Predicting Advertisement Demand on Avito Platform

Artur Zagitov, Ildar Zalialiev, Artem Nazarov

Contents

1 Introduction

In the digital age, the online marketplace has become an essential platform for buying and selling goods. However, sellers often face challenges in predicting the demand for their products, which can lead to frustration and suboptimal pricing strategies. Avito, Russia's largest classified advertisements website, encounters these issues regularly. Sellers are frequently frustrated by the lack of demand for their listings or overwhelmed by unexpected high demand, indicating that the product description or price might not be optimal.

This report presents the first phase of the development of a machine learning system to predict demand for online advertisements. By using comprehensive data, provided by the platform itself, including ad descriptions, geographical context and other relevant information, we aim to provide sellers with insights to optimize their listings and better understand market interest.

2 Business and Data Understanding

2.1 Business Problem

The primary business problem is to accurately predict the demand for online advertisements on Avito. This involves understanding the details of product descriptions and other contextual factors that influence buyer interest. The goal is to provide sellers with insights to enhance their listings and manage expectations regarding demand.

Addressing this problem is crucial for several reasons:

- **Optimizing Listings:** By predicting demand more accurately, sellers can adjust their product descriptions, category and prices to better match market interest.
- **Reducing Frustration:** Sellers often face frustration when their products do not sell or sell too quickly, indicating potential underpricing. Accurate demand predictions can help mitigate these issues by providing realistic expectations.
- **Increasing Revenue:** For Avito, helping sellers optimize their listings can lead to higher transaction volumes, which will translate to increased revenue through commissions and advertising fees.
- **Enhancing User Experience:** Buyers benefit from more relevant and well-described listings, improving their overall shopping experience on the platform and increasing their retention.
- **Competitive Advantage:** In a competitive market, offering advanced tools and insights for sellers can set Avito apart from other classified advertisement websites.

2.2 Assessment of the Current Situation

Avito's platform hosts millions of advertisements across various categories and regions. Despite the platform's extensive reach, many sellers struggle with optimizing their ad listings due to a lack of understanding of what drives demand. This results in either overpricing or underpricing of products, leading to missed opportunities or dissatisfaction.

In data, provided by Avito, Avito computes deal probability based on a combination of how many people view the ad and how many people click the button. As illustrated in Figure 6, the deal probability for the majority of ads is very low, with approximately 66.4% of ads having a deal probability of less than 0.05. This indicates a significant challenge in generating interest and converting views into sales for a large portion of the listings on Avito. The histogram shows a long tail distribution with a few ads achieving high deal probabilities, suggesting that while some ads are highly effective, most struggle to attract buyers.

2.3 Terminology

2.3.1 Business Terminology

- **Ad Listing:** A product advertisement posted by a seller on Avito.

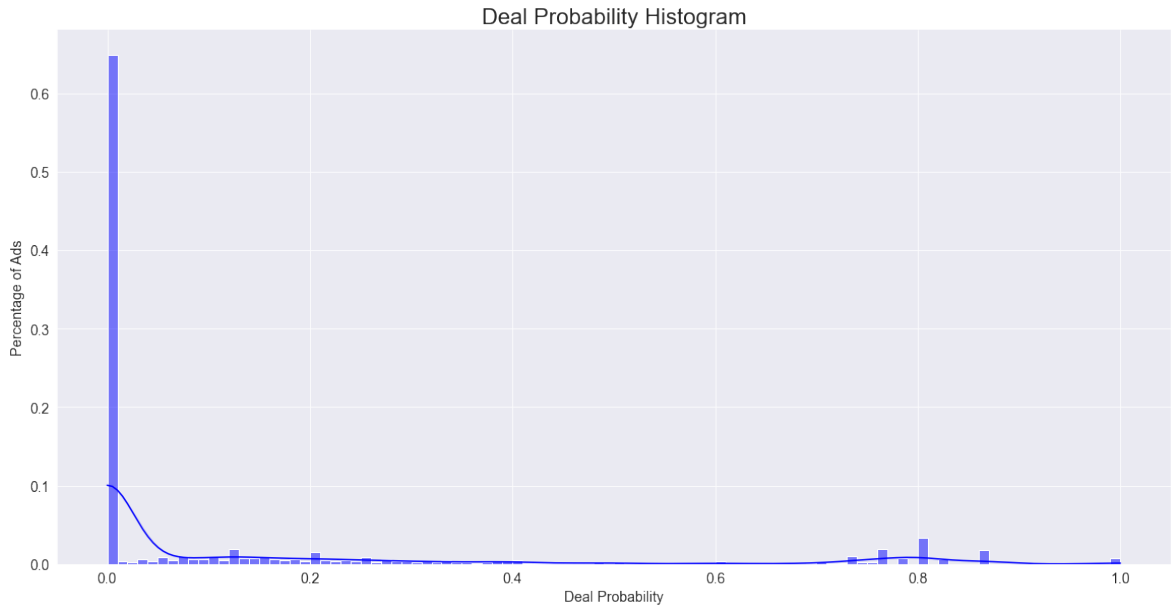

Figure 1: Deal Probability Histogram

- **Demand:** The interest shown by potential buyers, measured by views and interactions with the ad.
- **Deal Probability:** The likelihood that an advertisement will result in a transaction.

2.3.2 ML Terminology

- **Root Mean Squared Error (RMSE):** A measure of the differences between predicted and observed values. It is used to evaluate the accuracy of the machine learning model.
- **Feature:** An individual measurable property or characteristic of a phenomenon being observed.
- **Categorical Data:** Variables that contain label values rather than numeric values.
- **Text Data:** Unstructured data that contains words and sentences.
- **Regression:** A type of predictive modeling technique which estimates the relationships among variables.
- **Feature Engineering:** The process of using domain knowledge to create features that make machine learning algorithms work better.
- **Overfitting:** A modeling error which occurs when a function is too closely aligned to a limited set of data points.
- **Cross-Validation:** A technique for evaluating ML models by training several models on subsets of the available input data and evaluating them on the complementary subset of the data.

2.4 Scope of the ML Project

2.4.1 Background

Avito is the largest classified advertisements website in Russia, providing a platform for users to buy and sell a wide range of products and services. Founded in 2007, Avito has grown rapidly and now has millions of active listings and a large user base across the country. The platform handles vast volumes of data daily, including user interactions, ad postings, and transaction records, click streams, etc. Avito operates in multiple regions, to a diverse audience with varying needs and preferences and allows to list advertisements for many different types of products and services.

Being Russia's largest e-commerce platform, Avito faces the challenge of maintaining high-quality listings and user satisfaction. Due to the large volume of data processed by the platform, a machine learning and MLOps solutions are required to handle data ingestion, processing, model training, and deployment efficiently.

