Project Report

INNOPOLIS UNIVERSITY

Car price prediction

	Dinislam Gabitov
Students	Andrei Palaev
	Vladimir Bazilevich
Course	MLOps Engineering
Semester	Summer
Year	2024

Contents

1. Introduction

The US market for used cars exceeds $153 billion, while the global market turnover has reached $1.6 trillion [2]. However, a persistent challenge lies in determining fair market value for used cars. Traditional pricing methods, often reliant on subjective judgements by individual sellers or dealerships, can lead to significant inefficiencies. Overpriced listings deter potential buyers, extending selling times and incurring storage costs. Conversely, underpriced cars result in lost revenue for sellers. A substantial portion of used car listings languish on the market due to inaccurate pricing strategies. This inefficiency not only impacts sellers' profits but also hinders a smooth buying experience for consumers.

This project aims to address this challenge by developing a robust Machine Learning (ML) model capable of predicting fair market value for used cars. By leveraging MLOps practices, we can create a reliable and scalable solution that empowers both sellers and buyers in the used car marketplace.

An accurate and objective pricing model empowers both sellers and buyers. Sellers can leverage the model to price their vehicles competitively, facilitating faster sales and maximizing their return on investment. A study by Grand View Research in 2024 suggests that accurately priced vehicles can sell up to 17% faster [1]. Buyers, on the other hand, benefit from increased transparency, avoiding situations where they might overpay for a vehicle.

This project contributes to the growing body of research on applying MLOps principles to real-world business problems. By implementing an automated and efficient ML lifecycle management system, we aim to ensure the continuous development, deployment, and monitoring of our used car pricing model. This will not only guarantee the model's accuracy but also facilitate its adaptation to market fluctuations and evolving consumer preferences. Through the development of this MLOps-driven solution, we hope to contribute to a more efficient and transparent used car market, benefiting both sellers and buyers.

2. Business and Data understanding

This phase focuses on understanding the project objectives and requirements from a business perspective, then converting this knowledge into a machine learning problem. All objectives and constraints must be properly balanced. The goal of this stage of the process is to uncover important factors that could influence the outcome of the project. Neglecting this step can mean that a great deal of effort is put into producing the right answers to the wrong questions.
The used car market suffers from a significant inefficiency due to the prevalence of subjective judgement in pricing vehicles. This practice, documented in industry reports, leads to listings that are either overvalued or undervalued. Overpriced cars languish on the market for extended periods, incurring holding costs for sellers. Conversely, underpriced cars sell quickly but leave significant profit on the table. This lack of objective pricing data hinders both sellers and buyers. Sellers struggle to determine a competitive yet realistic price, while buyers risk overpaying for a vehicle due to the absence of a clear market value benchmark.
In this project, we leverage the Craigslist car sale dataset [5] to develop a machine learning model that predicts a car's fair market value. This model will incorporate various car attributes that influence price, such as mileage, manufacturer, model, and year. By providing a data-driven pricing estimate, our model can empower sellers to optimize their listing prices, facilitating faster sales and maximizing their return on investment. Furthermore, the model can equip buyers with valuable information to avoid overpaying for a used car.

2.1 Terminology

2.1.1 Business terminology

- Fair Market Value (FMV): The estimated price a car would sell for in an open and competitive market, assuming neither buyer nor seller is under pressure.
- Asking Price: The price a seller lists a car for on Craigslist or any other platform.
- Selling Price: The final price at which a car is sold.
- Market Price: The typical price at which similar used cars are selling in the current market.
- MSRP (Manufacturer's Suggested Retail Price): The recommended price set by the manufacturer, not necessarily reflective of market value.
- Market Segmentation: Dividing the car market into smaller groups based on specific characteristics (e.g., luxury cars, fuel-efficient cars).
- Return on Investment (ROI): The net profit gained from selling a car compared to its purchase price.

2.1.2 ML terminology

- Underfitting: When a machine learning model is too simple and fails to capture the underlying patterns in the data.
- Overfitting: When a model performs well on training data but poorly on unseen data. It has memorized specific examples instead of learning general patterns.
- Mean Absolute Error (MAE): metric that measures the difference between values predicted by the model and the actual target values in the dataset. Has the following formula: $MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$, where N - is the number of points in the evaluation set, y_i, $\hat{y}_i$ are the true and predicted values respectively. The closer MAE to 0, the better the model is.
- R^2 - metric that represents the proportion of variance in the target variable that can be explained by the model. Has the following formula: $R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \frac{1}{N}\sum_{i=1}^{N} y_i)^2}$, where N - is the number of points in the evaluation set, y_i, $\hat{y}_i$ are the true and predicted values respectively. The closer the value to 1, the better the model is.

2.2 Scope of the ML project

2.2.1 Background

Craiglist is a company operating a classified advertisements website with sections devoted to jobs, housing, for sale, items wanted, services, community service, gigs, résumés, and discussion forums. Specifically, it contains the information related to car sales.

2.2.2 Business problem

The current process for pricing used cars relies heavily on subjective judgement, leading to listings that are either overpriced and linger on the market or underpriced and leave money on the table for sellers. This sellers want to utilize a variety of car attributes to predict a car's fair market value. This will empower sellers to price their vehicles competitively and realistically, facilitating faster sales and maximizing their return on investment. In addition, it will help buyers avoid overpaying for a vehicle.

