1. **Results:**
   1. **Car Price Prediction:**

**Dataset:**

Total Set Shape: 8218 Rows

Train Set Shape: 5201 Rows

Valid Set Shape: 1301 Rows

Test Set Shape: 1626 Rows

Sampled Test Set Shape: 400 Rows

**Learnings:**

* Wie erwartet: Mit besserem Modell und besseren prompts verbessert sich das Ergebnis
* Fehler bei sehr guten Abfragen werden teilweise relativ gering – ggf. auf ein Dataset mit weniger Infos, nicht dem genauen Modell sondern nur der Brand abweichen?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | MAE | MSE | RMSE | R2 |
| Zero Shot | GPT-3.5-Turbo | 102.472 | 53.579.280.930 | 231.471 | (0.8964) |
| Zero Shot | GPT-4-Turbo-Preview | 86.124 | 27.711.918.630 | 166.468 | (0.9464) |
| Dynamic Few Shot | GPT-3.5-Turbo | 65.557 | 22.055.967.680 | 148.512 | (0.9574) |
| Dynamic Few Shot | GPT-4-Turbo-Preview | 60.372 | 12.884.017.255 | 113.507 | (0.9751) |
| Linear Regression | 205.091 | 102.227.548.422 | 319.730 | 0.8048 |

**Prompt Examples:**

**Zero – Shot example:**

System: Based on the provided features of a used car listed below, please predict its selling price in Indian Rupees in the Indian market. The predicted price should be expressed solely as a number followed by the currency "INR".

Ensure that the output contains no additional text or characters beyond this specified format.

Features:

name: Mahindra Scorpio VLX 2WD AIRBAG BSIII,

year: 2012,

km\_driven: 120000,

fuel: Diesel,

seller\_type: Individual,

transmission: Manual,

owner: First Owner,

mileage: 12.05 kmpl,

engine: 2179 CC,

max\_power: 120 bhp,

torque: 290Nm@ 1800-2800rpm,

seats: 8.0

Required Output:

"price": <predicted price> INR

Please provide the prediction strictly adhering to the above instructions.

**Few – Shot example:**

Example 1:

Features:

…

Output: "price": 459999 INR

Example 2:

Features:

…

Output: "price": 750000 INR

Example 3:

Features:

…

Output: "price": 600000 INR

System: Based on the provided features of a used car listed below, please predict its selling price in Indian Rupees in the Indian market. The predicted price should be expressed solely as a number followed by the currency "INR".

Ensure that the output contains no additional text or characters beyond this specified format.

Features:

name: Mahindra Scorpio VLX 2WD AIRBAG BSIII,

…

Required Output:

"price": <predicted price> INR

Please provide the prediction strictly adhering to the above instructions.

* 1. **Sentiment Analysis:**

**Dataset:**

Sampled Test Set Shape: 451 Entities

**Learnings:**

* Alle Abfragen machen genug Fehler
* Merkliche / Erklärbare Performance Unterschiede immer nur zwsichen Single-Term und Mutli-Term Abfragen 🡪 in Multi-Term wird “Neutral” wesentlich besser erkannt
* Sonst keine merklichen Performance Unterschiede – „Neutral“ immer mit schlechtestem F1 Score
* Keine Performance Steigerung durch Few Shot 🡪 Bester F1 Score bei Zero-Shot | Multi-Term | GPT-4

**Overall Metrics Table (Weighted Averages)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | Accuracy | Precision | Recall | F1-Score |
| Zero Shot | Single Term | GPT-3.5-Turbo | 0.59 | 0.6 | 0.59 | 0.57 |
| Zero Shot | Multi Term | GPT-3.5-Turbo | 0.64 | 0.66 | 0.64 | 0.62 |
|  |  |  |  |  |
| Zero Shot | Single Term | GPT-4-Turbo-Preview | 0.59 | 0.61 | 0.59 | 0.57 |
| Zero-Shot | Multi Term | GPT-4-Turbo-Preview | 0.67 | 0.69 | 0.67 | 0.67 |
|  |  |  |  |  |
| Fix Few Shot | Single Term | GPT-3.5-Turbo | 0.48 | 0.51 | 0.48 | 0.40 |
| Fix Few Shot | Multi Term | GPT-3.5-Turbo | 0.63 | 0.65 | 0.63 | 0.61 |
|  |  |  |  |  |
| Dynamic Few Shot | Single Term | GPT-3.5-Turbo | 0.63 | 0.63 | 0.63 | 0.63 |
| Dynamic Few Shot | Multi Term | GPT-3.5-Turbo | 0.63 | 0.65 | 0.63 | 0.62 |
|  |  |  |  |  |
| Dynamic Few-Shot | Single Term | GPT-4-Turbo-Preview | 0.58 | 0.62 | 0.58 | 0.56 |
| Dynamic Few Shot | Multi Term | GPT-4-Turbo-Preview | 0.67 | 0.67 | 0.67 | 0.66 |

**Prompt Examples:**

**Single Term | Zero - Shot:**

Amusing details distinguish desserts, from dulce de leche ice-cream profiteroles dotting a chocolate sauce tic-tac-toe board, to coconut custard surrounded by a sea of Malibu-rum gelee and poached pineapple.