2.4.2 Business Problem

To provide sellers with accurate predictions of demand for their advertisements based on ad descriptions, contextual information, and historical data.

2.4.3 Business Objectives

- Increase the efficiency of ad listings by providing sellers with predictive insights.
- Reduce the number of underpriced and overpriced ads by optimizing pricing strategies.
- Enhance user satisfaction.

2.4.4 ML Objectives

- Develop a machine learning model to predict deal probability based on ad features and historical data.
- Achieve an RMSE of 0.25 or lower.

2.5 Success Criteria

2.5.1 Business Success Criteria

The success of the ML application will be measured by its ability to increase the number of successfully completed transactions on Avito by providing more accurate demand predictions.

2.5.2 ML Success Criteria

The machine learning model must achieve an RMSE of 0.25 or lower on the validation dataset, indicating that the predictions are sufficiently accurate for practical use. We choose this target based on the best performance ever achieved for this task, which has RMSE of about 0.22. However, we not only do not use image information, but also we use only 10,000 samples instead of 1.5m, due to limited resources and time. That is why we set our target a bit higher then the best achievable RMSE.

2.5.3 Economic Success Criteria

A key performance indicator (KPI) will be the increase in revenue generated from ad listings due to improved pricing strategies and higher demand.

2.6 Data Collection

2.6.1 Data Collection Report

The data used in this project is sourced from Avito's classified advertisements platform, which is available on Kaggle. The dataset initially consisted of 1.5 million rows, but was undersampled to 10,000 rows. Dataset containts 17 columns, including features such as ad titles, descriptions, prices, and geographical information. The data is collected in CSV format and includes a mix of date, categorical, numerical, and text data types.

To assist with the data collection process, we use the Kaggle API. This allows us to programmatically download the dataset directly from Kaggle.

Once the data is downloaded, we manage its versioning using Data Version Control (DVC). DVC is an open-source tool that enables versioning of data and machine learning models. It allows us to track changes to the dataset over time, ensuring that we can reproduce experiments and maintain a history of the data.

2.6.2 Data Version Control Report

Data versions are managed using Data Version Control (DVC), which integrates with our Git version control system. DVC allows us to version control large datasets and machine learning models. Currently, there are 2 versions of the data: raw data with all 1.5 million rows, and undersampled data with 10,000 rows.

2.7 Data Quality Verification

2.7.1 Data Description

The dataset includes the following 17 columns and contains 10,000 rows. Table 1 presents the information about each feature.

Column Name	**Description**	**Data Type**	**Feature Type**
item_id	Ad id.	object	Categorical
user_id	User id.	object	Categorical
region	Ad region.	object	Categorical
city	Ad city.	object	Categorical
parent_category_name	Top level ad category as classified by Avito's ad model.	object	Categorical
category_name	Fine grain ad category as classified by Avito's ad model.	object	Categorical
param_1	Optional parameter from Avito's ad model.	object	Categorical
param_2	Optional parameter from Avito's ad model.	object	Categorical
param_3	Optional parameter from Avito's ad model.	object	Categorical
title	Ad title.	object	Text
description	Ad description.	object	Text
price	Ad price.	float64	Numerical
item_seq_number	Ad sequential number for user.	int64	Numerical
activation_date	Date ad was placed.	date	Datetime
user_type	User type.	object	Categorical
image	Id code of image.	object	Categorical
image_top_1	Avito's classification code for the image.	int	Categorical
deal_probability	The target variable. This is the likelihood that an ad actually sold something. It can be any float from zero to one.	float64	Numerical

Table 1: Dataset Description

2.8 Data Exploration

In the data exploration stage, we aim to understand the underlying structure of the dataset, identify initial trends and patterns, and formulate hypotheses that can guide further analysis and model development. Here we present the results of our exploration, highlighting the most insightful findings.

2.8.1 Exploratory Data Analysis

Parent Category Name and Category Name

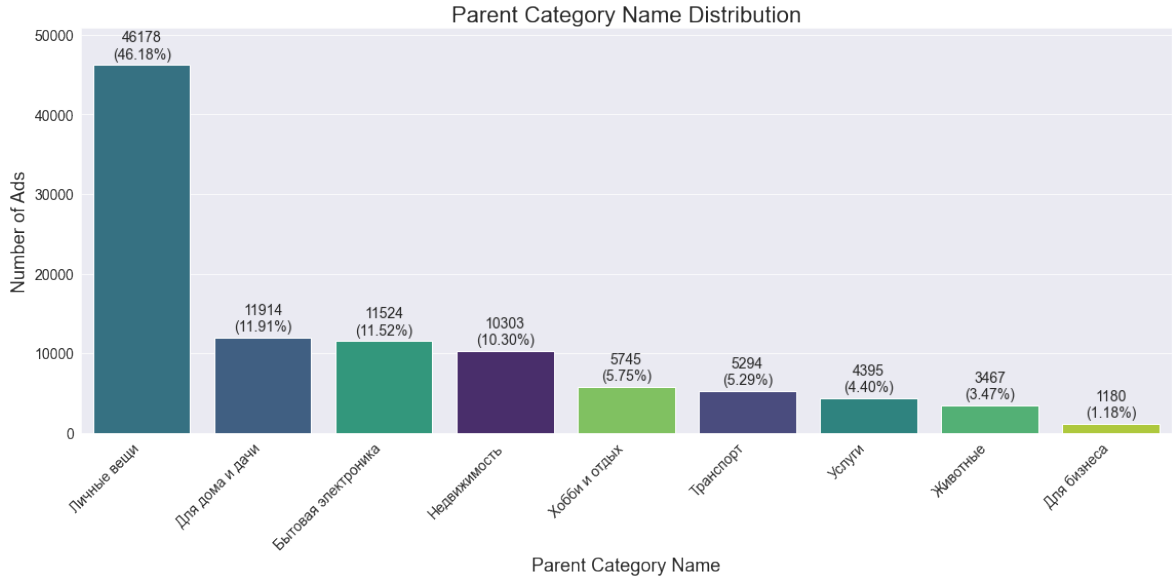

Figure 2: Parent Category Name Distribution

The distribution of parent category names shows that the most common parent category is "Personal belongings," followed by "For the home and garden" and "Consumer electronics." The distribution of ads across parent categories is not uniform, with some categories having significantly more ads than others, especially "Personal belongings", which contains almost half of the ads in the dataset. This indicates a heavy skew towards personal items in the dataset, which may impact the demand predictions.

There are also 47 unique categories in the dataset. The most common category is "Clothing, shoes, and accessories," followed by "Children's clothing and shoes". The distribution of ads across categories is not uniform. The first two categories, related to clothing, have significantly more ads than others, while 36 categories have less than 1% presence in the dataset.