2.2.3 Business objectives

- Maximize seller profit and buyer satisfaction by predicting the fair market value of used cars. This will:
    1. Reduce the time cars sit on the market (days listed) for sellers.
    2. Increase the likelihood of getting the asking price for sellers.
    3. Help buyers avoid overpaying for a vehicle.
- Related objectives:

1. How does the accuracy of the predicted price impact the time cars stay listed?
2. Will segmenting the market and offering targeted pricing strategies for different buyer demographics impact the seller profit and buyer satisfaction?
3. Does using the predicted price in conjunction with other pricing strategies (e.g., dynamic pricing) further improve sales velocity and profitability?

2.2.4 ML objectives

The ML objective for this project is to develop a machine learning model that can accurately predict the fair market value of a car, given a dataset of used car listings, including attributes such as mileage, make, model, year, features, and location.

2.3 Success criteria

2.3.1 Business Success Criteria

- Decrease the average number of days a car listing remains active by 15% compared to the current process.
- Achieve a 5% increase in the average selling price for a car compared to the current process. This can be measured by comparing the predicted price to the actual selling price.

2.3.2 ML Success Criteria

- The model should achieve an MAE of less than $3,000 on the car price predictions.
- The models should achieve an R^2 of greater than 0.8 on the car price predictions.

2.3.3 Economic Success Criteria

- Average Time to Sell
  - Measure the average time it takes for cars listed with the help of the model to sell compared to the current process. Significant reduction in listing days will indicate faster sales.
- Seller Profit Increase
  - Measure the average increase in seller profit when using the model prediction compared to the current process.
- Buyer Satisfaction Survey Score
  - Conduct surveys with buyers who used the model pricing suggestion to gauge their satisfaction with the final purchase price.

2.4 Data collection

2.4.1 Data collection report

For the project we are using kaggle dataset with used car price data from craiglist website. The data is collected by kaggle python package. The data have 26 features and 426880 records. Data types of the features are:

- **id** - int
- **url** - string
- **region** - string (categorical)
- **region_url** - string (categorical)
- **price** - int
- **year** - string (date)
- **manufacturer** - string (categorical)
- **model** - string
- **condition** - string (categorical)
- **cylinders** - string (categorical)
- **fuel** - string (categorical)
- **odometer** - int
- **title_status** - (categorical)
- **transmission** - (categorical)
- **VIN** - string
- **drive** - string (categorical)
- **size** - string (categorical)
- **type** - string (categorical)
- **paint_color** - string (categorical)
- **image_url** - string
- **description** - string
- **county** - all nan's
- **state** - string (categorical)
- **lat** - float
- **lon** - float
- **posting_date** - string (date)

2.4.2 Data version control report

We are using DVC for version control with Google Drive as remote storage.

- **Data Version:** The current data version is v1.0, which was updated on June 19, 2024.
- **Data Change Log:** We will maintain a change log that documents any modifications made to the data. This ensures transparency and allows us to replicate the data preparation process for future iterations.
- **Data Backup:** Daily backups of the downloaded data will be stored on a Google Drive. This safeguards against accidental deletion or system failures.
- **Data Archiving:** Since the project duration is too short, none of the data will be archived.
- **Data Access Control:** Access to the data will be restricted only to the project team and Teacher Assistant. This mitigates the risk of unauthorized modifications.

2.5 Data quality verification

2.5.1 Data description

Data consists of a single CSV table with 426880 rows and 26 columns. The identities of the fields are:

- **id** - id of the record
- **url** - link to the post on craiglist website
- **region** - location of the car
- **region_url** - link to craiglist regional website
- **price** - price in usd
- **year** - year when car was produced
- **manufacturer** - car manufacturer
- **model** - model of the car
- **condition** - car condition
- **cylinders** - number of cylinders in car
- **fuel** - fuel type
- **odometer** - total distance travelled
- **title_status** - vehicle title status
- **transmission** - transmission type
- **VIN** - Vehicle Identification Number
- **drive** - drive type

- **size** - car size
- **type** - car type
- **paint_color** - car color
- **image_url** - link to the car image
- **description** - text description of the seller
- **county** - all nan's, do not use
- **state** - US state
- **lat** - latitude
- **lon** - longitude
- **posting_date** - when ad was posted

2.5.2 Data exploration

We found that

- Odometer correlates with the target feature with coefficient ∼0.3 (Figure.1)
- All other features are weakly correlated with the target (Figure.1)
- County contains only NaN values. Hence, we will drop it since it has no information to recover or utilize.
- As year increases, the deviation in price also increases
- Condition, title_status, odometer - better condition (or less travelled distance) leads to increased price for the same car model
- Car defining features: transmission, size, paint color, type, fuel, drive - car price may vary based on the car equipment
- Manufacturer - some of the manufacturers are producing only luxury cars, affecting the price

2.5.3 Data requirements

The data requirements for this project are defined as follows:

- id should be unique.
- price should not be NaN and should be greater than or equal to 0.
- region should not be NaN.
- VIN (Vehicle Identification Number) should contain exactly 17 characters.
- cylinders should be greater than or equal to 0.
- odometer (i.e., number of miles traveled by vehicle) should be greater than or equal to 0.

