben aufgezählten Kriterien entsprochen. Was dazu geführt hat, dass diese ausgeschlossen wurden.

- Der Einsatz weiterer Modelle. ChatGPT-3.5-turbo wurde in dieser Arbeit hauptsächlich wegen seiner einfachen Handhabung und seiner kostenlosen Prompts benutzt. Es gibt mittlerweile weitere Modelle, welche man in dieser Arbeit benutzen könnte. Diese sind leider meisten entweder gebunden an einem einzelnen Betriebssystem welche spezielle Hardware voraussetzt oder kostet Geld für den Gebrauch. Diese Modelle, wie z.B. ChatGPT-4/BingChat/Aplaca usw., sehen aber durch ihre erhöhte Leistung im Vergleich zu ChatGPt-3.5 vielversprechend aus. Der Einsatz verschiedener Modelle könnte bessere Ergebnisse als bisher hervorbringen und vielleicht Unterschiede zwischen den einzelnen Modellen aufzeigen. Ein Problem könnte es hier aber geben, die neuen Modelle wie BingChat und ChatGPT-4 besitzen eine Koppelung an das Internet. Das könnte dazu führen das diese LLMs alle Datensätze bereits „kennen“. Bringt aber auch den Vorteil mit sich, dass sie Domainwissen über spezifische Fachbereiche einfacher finden können.

- Der Vergleich mit herkömmlichen Methoden. Um die bessere Methodik zu finden beim Schätzen von Wertebereichen, wäre es durchaus Plausibel herkömmliche Methoden zu benutzen. Diese sollten auf die Features eingesetzt werden, um die Wertebereiche zu erlangen. Die daraus gewonnenen Wertebereiche könnten dann mit der „Ground Truth“ aus den Daten verglichen werden. Der daraus gewonnene JI kann dann beispielsweise mit dem vom ChatGPT verglichen werden, um herauszufinden welche Methodik am besten geeignet ist.

IX. Diskussion

Um ein Geeigneten Wertebereich schätzen zu können, benötigt das Chat-GPT3 Model einiges an Informationen. Wie in den Ergebnissen aufgezeigt, kann das Model nur mit dem featurename oder mit allen anderen featurenamen aus dem Datensatz keine guten Schätzungen durchführen. Das Model benötigt mehr Informationen, wie eine Beschreibung des Datensatzes, eine Beschreibung der Domain oder eine Ausführliche Beschreibung, was das Model machen soll. Falls einer dieser Punkte erfüllt sind, werden die Ergebnisse dementsprechend besser, wie in den Ergebnissen von Experiment 3 zu sehen ist. Das Bedeutet zum jetzigen Zeitpunkt ist das Model nicht geeignet, um Wertebereiche von Featuren zu schätzen. Dennoch sind die Ergebnisse, mit Aussicht für die Zukunft, viel versprechend. Durch die ständige Verbesserung der LLMs und der konstant wachsenden Gemeinde, welche sich mit dem Training und der Optimierung dieser Modelle widmet, können in naher Zukunft die Modelle auf ein Niveau gebracht werden, in denen sie bei jedem Datensatz verstehen um Welches Thema es sich dabei handelt, wie die Features sich in der Domain sich verhalten und akkurat schätzen können welche Wertebereiche hierbei am besten geeignet sind.

Das Modell benötigt, um gute Ergebnisse zu liefern viele externe Domain spezifische Informationen. Welche man dem Modell mitgeben muss. Diese Studienarbeit hat sich darauf bezogen, Nutzern die kaum Wissen im Data Science Bereich und über den Datensatz aufweisen, dabei zu unterstützen Wertebereiche zu schätzen. Durch die Ergebnisse ist es klar, dass, dass Modell dies nicht zum jetzigen

Zeitpunkt erfüllen kann, dies kann sich aber in der Zukunft ändern.

REFERENCES

[1] L. Turoňová, L. Holík, O. Lengál, O. Saarikivi, M. Veanes, und T. Vojnar, „Regex matching with counting-set automata“, *Proc. ACM Program. Lang.*, Bd. 4, Nr. OOPSLA, S. 1–30, Nov. 2020, doi: 10.1145/3428286.

[2] G. Jaimovitch-López, C. Ferri, J. Hernández-Orallo, F. Martínez-Plumed, und M. J. Ramírez-Quintana, „Can language models automate data wrangling?“, *Mach Learn*, Bd. 112, Nr. 6, S. 2053–2082, Juni 2023, doi: 10.1007/s10994-022-06259-9.

[3] N. T. Nguyen, K. Jearanaitanakij, A. Selamat, B. Trawiński, und S. Chittayasothorn, Hrsg., Intelligent Information and Database Systems: 12th Asian Conference, ACIIDS 2020, Phuket, Thailand, March 23–26, 2020, Proceedings, Part I, Bd. 12033. in Lecture Notes in Computer Science, vol. 12033. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-41964-6.

[4] Fim Frost, „Range of a Data Set“, *Statistics with JF*, 12. August 2023. https://statisticsbyjim.com/basics/range/#:~:text=The%20range%20of%20a%20data,but%20it%20has%20some%20limitations. (zugegriffen 12. August 2023).

[5] T. B. Brown *u. a.*, „Language Models are Few-Shot Learners“. arXiv, 22. Juli 2020. Zugegriffen: 12. August 2023. [Online]. Verfügbar unter: http://arxiv.org/abs/2005.14165

[6] A. Tam, „What are Large Language Models“, Zugegriffen: 12. August 2023. [Online]. Verfügbar unter: https://machinelearningmastery.com/what-are-large-language-models/

[7] A. Vaswani *u. a.*, „Attention Is All You Need“. arXiv, 1. August 2023. Zugegriffen: 12. August 2023. [Online]. Verfügbar unter: http://arxiv.org/abs/1706.03762

[8] Andreij Karpathy, „The Unreasonable Effectiveness of Recurrent Neural Networks“, 21. Mai 2015. http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[9] C. E. Shannon, „Prediction and Entropy of Printed English“.

[10] „Wie funktionieren APIs?“, *Was ist eine API?*, 2023. https://aws.amazon.com/de/what-is/api/

[11] L. Turoňová, L. Holík, O. Lengál, O. Saarikivi, M. Veanes, und T. Vojnar, „Regex matching with counting-set automata“, *Proc. ACM Program. Lang.*, Bd. 4, Nr. OOPSLA, S. 1–30, Nov. 2020, doi: 10.1145/3428286.

[12] A. N. Bazhenov und A. Yu. Telnova, „Generalization of Jaccard Index for Interval Data Analysis“, *Meas Tech*, Bd. 65, Nr. 12, S. 882–890, März 2023, doi: 10.1007/s11018-023-02180-2.

[13] K. Kim und N. Joukov, Hrsg., *Information Science and Applications 2017: ICISA 2017*, Bd. 424. in Lecture Notes in Electrical Engineering, vol. 424. Singapore: Springer Singapore, 2017. doi: 10.1007/978-981-10-4154-9.

[14] T. De Bie, L. De Raedt, J. Hernández-Orallo, H. H. Hoos, P. Smyth, und C. K. I. Williams, „Automating data science“, *Commun. ACM*, Bd. 65, Nr. 3, S. 76–87, März 2022, doi: 10.1145/3495256.

[15] S. Liu, T. D. Ullman, J. B. Tenenbaum, und E. S. Spelke, „Baby steps in evaluating the capacities of large language models“, *Science*, Bd. 358, Nr. 6366, S. 1038–1041, Nov. 2017, doi: 10.1126/science.aag2132.