User Type

The majority of ads are posted by private users, followed by companies and shops. Private users dominate the dataset, which might suggest different behaviors and demand patterns compared to commercial users.

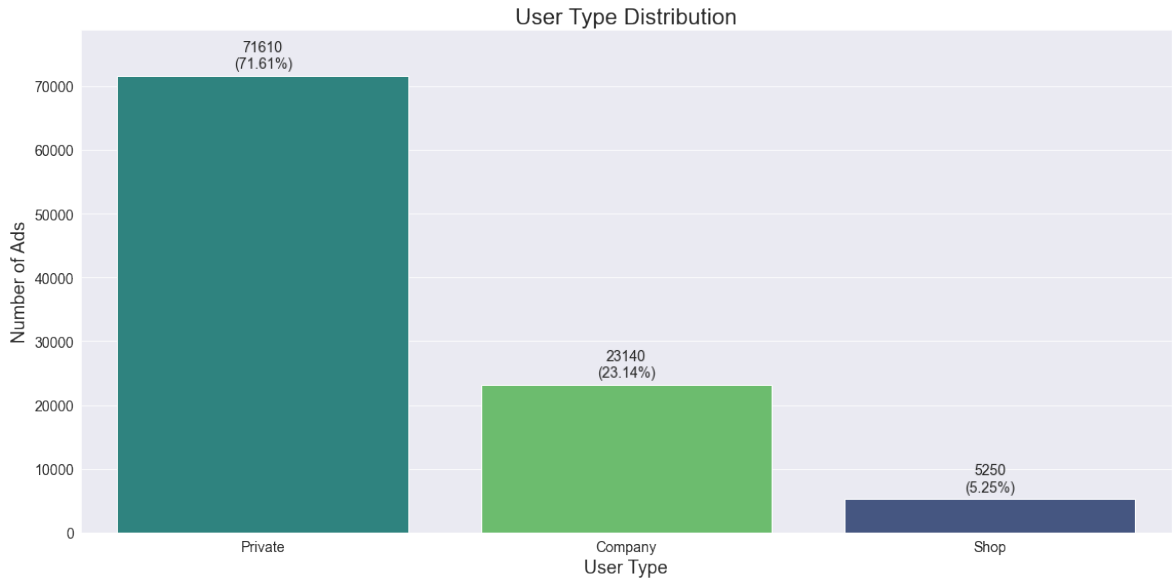

Figure 3: User Type Distribution

Region

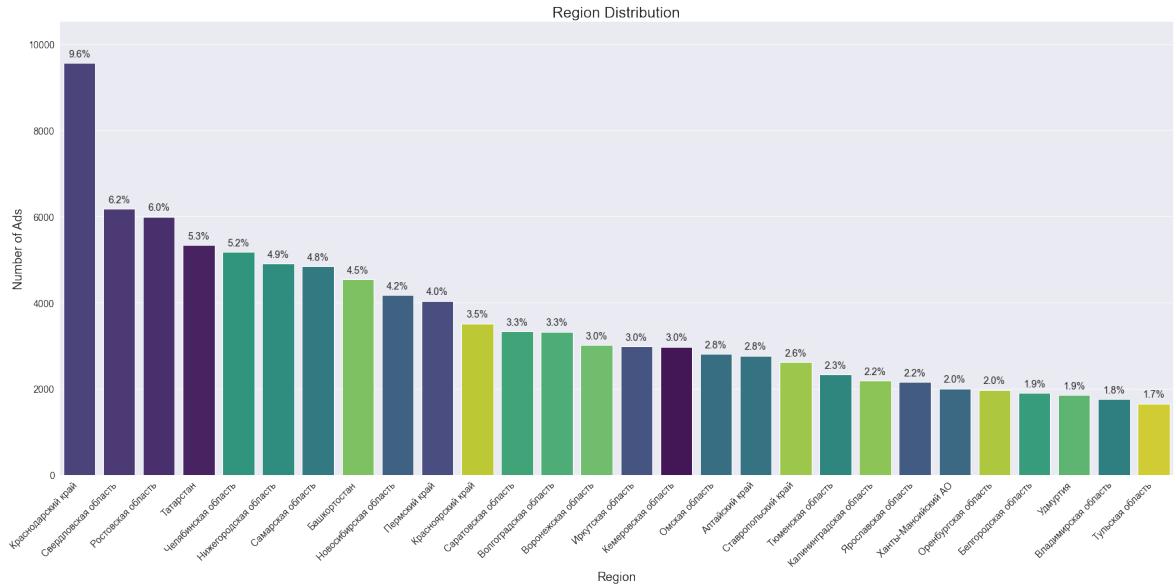

Figure 4: Region Distribution

The most common region is "Krasnodar region", followed by "Sverdlovsk region" and "Rostov region". The distribution of ads across regions is not uniform, with some regions having significantly more ads than others, especially "Krasnodar region", however, the difference is not very significant.

Price

Figure 5: Log_{10} of Price Histogram

The price distribution is right-skewed with a peak around 100 to 10,000 rubles. There are significant outliers with very high prices, which may need to be handled separately in the modeling phase to prevent skewing the predictions.

Deal Probability

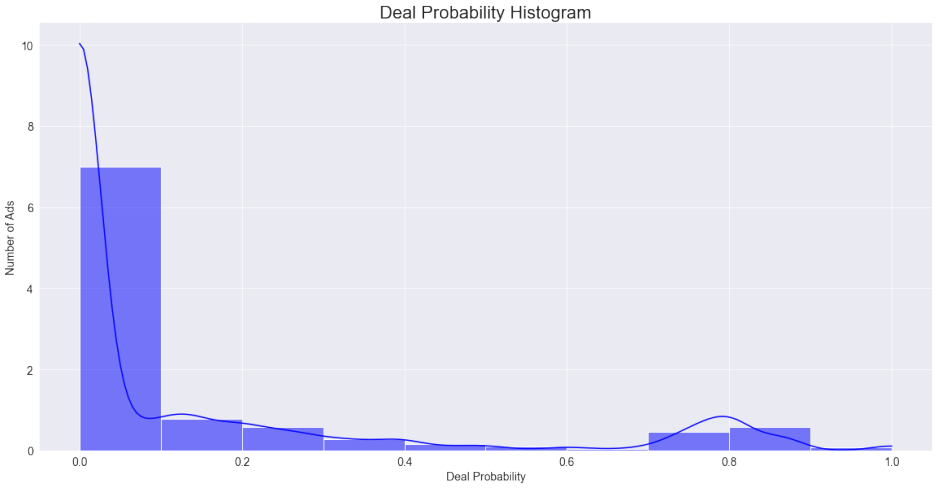

Figure 6: Deal Probability Histogram

The main target - deal probability, is calculated as a ratio between the total number of deals and the number of users who click on the ad, and has values from 0 to 1.