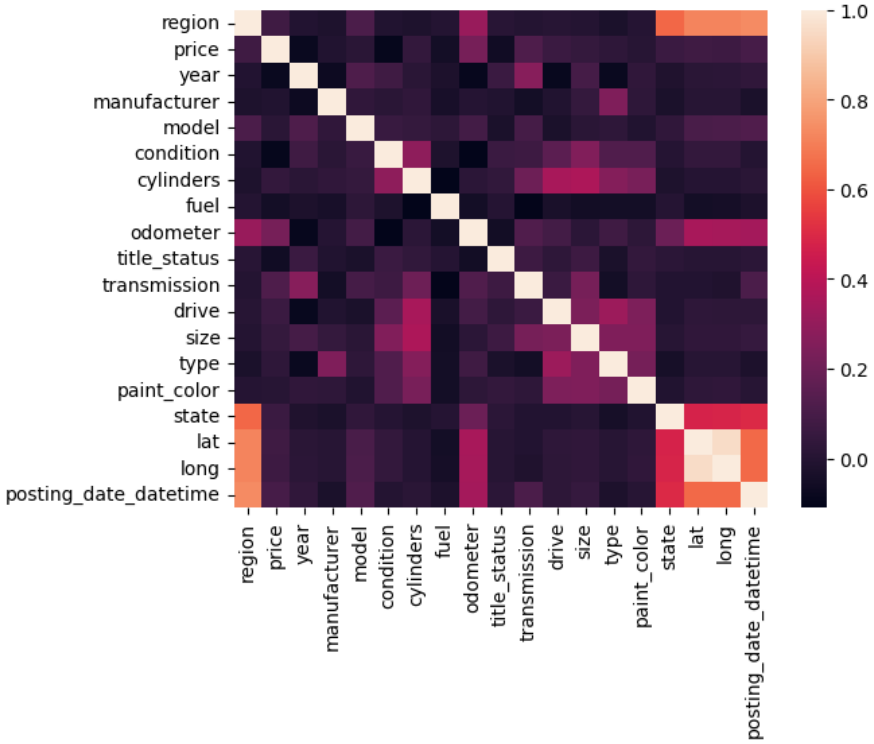

Figure 1: Correlation matrix

- year should be greater than or equal to 1800.
- lat should be in the range $[-90, 90]$, long - $[-180, 180]$.
- drive must be one of [4wd, fwd, rwd].
- transmission must be one of [manual, automatic].
- posting_date must satisfy the format "%Y-%m-%dT%H:%M:%S.%fZ", where Y - year, m - month, d - day, H - hour, M - minutes, S - seconds, Z - milliseconds.

2.5.4 Data quality verification report

- Completeness: The data is complete in the sense that it covers all the required cases. Cars with the different years of manufacture, cars in the different condition, cars with different transmissions, size, colors, fuel types, drive, cars of different types and models are present in the dataset.
- Correctness: The data appears to be correct, with no errors.
- Missing Values: There are missing values in some columns which require different ways of handling.

Overall, the data quality is high, and the data is suitable for analysis and modelings.

2.6 Project feasibility

2.6.1 Inventory of resources

1. Personnel:
    - Data and ML experts:
        - Data Engineer - Responsible for data acquisition, cleaning, transformation, and preparation for modeling.
        - Data Scientist - Responsible for exploring data, feature engineering, model selection training, evaluation, and interpretation.
        - ML Engineer - Responsible for designing, building, deploying, and monitoring the machine learning model.
    - Business experts:
        - Actually, we do not have contacts with someone who has enough expertise in the field of car selling, but we can contact the professor and the TA in case of difficulties.
    - Technical support
        - We can contact the professor and the TA in case of difficulties.
2. Data:
    - Fixed extract:
        - The provided dataset from Kaggle: https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data
3. Computing resources
    - Since we will utilize the data by 20% samples and not the whole data by once, our own computing resources will enough to complete the project. Otherwise, we can use Kaggle or Google Colab to get additional resources, especially in terms of GPU.
4. Software:
    - Machine learning tools:
        - Python libraries such as scikit-learn, PyTorch and skorch
        - MLFlow for model managing
        - Giskard for model validation
    - Other relevant software:
        - Pandas, Matplotlib, Seaborn for Exploratory Data Analysis
        - DVC for data versioning
        - Hydra for configuration file

  - Pytest for testing the code
  - Great Expectations for validating and documenting the data
  - Airflow, ZenML for creating workflows

2.6.2 Requirements, assumptions, and constraints

1. Requirements
   - Data requirements:
     - Accessibility: the dataset is publicly available on Kaggle, so we are allowed to use it without restrictions.
   - Time requirements
     - The project must be completed by the end of the semester (approximately the end of July)
   - Model requirements:
     - At least one of the trained models must give RMSE and R^2 satisfying success criteria on unseen data
   - Technical requirements:
     - The project must satisfy the minimal requirements stated by the Teaching Assistant
2. Assumptions
   - Data relevance
     - The dataset is assumed to be representative of the used car market.
   - Data accuracy
     - There are no mistakes in the dataset.
   - Relationship
     - There is a relationship between the car's attributes and the price.
3. Constraints
   - Time constraint
     - The project must be completed by the end of the semester as stated above.
   - Resource constraint
     - Since no additional resources has been provided, we must use our own resources, which may limit our project in some aspects such as training complex ML models.