What is the sentiment on 'dulce de leche ice-cream'? Only respond with "positive", "negative" or "neutral" as one word.

**Multi Term | Zero - Shot:**

Task: Analyze the sentiment of specific terms mentioned in a sentence.

You are required to evaluate whether the sentiment towards each term is 'positive', 'negative', or 'neutral'.

Sentence: "Amusing details distinguish desserts, from dulce de leche ice-cream profiteroles dotting a chocolate sauce tic-tac-toe board, to coconut custard surrounded by a sea of Malibu-rum gelee and poached pineapple."

Terms:

'desserts'

'dulce de leche ice-cream'

'chocolate sauce tic-tac-toe'

'poached pineapple'

Return the final result as JSON in the format {"term\_sentiments": "<a list of [term, sentiment] pairs>"}.

ONLY return the JSON.

Answer:

**Single Term | Few - Shot:**

Example 1:

Input: "The decor is not special at all but their food and amazing prices make up for it."

Term: decor

Output: negative

Example 2:

…

Example 3

…

Task:

Input: Amusing details distinguish desserts, from dulce de leche ice-cream profiteroles dotting a chocolate sauce tic-tac-toe board, to coconut custard surrounded by a sea of Malibu-rum gelee and poached pineapple.

Prompt: What is the sentiment in the text towards 'desserts'? Only respond with "positive", "negative" or "neutral" as one word.

**Multi Term | Few - Shot:**

Example 1:

Input: "The decor is not special at all but their food and amazing prices make up for it."

Terms:

'decor'

'food'

'prices'

Output: {"term\_sentiments": [["decor", "negative"], ["food", "positive"], ["prices", "positive"]]}

Example 2:

…

Example 3

…

Task:

Input: Amusing details distinguish desserts, from dulce de leche ice-cream profiteroles dotting a chocolate sauce tic-tac-toe board, to coconut custard surrounded by a sea of Malibu-rum gelee and poached pineapple.

Terms:

'desserts'

'dulce de leche ice-cream'

'chocolate sauce tic-tac-toe'

'poached pineapple'

Prompt: Analyze the sentiment of specific terms mentioned in a sentence.

You are required to evaluate whether the sentiment towards each term is 'positive', 'negative', or 'neutral'.

Return the final result as JSON in the format {"term\_sentiments": "<a list of [term, sentiment] pairs>"}.

ONLY return the JSON.

* 1. **Schema Matching:**

**… Bisher noch keine Ergebnisse, da Evaluation unklar**

1. **Fragen:**
2. **Car Price Prediction:**
   1. Reicht hier Linear Regression (bzw. anderer Regressor wie RandomForestRegressor, GradientBoostingRegressor usw.) oder soll ich doch diskrete Variablen Predicten (also Preiseinordnung Hoch/Mittel/Niedrig) und dann Logistic Regression o.ä.
   2. Weniger Detailliertes Datenset für großere Fehler bei den LLMs?
3. **Sentiment Analysis:**
   1. Beide Variante oder auf eine spezialisieren bzgl. Single vs Multi Term?
   2. Welcher Machine Learning Classifier? —> Habe hier etwas geresearched, aber für Aspect Based Prediction findet sich nur sehr schwer in der nativen Machine Learning / Classifier Richtung etwas und es geht immer schnell in die NLP / LLM Richtung
      1. Implementierung mit Naïve Bayes o.ä. schwierig, da die Sätze vorher immer nach Terms aufgeteilt werden müssen
4. **Schema Matching:**
   1. Was ist hier die richtige Interpretation der Positives / Negatives?
   2. Wie gehe ich hier bezüglich der Nativen ML Variante vor?
5. **Next Steps:**
6. Schema Matching F1 Values
7. Sentiment Analysis / Schema Matching ML Lösung (?)
8. Fehlerklassen ermitteln u. Evaluieren

* Anstatt linear regression lieber Decision Tree probieren – Schauen ob es linear ist oder nicht (ist das extrem viel besser?)
  + Wenn besser: Notebook von Ralph (poly Regression) ausprobieren
* Bei Linear Regression: Polymorph Feature Transformer. 🡪 Feature Space in höhere dimensionen setzen um es linear zu lösen

Sentiment Analysis:

* Kaggle Datasets und Notebooks suchen mit simplen Algorithmen
* Ggf. besseres Datenset finden

Schema Matching:

* Gold Standard als Vergleich nehmen
* Machine Learning Model: Speziell bei LLM fragen ob zwei die gleiche Column sind