The deal probability distribution is heavily right-skewed, with approximately 66.4% of ads having a deal probability of less than 0.05. This indicates that most ads struggle to convert views into transactions. There is also a small peak around 0.8, indicating a smaller number of ads with higher deal probabilities.

If we consider deal probability as a binary feature, where 1 means that the ad has a deal probability greater than 0.5, and 0 otherwise, the difference becomes clearer.

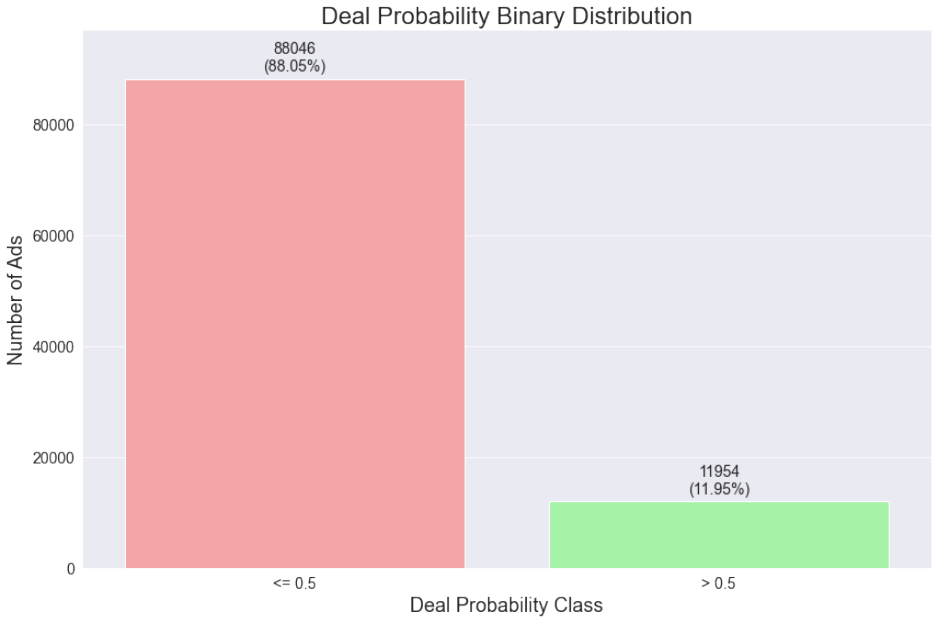

Figure 7: Deal Class Distribution

Title length

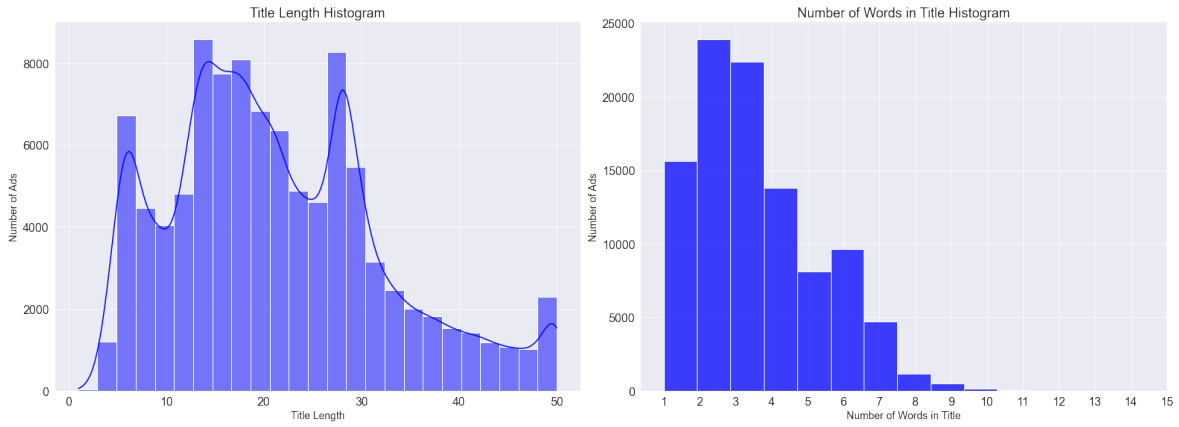

Figure 8: Title length Distribution

The title length histogram shows that the titles have a length between 0 and 50 characters, and 50 seems to be the character limit for titles. Majority of titles have a length between 0 and 30 characters and after that, the number of ads decreases. The number of words in the title histogram shows that the titles have between 1 and 15 words, with most titles having 1 to 10 words, and the majority having 1 to 5 words.

Price vs. Deal Probability

Figure 9: Price vs Deal Probability

- There is a high density of ads with lower prices (left side of the plot).
- Many of these low-priced ads have varying deal probabilities
- A few ads with low prices have a deal probability close to 1
- There is a line of ads with around 0.78 deal probability that have prices between 0 and 3,000,000, which could indicate a specific category of ads or a specific users group.
- As the price increases, the density of ads decreases, and the deal probability tends to be lower.
- There are a few outliers where some high-priced ads have achieved moderate deal probabilities.

User Type and Parent Category vs. Deal Probability

- The box plot in Figure 10 shows the distribution of deal probabilities for each parent category and user type.
- Deal probabilities for services are the highest, followed by transport and animals.
- Shop users are present only in 4 parent categories: 'Real estate', 'Transport', 'Animals', and 'Services'.
- Shops have the lowest median deal probability, while private users have the highest median deal probability.