2.6.3 Risks and contingencies

1. Inaccurate or incomplete data: The data from Craigslist may contain errors, typos, or missing information about the cars. This can lead to a poorly trained model that produces inaccurate predictions.
    - Contingency Plan: Implement data cleaning techniques to identify and fix errors or inconsistencies in the data. This may involve removing or imputing entries with missing values or removing outliers.
2. Model underfitting or overfitting: The model might not learn the underlying patterns in the data effectively, leading to poor predictions. On the other hand, the model might memorize the training data too well and fail to generalize to unseen data, leading to inaccurate predictions on new cars.
    - Contingency Plan: experiment with different model hyperparameters to find the best configuration that balances underfitting and overfitting. Apply techniques such regularization or dropout layers to prevent the model from overfitting. Train multiple models on different subsets of the data and combine their predictions to potentially improve overall accuracy.
3. Market fluctuations: The used car market can fluctuate significantly due to economic factors, fuel prices, or new car releases. This can affect the accuracy of the model's predictions over time.
    - Contingency Plan: Regularly retrain the model with new data to account for market changes and ensure its predictions remain relevant.

2.6.4 Costs and benefits

Proposed alternative: Develop a machine learning model to predict the fair market value of used cars based on Craigslist car listings data.

Benefits:

1. Increased Sales Velocity (Benefit Impact: 3): Accurate pricing will lead to listings attracting qualified buyers faster, reducing the time a car sits on the market. This translates to faster revenue generation.
2. Improved Seller Profitability (Benefit Impact: 2): Fair market value pricing ensures sellers get a competitive price without undercutting themselves.
3. Enhanced Buyer Experience (Benefit Impact: 2): Buyers avoid overpaying for vehicles and can find cars within their budget more efficiently.

Costs:

1. Data Acquisition (Cost Impact: 1): The Craigslist dataset is publicly available, minimizing data acquisition costs.
2. Model Development and Training (Cost Impact: 2): This requires time from all the team to develop, train, and maintain the model.
3. Deployment and Integration (Cost Impact: 1): model deployment might require development effort.

4. Model Maintenance (Cost Impact: 1): Cost of ongoing monitoring and retraining the model to ensure accuracy over time.

Ratio Benefits/Costs: Assuming that this ratio is calculated as sum(benefits impact)/sum(costs impact), then this ratio is 7/5, meaning that benefits significantly outweight the costs.

Ranking: This project ranks highly due to the potential for significant benefits (increased sales velocity, maximized seller profit and enhanced buyer epxperience) with relatively moderate costs (data acquisition, model development, deployment, maintenance).

2.6.5 Feasibility report

Based on our preliminary ML modelling experiments we find project to be feasible. Based on a prior experience we believe that even in case of unexpected difficulties we will manage to deliver a product that solves the formulated problem and satisfies success criteria stated.

2.7 Project plan

1. **Business and Data Understanding (13.06-25.06):**
    - *Description:* Elicit the project requirements and formulate business problem. Specify business and ML goals. Determine the success criteria for business and ML modeling. Analyse the risks and set mitigation approaches. Explore and analyze the dataset.
    - *Resources:* Domain experts, students, data exploration tools.
    - *Outputs:* Business problem, project requirements, business and ML goals, success criteria, risks and mitigation approaches, explored and analyzed dataset.
    - *Risks:* Inaccurate understanding of business needs, data quality issues (missing values, inconsistencies).
    - *Actions:* Refine problem definition through discussions with stakeholders, data cleaning and pre-processing.
2. **Data preparation (26.06-09.07):**
    - *Description:* Perform data transformation, check the quality of the data and perform data cleaning, create ML-ready features.
    - *Resources:* Students, data manipulation libraries.
    - *Inputs:* Explored dataset.
    - *Outputs:* Tranformed and cleaned dataset with ML-ready features.
    - *Dependencies:* Business and Data Understanding
    - *Risks:* May be delayed due to extensive data cleaning.

- *Actions:* Prioritize critical data cleaning tasks based on impact on modeling.

3. **Model Engineering (10.07-21.07):**
    - *Description:* Select and build ML models, optimize models and select best models.
    - *Resources:* Students, ML and DL libraries.
    - *Inputs:* Tranformed and cleaned dataset with ML-ready features.
    - *Outputs:* Trained ML models.
    - *Dependencies:* Data Preparation.
    - *Risks:* Bad model performance.
    - *Actions:* Experiment with different algorithms and hyperparameter tuning.
4. **Model Validation (21.07-21.07):**
    - *Description:* Prepare one model for production, check the success criteria of ML, check the business and ML modeling objectives, check the quality of the model for production, select one model to be deployed.
    - *Resources:* Students, evaluation metrics.
    - *Inputs:* Trained ML models, evaluation data.
    - *Outputs:* Model chosen for deployment.
    - *Dependencies:* Model Engineering.
    - *Risks:* Bad model performance on evaluation dataset.
    - *Actions:* Experiment with different algorithms and hyperparameter tuning.
5. **Model Deployment (21.07-25.07):**
    - *Description:* Deploy the model.
    - *Resources:* Students, deployment tools.
    - *Inputs:* Model chosen for deployment.
    - *Outputs:* Deployed model.
    - *Dependencies:* Model Validation.
    - *Risks:* Limited time for deployment.
    - *Actions:* Focus on documenting a basic deployment plan for future implementation.
6. **Project Presentation (24.07-25.07):**
    - *Description:* Finalize all the project deliverables, present the final project.
    - *Resources:* Students.
    - *Inputs:* Project results.