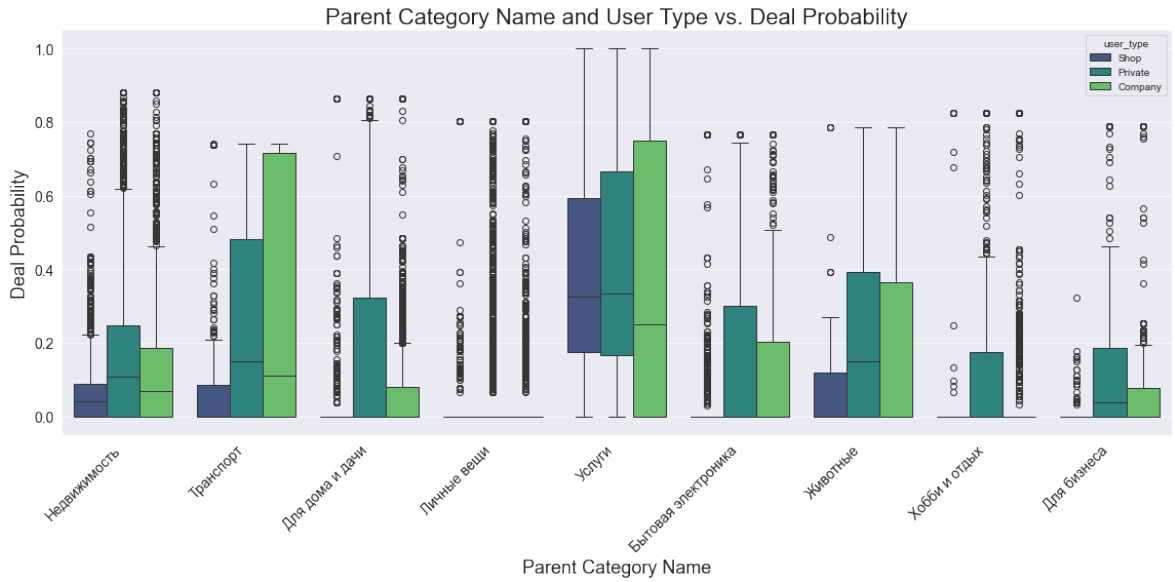

Figure 10: User Type and Parent Category vs. Deal Probability

2.8.2 Correlation Analysis

- The correlation matrix in Figure 11 shows the correlation coefficients between numerical features in the dataset.
- There is a correlation between the length of the title and the length of the description, as well as between the number of words in the title and the number of words in the description.
- Surprisingly, there is a correlation between image_top_1 and deal_probability, which could indicate that the image classification code may have some influence on the deal probability and may have some order in it, but it was supposed to be a categorical nominal feature. We may try to use it as a numerical feature in the model instead of encoding it.
- There is also a large negative correlation between image_top_1 and params_length, which could indicate that parameters are somehow related to the image classification code.
- Other then that, there are no significant correlations between deal probability and other numerical features.

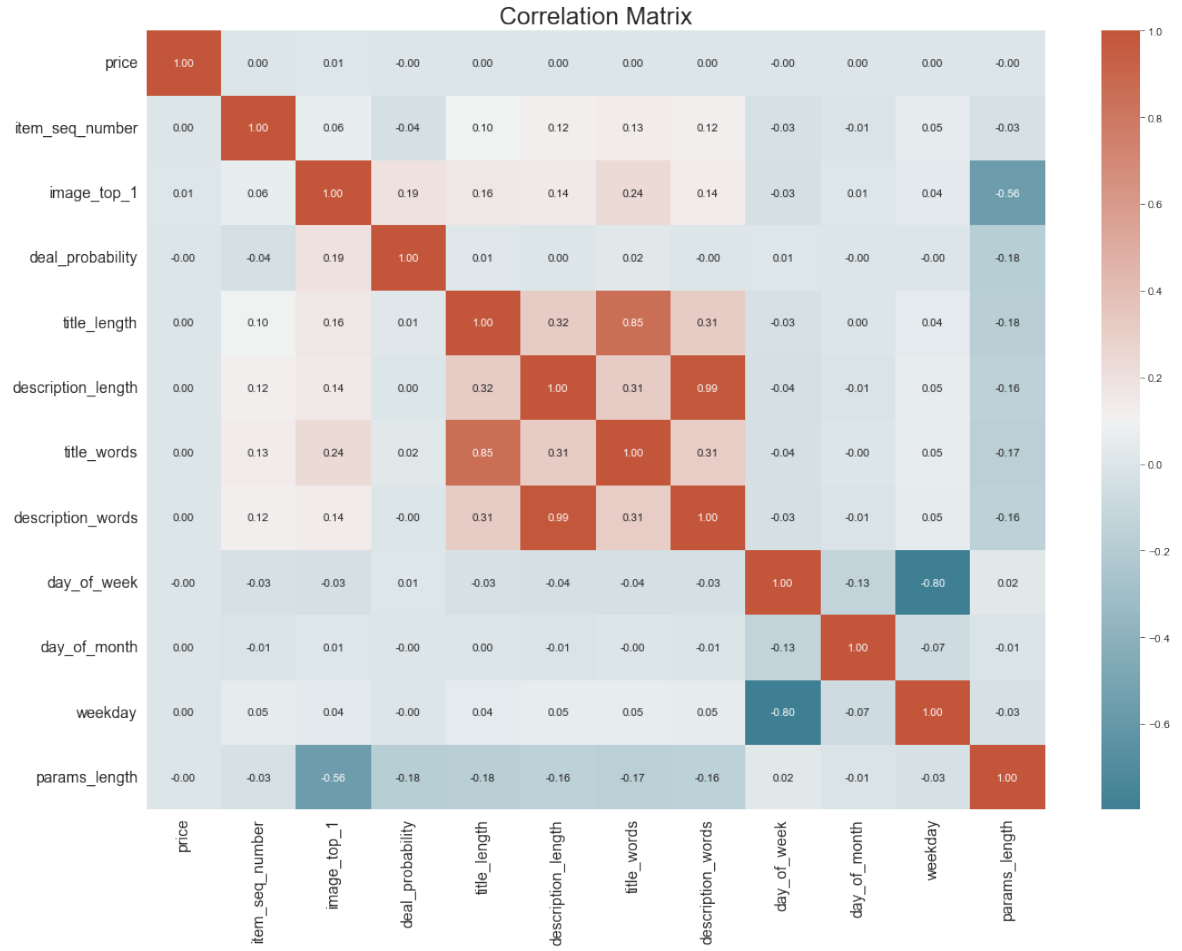

Figure 11: Correlation heatmap of numerical features

2.8.3 Initial Hypotheses and Impact

Based on our initial exploration, we hypothesize that:

- **Price Sensitivity:** Ads with moderate prices (neither too low nor too high) are more likely to have higher deal probabilities.
- **User Type Influence:** Ads posted by private users may have different demand patterns compared to those posted by companies or shops.
- **Category Impact:** Certain categories, such as "Personal belongings" and "For the home and garden," may have higher demand due to their prevalence in the dataset.

2.8.4 Data Requirements

To ensure the reliability and accuracy of our predictions, we define the following data requirements:

- **Completeness:** All mandatory fields, such as 'title', 'description', 'price', and 'deal_probability', should not have missing values. Optional parameters ('param_1', 'param_2', 'param_3') should be imputed as 'missing' if not provided.
- **Validity:** Numerical features like 'price' and 'deal_probability' should have values within plausible ranges (e.g., 'price' > 0, 'deal_probability' between 0 and 1). Categorical features should have predefined valid categories.
- **Consistency:** Data should be consistent in terms of types and formats. For example, 'activation_date' should be a valid date, and 'user_type' should be one of the predefined categories (e.g., Private, Company, Shop).

- **Accuracy:** The data should accurately reflect real-world conditions. Any anomalies or outliers should be investigated and handled appropriately.

2.8.5 Data Quality Verification Report

We conducted a thorough quality check of the dataset, focusing on completeness, correctness, and the presence of missing values.

Completeness:

- The dataset is complete with respect to fields like 'title', 'parent_category', 'category_name', 'city' and etc. which are essential for demand prediction.
- Optional fields like 'param_1', 'param_2', and 'param_3' have a high number of missing values, which we addressed by imputing them as 'missing'; 'description', 'price', 'image_top_1' has about 7% missing data and should be imputed.

Correctness:

- The 'deal_probability' values are within the expected range of 0 to 1.
- There are no duplicate entries in the dataset.
- The 'activation_date' values are valid dates within the specified range.

Missing Values:

- Missing values were primarily found in 'description', 'price', 'param_1', 'param_2', 'param_3', and 'image_top_1'.
- We addressed missing values by imputing with appropriate placeholder values for categorical features and price feature with the median price within each category to maintain the distribution of prices within similar ads.

2.9 Project Feasibility

2.9.1 Inventory of Resources

The project has access to the following resources:

- **Personnel:** A team of consiting of 3 students with the following roles: data scientist, machine learning engineer and data engineer.
- **Data:** Access to historical advertisement data from Avito, including ad descriptions, prices, and other information.
- **Computing Resources:** PC with 8 CPU cores, 16 GB of RAM and NVIDIA GPU RTX 3050 with 4GB of VRAM.
- **Software:** Machine learning libraries such as scikit-learn, and PyTorch, skorch; data processing tools like pandas; data visualization tools like Matplotlib; ETL Framework Apache Airflow and ZenML.

2.9.2 Requirements, Assumptions, and Constraints

Requirements:

- The project must be completed within two months.
- The model should achieve an RMSE of 0.25 or lower.

Assumptions:

- The data provided is representative of the overall advertisement behavior on Avito.

- The features selected for modeling are sufficient to capture the factors influencing deal probability.

Constraints:

- Limited historical data may restrict the model's ability to generalize.
- High-dimensional data may require significant computational resources for processing and model training.
- Large amount of unique categories in categorical features may cause curse of dimensionality after one-hot encoding.

2.9.3 Risks and Contingencies

Risks:

- **Data Quality Issues:** Poor data quality could affect model performance.
- **Model Overfitting:** The model may overfit the training data and perform poorly on unseen data.
- **Computational Limitations:** High computational requirements may delay model training and deployment.

Contingencies:

- Implement data quality checks and perform data cleaning.
- Implement cross-validation techniques to mitigate overfitting.
- Optimize computational resources and GPU-based algorithms.

2.9.4 Costs and Benefits

Costs:

- Time and effort spent by the data science and engineering team.
- Potential costs associated with data storage and management.

Benefits:

- Improved accuracy in demand predictions, leading to optimized ad listings and pricing strategies.
- Increased user satisfaction and retention on the Avito platform.
- Higher revenue generation through improved transaction volumes.

2.9.5 Feasibility Report

Based on the initial exploration and data quality checks, the project is feasible. The data is sufficiently detailed and comprehensive to support the development of a robust machine learning model. The potential benefits in terms of increased revenue and user satisfaction justify the investment in this project.

We conducted a preliminary proof-of-concept (POC) using a linear regression model and our full feature engineering pipeline. The initial results show an RMSE of 0.2633, which is already close to the target RMSE of 0.25. With a better model and improved feature engineering, it is likely that we can achieve the desired performance.

Overall, the project is viable and has the potential to significantly enhance Avito's service offering, providing a competitive edge in the online marketplace.

2.9.6 Project Plan

Project plan and gantt chart is available in ClickUp by the following link: https://sharing.clickup.com/9012093001/g/h/8cjk829-412/85b8479c837f6cd

3 Data Preparation

Data preparation is a crucial step in any machine learning project as it involves constructing the final dataset from the raw data. This phase includes various tasks such as data selection, cleaning, construction, and standardization to ensure that the data fed into the machine learning pipelines is optimized for better modeling performance.

3.1 Select Data

3.1.1 Data Selection Rationale

Given the project's constraints and objectives, we decided to use a subset of 10,000 rows sampled from the initial dataset. This decision was based on the balance between computational efficiency and maintaining enough data variety for model training.

Columns Excluded

- **'image':** Excluded due to the lack of resources and time to process image data effectively.
- **'item_id':** Removed as it is a unique identifier with no predictive power regarding ad demand.
- **'user_id':** Omitted because there are not other specific user information that could be used to make this feature relevant to predicting the demand.

3.1.2 Data Inclusion

The columns retained are the ones that are essential for predicting advertisement demand, including ad descriptions, prices, categorical data about the ad's category, and user type and price.

3.2 Clean Data

3.2.1 Handling Missing Values

Missing data was addressed using following strategies to improve the dataset quality:

- **'param_1', 'param_2', 'param_3':** Imputed with 'missing' to treat missing values as a separate category.
- **'description':** Missing descriptions were filled with placeholder text 'No description.' to maintain uniformity.
- **'price':** Missing prices were imputed with the median price of the respective category to reflect typical values.
- **'image_top_1':** Missing entries were assigned a new category by using the maximum value+1.

3.2.2 Category Collapsing

For columns with numerous unique categories, we reduced complexity by retaining the top 10 categories and grouping all others into an 'Other' category. This was done for:

- **'category_name', 'city', 'param_1', 'param_2', 'param_3':** This approach was chosen to reduce the dimensionality and avoid sparse data issues post one-hot encoding. We ensured that top 10 categories encompassed more then 50% of all entries in the dataset.

3.3 Construct Data

3.3.1 Derived Attributes

New attributes created include:

- **'description_length', 'title_length', 'param_length':** These features were engineered based on exploratory data analysis insights to capture the potential impact of content length on ad demand.

3.3.2 Text Feature Construction

To handle the text data efficiently:

- **Text Vectorization:** ‘description‘ and ‘title‘ were transformed using TfIdf vectorization with 128 features for each of two columns.
- **Dimensionality Reduction:** The number of features from the TfIdf vectorization was reduced using PCA, where ‘description‘ was brought down to 16 features and ‘title‘ to 8 features. This was essential to capture the most relevant information while keeping the number of dimensions manageable.

3.3.3 Time Feature Extraction

Given the dataset's time-related attributes, we extracted and encoded the following features:

- **‘day_of_week‘ and ‘day_of_month‘:** From the ‘activation_date‘, we extracted the day of the week and the day of the month to capture temporal trends and cycles in advertisement activity and demand. We choose only these features, as our EDA showed, that all items in the dataset were posted on the same year and within only two different months.
- **Sine-Cosine Encoding:** We applied sine-cosine transformations to these features to maintain their cyclical nature.