- *Outputs:* Final project report and presentation.
- *Dependencies:* All prior phases.
- *Risks:* Presentation shortcomings.
- *Actions:* Practice and refine presentation beforehand.

3. Data Preparation

The data preparation phase covers all activities to construct the final dataset (data fed into the machine learning pipelines) from the initial raw data. Data preparation tasks are likely to be performed multiple times and not in any prescribed order. Tasks include table, record and attribute selection as well as transformation and data cleaning for the modelling phase.

3.1 Select data

We dropped the columns that contain more than 20% of NaNs values. These columns are:

1. 'condition'
2. 'cylinders'
3. 'drive'
4. 'size'
5. 'type'
6. 'paint_color'
7. 'county'

In addition, we have dropped the following columns:

1. 'image_url' - to properly preprocess the images, we will need complex deep neural networks which will require a lot of resources and add significant computational overhead
2. 'description' - similar to 'image_url'
3. 'posting_date' - created a new column 'posting_date_datetime', which is 'posting_date' converted in datetime type, and transformed it further
4. 'id' - ID does not contain any valuable information for predicting car price since its value is unique for any entry
5. 'url' - similar to 'id'
6. 'region_url' - contains only the information that is shown in the 'region' column
7. 'lat' and 'long' - new features were derived from them using periodic transform
8. 'VIN' - 'WMI' and 'VDS' are derived from it

For the rows, we have kept only the rows with the price between $1,000 and $40,000 since the typical car price lies in the range. If the price does not belong to this range, it is likely to be an outlier and can skew the model's understanding of the relationship between features and the target variable (price). In addition, we have dropped all the rows which have NaN value in the 'VIN' column. This is because 'VIN' is an important feature for one of our models as will be shown in Chapter 4.

3.2 Clean data

The following columns were imputed with the most frequent value in the column:

1. 'manufacturer'
2. 'model'
3. 'fuel'
4. 'title_status'
5. 'transmission'
6. 'state'
7. 'region'

These are categorical columns representing distinct choices. Imputing with the most frequent value fills the missing value with the most common option, which is a reasonable assumption for these types of features.
The 'year' column was imputed with the median value. Year is a discrete variable. Since it is not continuous data, using the mean is an option here. The median represents the "middle" year in the dataset, offering a more robust estimate for missing values compared to the mean which can be skewed by outliers. The following columns were imputed with the mean value in the column:

1. 'odometer'
2. 'lat'
3. 'long'
4. 'posting_date'

The first three columns represent continuous numerical data. In such cases, the mean provides a central tendency of the data and can be a reasonable estimate for missing values, when the data is not heavily skewed.
Finally, we have imputed the 'year' column with the median value. The reason is that it has an integer value, meaning that mean strategy can not be applied. We chose the median strategy since the median is robust to outliers and skewed distributions. The median maintains the natural spread and central tendency of the data, avoiding the distortion that can be introduced by the mode. In contrast, the mode strategy would over-represent the most frequent value, which could distort analysis and modelling results. We have not faced any issues during data quality verification in the previous phase. Hence, no additional data cleaning was needed.

3.3 Construct data

The 'posting_date_datetime' column was split into 'posting_date_month' and 'posting_date_day' which correspond to the month and day of the posting date. Year is not extracted here because posting_date lies in a range of few months. Then, we applied periodic encoding with sine and cosine to these features. Encoding month and day of posting with sine and cosine can capture seasonal trends in pricing.

'VIN' was split into two features: 'WMI' which corresponds to World Manufacturer Identifier (WMI) and is extracted as the first three characters of VIN, and 'VDS' which corresponds to Vehicle Descriptor Section (VDS) and is extracted from the fourth to the eighth characters of VIN.
The following features were encoded with one-hot encoding:

1. 'title_status'
2. 'transmission'
3. 'fuel'
4. 'state'
5. 'manufacturer'

The reason is that these are categorical features with no inherent order (e.g., "automatic" vs "manual" transmission, "gas" vs "electric" fuel). One-hot encoding avoids introducing false ordinal relationships and creates separate binary features for each category, allowing the model to learn their independent effects on price.
The following columns were encoded with label encoding:

1. 'WMI'
2. 'VDS'
3. 'region'
4. 'model'

The first reason for that is explained in Chapter 4. Secondly, while these are categorical features, they have too many categories for one-hot encoding which can significantly increase the number of features. Label encoding assigns a numerical value to each category, keeping the feature space more manageable.
Finally, 'lat' and 'long' were encoded with periodic encoding with sine and cosine. Latitude and longitude represent locations on a sphere, and directly using them might not capture the cyclical nature of geographic influences on price (e.g., coastal areas might be more expensive). Periodic encoding with sine and cosine transforms these values into features that represent their position on a circular system, potentially allowing the model to learn these cyclical relationships with price.

3.4 Standardize data

The 'year' column was scaled with a MinMax scaler. The reason is that it normalizes the range of the 'year' values, ensuring they contribute uniformly to the machine learning model. This scaling helps maintain consistency across features, improves algorithm performance, and enhances interpretability, especially for models sensitive to the scale of input features. In addition, Standard scaling was applied to the 'odometer' column. It is a continuous numerical feature representing mileage. Applying standard scaling to it will equalize its influence compared to other features and stabilize the convergence of ML models. Standard scaling is more resistant to outliers compared to MinMax scaling. In contrast to Robust scaling which requires sorting the dataset to find the median, standard scaling has a linear computational complexity since it simply requires

a mean and standard deviation to be found.
After all the sites described above, we load and version the features and the target in the artifact store of ZenML.

4. Model engineering

The choice of modelling techniques depends on the ML and the business objectives, the data and the boundary conditions of the project the ML application is contributing to. The requirements and constraints that have been defined in Chapter 1 are used as inputs to guide the model selection to a subset of appropriate models. The goal of the modelling phase is to craft one or multiple models that satisfy the given constraints and requirements.

4.1 Literature research on similar problems

Our literature review identified relevant research applying ML to used car price prediction.
Samruddhi and Kumar [6] have applied the K-Nearest Neighbour (KNN) algorithm to predict the price of the car. The results have shown that the number of neighbours $k = 4$ gives the best accuracy Root Mean Squared Error (RMSE) rate of 4.73 and Mean Absolute Error (MAE) rate of 2.13, but how the price is defined in the original dataset remains unclear from the paper.
While KNN offers simplicity and potentially good performance, the problem of this project is that the use of classical Machine Learning (ML) algorithms is prohibited and we need to use neural networks. Pillai [4] suggested using the embedding layers for the region, VDS, WMI, and model. Then, they concatenate these 4 embeddings with the rest of the numerical features they use and pass it through a simple Multi-Layer Perceptron (MLP). They compared the results of this neural network to classical ML models such as Random Forest and Decision Trees. The results have shown that MLP achieves significantly better results achieving MAE of 1060.37, MAPE of 0.11, RMSE of 2104.13 and R^2 of 0.9634.
To summarize, the results suggest that a simple neural network may be enough to achieve the success criteria defined in Chapter 2. Building upon these findings, the next step will be to explore and implement various neural network architectures suitable for used car price prediction on our chosen dataset.

4.2 Define quality measures of the model

In this project, we will evaluate our models using a combination of performance metrics, and soft measures. Unfortunately, the time limit of this project does not enable to utilize the business and economic success criteria as it requires using selected models in the real-world business which is out of the project's scope.
For performance metrics, we will utilize MAE and R^2 as defined in ML success criteria in Chapter 2. MAE was chosen because it enables to measure the average difference between predicted and actual car prices, crucial for setting realistic car prices. R^2 was selected to evaluate how well the model explains the variability in car prices, ensuring the model captures essential patterns.
In addition to these performance metrics, our modelling strategy will consider several soft measures: robustness, explainability, scalability, resource demand, and model complexity. Robustness ensures the model is resilient to outliers and noise in the data to maintain accuracy in real-world scenarios with imperfect data. Explainability

was chosen since understanding how the model arrives at its predictions is crucial for building trust in its outputs. This is particularly important for sellers who need to justify pricing decisions. Scalability assesses the model's performance as data volume increases. Resource demand evaluates the computational resources required for training and inference. Model complexity balances performance with maintenance ease and interpretability.
Using these quality measures, we can rank models by summing weighted measures or identifying Pareto optimal models, ensuring a balance between accuracy and practical requirements such as fairness and trust. This approach will help us develop a robust, scalable, and explainable model that meets our success criteria, ultimately improving the pricing strategy for used cars.

4.3 Model Selection

In our project, we utilized two primary models to build the ML system: a multi-layer perceptron (MLP) consisting of four linear layers. After each ReLU activation function is applied except the last layer. The architecture of this model can be seen in Fig. 2. Another model is the same as the one used in [4] except that after concatenation we use only four linear layers with ReLU activations similar to MLP. The architecture of this model can be seen in Fig. 3. These models were chosen to compare and understand their characteristics in line with the No Free Lunch Theorem, which suggests no single model is universally best for all problems. Starting with lower-capacity models like the MLP serves as a baseline, while gradually increasing complexity with the use of Embedding layers allows validation at each step, ensuring benefits without unnecessary complexity.
The input dimension for both models consists of 120 features, corresponding to various car attributes. The output dimension is a single value representing the predicted car price. This approach ensures we balance performance and complexity, allowing for an effective, scalable solution to predict used car prices.