3.4 Standardize Data

3.4.1 Normalization and Encoding

All categorical features were one-hot encoded to transform them into a format suitable for the model. Numerical features were scaled using Standard Scaler to normalize their distribution.

Uniform Data Schema

To ensure consistency across the data preparation and modeling phases, we:

- Trained and saved all encoders and scalers on a separate dataset of 100,000 non-overlapping samples with train and test sets to prevent data leakage.
- Used these trained components for subsequent data transformations.
- Ensured all features are in a specific order by sorting columns alphabetically, which is important, since input layer of the neural network expects features to be in the same order every time.

3.5 Automated Workflows and Pipelines

3.5.1 Apache Airflow Data Extraction Workflow

An automated workflow, implemented in Apache Airflow (‘data_extract_dag.py‘), performs four tasks for data extraction and versioning.

1. **Data Sampling:** Extracts a new data sample.
2. **Data Validation:** Validates the sample using Great Expectations to ensure data quality.
3. **Data Versioning:** Versions the data sample using DVC and logs the version in ‘./configs/datasets.yaml‘.
4. **Data Loading:** Loads the sample to the DVC-managed data store.

3.5.2 ZenML Pipeline for Data Preparation

The ETL pipeline ('transform_data.py') utilizes ZenML for executing four data processing tasks. This setup ensures that the data transformations and validations are consistently applied, and the resulting features are properly versioned and stored.

1. **Data Extraction:** Data is extracted from the DVC store based on the version provided.
2. **Data Transformation:** Applies preprocessing to transform the data sample into features suitable for machine learning models, using pipeline specified above.
3. **Feature Validation:** New expectations are created and validated using Great Expectations to ensure the quality of the features.
4. **Feature Loading and Versioning:** Features are loaded into the ZenML artifact store with versioning.

A second Apache Airflow DAG ('data_prepare_dag.py') manages the integration of the data extraction and preparation pipelines.

4 Model Engineering

4.1 Literature Research on Similar Problems

Based on our review of existing literature, we identified several studies that address problems similar to ours, namely demand prediction for online advertisements and click-through prediction.

In the first relevant study [1], the authors utilize both text features, processed through TfIdf vectorization and FastText embeddings, and different image features, such as features extracted using NIMA for aesthetic evaluation, to predict e-commerce advertisement demand. They experimented with various models, including AdaBoost, XGBoost, LightGBM, Multi-layer Perceptrons (MLP), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Unit (GRU) networks. The most effective models were found to be LightGBM, combined GRU and DNN, and MLP.

Another study [2] focuses on click-through prediction for display advertising, which is a task of predicting, whether an ad will be clicked or not, quite similar to our problem. They compare various deep learning approaches, including a proprietary model based on residual networks, against traditional methods. The study uses datasets from iPinYou, Criteo, and Avazu, and demonstrates superior performance of deep neural networks, particularly the residual network-based approach.

These insights are relevant to our project as they provide proven methodologies and baselines for performance, guiding our approach in model selection and implementation.

4.2 Quality Measures

The quality of the machine learning models in our project is assessed using the following criteria:

- **Root Mean Squared Error (RMSE):** The primary metric for evaluating the accuracy of our predictions. It measures the average difference between predicted and actual deal probabilities, with lower values indicating better performance. RMSE gives a relatively high weight to large errors and is more sensitive to outliers. It is chosen as a primary metric, since it is better suited for cases where large errors are particularly undesirable.
- **Model Scalability:** Ability to handle increasing amounts of data without a significant degradation in performance or speed.
- **Model Stability:** Consistency of the model's performance over time and across various data segments.

These measures are crucial because they ensure that the model not only performs well on average but also behaves predictably across different scenarios, which is crucial for deployment in dynamic environments like online marketplaces.

4.3 Model Selection

For our project, we selected two models: a simple Multi-layer Perceptron (MLP) and a more complex Residual Network (ResNet). These choices are driven by the need to balance between model complexity and computational efficiency, in line with our business objectives and data characteristics.

4.3.1 MLP Architecture

The MLP model consists of three layers with the hyperparameter configurations presented in Table 2. This architecture serves as a baseline model and is known for its ability to learn complex patterns in data. The model's architecture is illustrated in Figure 12.

Parameter	Options	Best Value
Hidden Layer 1 Size	[32, 64, 128]	32
Hidden Layer 2 Size	[16, 32, 64]	16
Hidden Layer 3 Size	[8, 16, 32]	32

Table 2: MLP Model HyperParameters

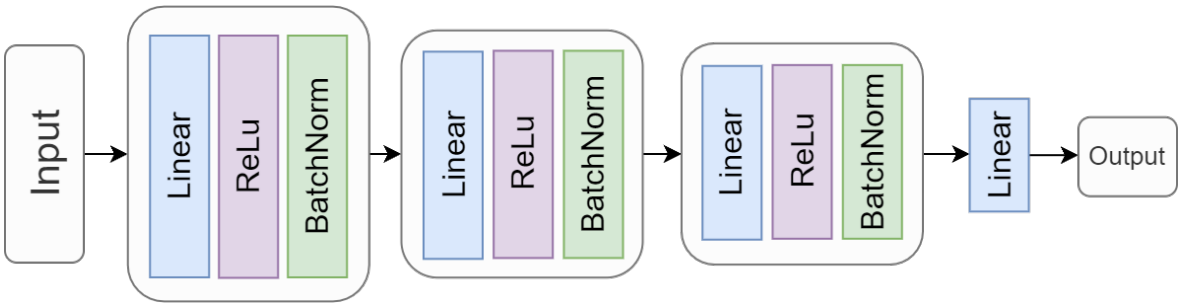

Figure 12: MLP Model Architecture

4.3.2 ResNet Architecture

The ResNet model uses residual blocks, each consisting of two linear layers accompanied by batch normalization and dropout at the end, to facilitate learning deeper representations without the vanishing gradient problem, while dropout layer helps in preventing overfitting. The embedding size and the number of residual blocks were optimized as shown in Table 3. The model's architecture is illustrated in Figure 13.