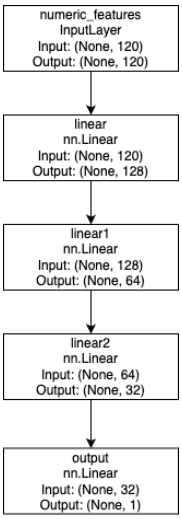

Figure 2: The architecture of MLP

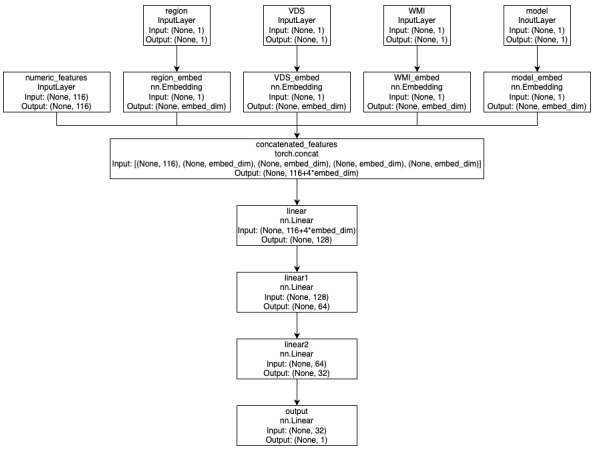

Figure 3: The architecture of Embedder model

4.4 Incorporate domain knowledge

To ensure the selection of quality metrics and models aligns with our business problem of predicting used car prices, we chose MAE and R^2 as our primary performance metrics. These performance metrics directly relate to the business goal of accurately predicting car prices. A low MAE translates to minimal deviations between predicted and actual prices, and a high R^2 signifies a strong correlation between the model's predictions and the market value.

The selected models, MLP and Embdder, offer a balance of simplicity and complexity, making them suitable for our task while allowing for stepwise validation and optimization. This method ensures that the added domain knowledge enhances model performance, staying true to our business goals of accurate and fair car pricing.

The 120 input features represent car attributes such as mileage, make, model, year, and region. This domain knowledge ensures the model considers factors that significantly influence car prices. The MLP acts as a baseline to compare against the more complex Embedder architecture. If the Embedder does not significantly outperform the MLP while considering explainability and resource demands, it might indicate that the added complexity does not justify the gains.

To summarize, our approach leverages domain knowledge while acknowledging the potential downsides of incorporating false assumptions. By employing a baseline model and carefully evaluating the impact of model complexity, we can ensure the selected models and quality metrics are well-aligned with the business objective of achieving accurate used car price predictions.

4.5 Model training

The training process for our models is critically dependent on the learning problem, encompassing the objective, optimizer, regularization, and cross-validation. Each of these components is meticulously aligned with our business success criteria to ensure optimal model performance in real-world scenarios.

The objective of our learning problem is to minimize the Mean Absolute Error (MAE) and maximize the R^2 score, metrics directly tied to our business goals of accurate and explainable car price predictions. MAE measures the average discrepancy between

predicted and actual prices, while R^2 indicates the proportion of variance explained by the model.
We employ the Adam optimizer [3] due to its adaptive learning rate capabilities, which efficiently handle the sparse gradients and noisy data typical of used car price prediction tasks. To prevent overfitting, weight decay (L2 regularization) is incorporated into the optimizer, penalizing large weights and encouraging simpler models that generalize better.
For robust model evaluation, we perform 3-fold cross-validation. This method splits the dataset into three parts, using two parts for training and one for validation in each fold, providing a comprehensive estimate of the model's generalization ability. To ensure thorough evaluation, we sorted the dataset based on 'posting_date' and split it into 5 samples. The first sample was used to fit the model using cross-validation, and the second was used as the testing dataset. In such a way, we can properly evaluate whether the model can generalize well to the whole distribution.
After performing grid searches with cross-validation, we evaluated the models using the test set. For the MLP model, the best hyperparameters were max_epochs=50, lr=0.001, and weight_decay=100.0, yielding an MAE of $5206.02 and R^2 of 0.5721. The Embedder model, with max_epochs=50, weight_decay=100.0, and embed_dim=20, achieved an MAE of $2639.07 and an R^2 of 0.8536. These results indicate that the Embedder model outperforms the MLP in both MAE and R^2, demonstrating the effectiveness of embedding layers for categorical features.

4.6 Assure Reproducibility

Reproducibility is a cornerstone of scientific research and a critical aspect of robust machine learning applications. We focus on two levels of reproducibility: method reproducibility and result reproducibility.
To ensure method reproducibility, we documented every aspect of our modelling process in detail. The algorithms, including the architecture of the MLP and Embedder models, are described comprehensively. The MLP comprises four linear layers with ReLU activations, except for the final layer. The Embedder model utilizes embedding layers for categorical features, which are then concatenated with numerical features and passed through four linear layers. The dataset used, the procedure for splitting the dataset into training and testing sets as well the procedure to train the model are described above.
The Grid Search space for each model is the following:

- MLP:
    1. max_epochs: [10, 20, 50]
    2. lr: [1e-4, 3e-4, 0.001]
    3. weight_decay: [1.0, 10.0, 100.0]
- Embedder:
    1. max_epochs: [10, 20, 50]
    2. weight_decay: [1.0, 10.0, 100.0]

3. embed_dim: [10, 20, 50]

In addition, we have set the lr parameter for all Embedder models to 0.001 To ensure that specific versions of used Python libraries (e.g., MLFlow, skorch and PyTorch) are installed, we have used the Poetry dependency management tool. Finally, we have set all the random seeds to 88 across all the runs. Due to PyTorch limitations, different runs on the same machine are reproducible but might yield different results on another machine even though all the seeds are fixed.

To validate result reproducibility, we ran multiple experiments with different random seeds to assess performance variance, identifying any model sensitivities and ensuring robustness. The results for MLP can be seen in Table 1, for the Embedder model - in Table 2. All the metrics in this experiment were calculated on the test dataset. The results show that the Embedder model shows better performance on both metrics across all the seeds, having better mean metric value and small variance, indicating better stability in terms of performance.

All the configurations and the code used for the described experiments can be found in our GitHub repository: https://github.com/Palandr123/MLOps-Project.

By adhering to these practices, we ensure our modelling process is transparent, reproducible, and robust, ultimately leading to reliable and trustworthy used car price predictions.

Table 1: Reproducibility experiment results for MLP model

Random seed	MAE	R^2
1	5572.65	0.5097
2	5167.86	0.584
3	5185.79	0.5762
4	5817.22	0.4715
5	5335.63	0.5505
Mean	5415.83	0.5384
Variance	61309.37	0.0018

Table 2: Reproducibility experiment results for Embedder model

Random seed	MAE	R^2
1	2608.87	0.8569
2	2621.78	0.857
3	2625.59	0.8556
4	2625.50	0.8548
5	2616.85	0.8555
Mean	2619.72	0.856
Variance	39.59	7.39e-07

5. Model evaluation

In this chapter, we will assess the extent to which the developed solution aligns with the business criteria. We will identify the approved models based on their performance results and the objectives stated in Chapter 2.

5.1 Model validation report

We have selected the champion model based on the following factors;

1. Passess ML success criteria
2. Giskard validation report gives the minimum number of vulnerabilities

Based on these factors, the Embedder model with embed_dim=50, weight_decay=100.0 and max_epochs=50 was chosen as a champion model. It gave MAE of $2,658.3 and R^2 of 0.852 which passes set ML success criteria.
From the Giskard validation, we got the following vulnerabilities:

- MSE +44.68% than global when WMI = 114
- MSE +44.26% than global when fuel is not 'gas'
- MSE +22.14% than global when the manufacturer is 'ford'
- MSE +14.26% than global when the state is 'ca'
- MSE +10.60% than global when manufacturer is 'chevrolet'
- MSE +6.55% than global when the transmission is not 'automatic'

All these vulnerabilities are the same. Model performance drops when the value of the column is equal to some value.

5.2 Discussion

We have the following success criteria for models:

- $MAE < 3000$
- $R^2 > 0.8$

From the model performance chart, we can see that the best-performing models on the test set also pass validation on Giskard. We also made a Giskard analysis on all models that pass success criteria to choose a champion by the minimal number of major issues.

5.3 Deployment decision

As our model passed the success criteria and successfully completed Giskard validation, we decided to deploy the champion model to production.

6. Deployment

In this section, we describe the deployment of our model.

6.1 Hardware demands

Even though we use neural networks, the models appear to be quite small. For that reason, we do not need the GPU. Further, based on our experiments data transfer overheads will exceed the inference time on the CPU.
We have tested our deployment on a machine with Ryzen 9 5900X and 32 GB RAM. Even this setup is being underutilized. Service does not require more than 1 CPU core and 2GB of RAM unless you aim to serve more than 500 queries per second which seems to be pretty unlikely in a real market.

6.2 Model evaluation under production condition

Based on the evaluation we performed, we find our project to be successful. However, we were unable to perform the online testing to evaluate all of the economic success criteria, especially ones related to reduced sale time and increased profit.
Based on subsection 6.1 we claim that the online testing is very unlikely to require some extra hardware. Yet, extra efforts are required to integrate our API into the application of the end user. Therefore, we do not expect the price of the online testing to exceed 100 thousands of rubles which will be mainly spent on salaries of frontend programmers and DevOps engineers to integrate our microservice.

6.3 Deployment strategy

The majority of car sales happen on online boards so our deployed model has to be easily incorporable into such web services. For that reason, we enable the deployment of our model in two fashions:

1. Docker image that can be deployed on the end user machine and that accepts POST requests for prediction;
2. Flask API that can be deployed locally and provides predictions similarly by POST requests from the end user.

Further, to simplify the sales of our produce we provide a minimalistic Gradio application to interact with the Flask API.

References

[1] Sejal Akre. Used vehicle market size, share, trends report 2030 - industry growth analysis, Jun 2024. URL https://www.marketresearchfuture.com/reports/used-vehicle-market-7616.

[2] Carsurance. 22 automotive industry stats & facts for 2024, January 2024. URL https://carsurance.net/insights/automotive-industry-statistics/.

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[4] Aravind Sasidharan Pillai. A deep learning approach for used car price prediction. *Journal of Science & Technology*, 3(3):31–50, 2022.

[5] Austin Reese. Used cars dataset, 2021. URL https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data.

[6] K Samruddhi and R Ashok Kumar. Used car price prediction using k-nearest neighbor based model. *Int. J. Innov. Res. Appl. Sci. Eng.(IJIRASE)*, 4(3):2020–686, 2020.