Parameter	Options	Best Value
Embedding Size	[16, 32, 64, 128]	128
Number of Residual Blocks	[1, 3, 5, 10]	3
Dropout Rate	[0.35, 0.5, 0.75]	0.75

Table 3: ResNet Model HyperParameters

4.3.3 Model Signature

The input and output dimensions of both models are as follows:

- **Input Dimension:** 125
- **Output Dimension:** 1

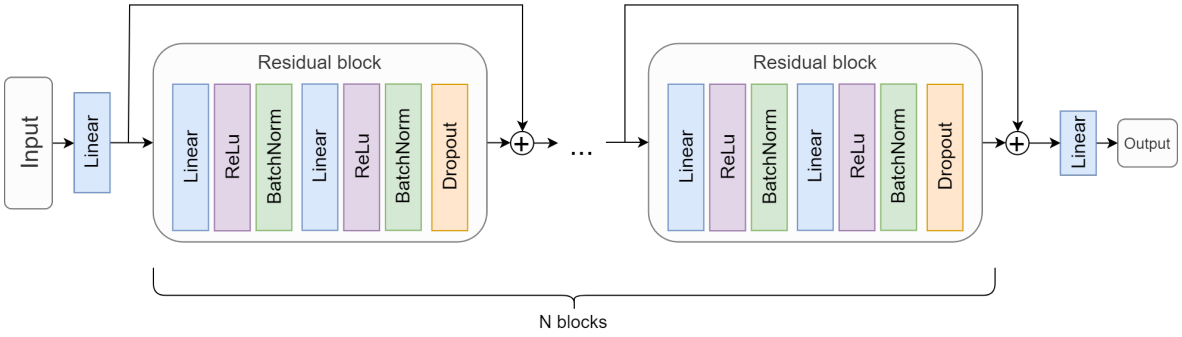

Figure 13: Residual Network Model Architecture

4.4 Domain Knowledge Incorporation

Incorporating domain knowledge involves ensuring that the selected models and metrics align with the business needs of predicting advertisement demand effectively. The use of RMSE aligns with the business's need for accurate predictions, while the choice of MLP and ResNet architectures is influenced by their proven effectiveness in similar tasks as identified through our literature research.

4.5 Model Training

For training, we employed a 3-fold cross-validation strategy using GridSearch for hyperparameter optimization. The test datasets comprise the next 10,000 samples, ensuring no overlap with training data. We used Root Mean Squared Error (RMSE) as our loss function, as in our experiments we found, that RMSE loss results in better models, then MSE loss. For the optimization of our models, we selected the AdamW optimizer. This choice was made after testing showed that AdamW has superior performance for our datasets and model architectures. Both models were trained using a fixed learning rate of 5×10^{-4} and a weight decay of 10.0. We limited the training to 25 epochs based on validation performance, since larger number of epochs usually was resulting in overfitting. The performance of both models after training is summarized in Table 4.

The performance of both models after training is summarized below:

Model	RMSE	MSE	MAE
MLP	0.242	0.0585	0.162
ResNet	0.246	0.0606	0.166

Table 4: Model Performance Summary

These results indicate that both models meet the project's success criteria, with the MLP slightly outperforming the ResNet.

We also used parallel plots to visually analyze the impact of hyperparameters on model performance. For the MLP, the parallel plot (Figure 14) indicated that variations in hyperparameters did not significantly impact the performance. On the other hand, the ResNet model showed a more significant variation in RMSE across different hyperparameter settings, as shown in the parallel plot in Figure 15.

4.6 Assure Reproducibility

4.6.1 Method Reproducibility

To ensure the reproducibility of our models, we have documented the following:

- **Model Architectures:** Detailed descriptions and diagrams of both the MLP (Figure 12) and ResNet (Figure 13) architectures are provided.
- **Hyperparameters:** The specific hyperparameters used for each model are outlined in Tables 2 and 3.

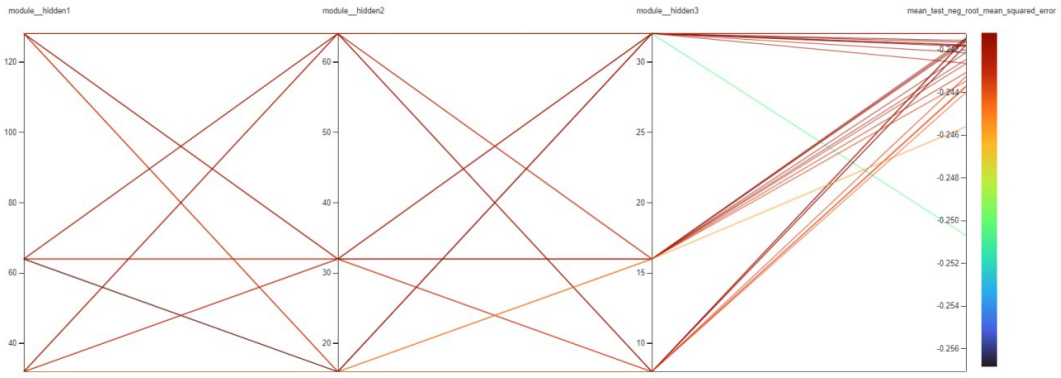

Figure 14: Parallel Plot for MLP Hyperparameters

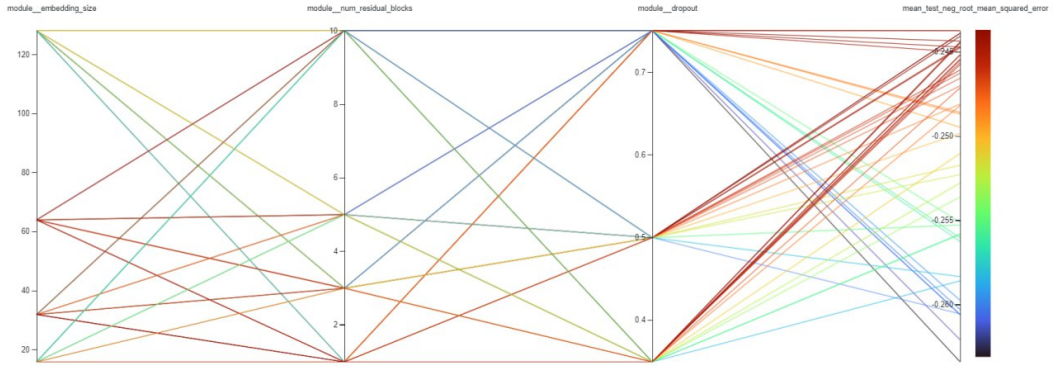

Figure 15: Parallel Plot for ResNet Hyperparameters

- **Software and Libraries:** We utilized skorch and PyTorch to implement the models. Skorch allows the use of PyTorch models with grid search methods from scikit-learn.
- **Environment:** The models were trained and tested in a WSL2 environment on a Windows PC.
- **Random Seed:** We primarily used the seed '42' for hyperparameter optimization and evaluation.

4.6.2 Result Reproducibility

To verify result reproducibility, we evaluated our models using five different random seeds. The evaluation metrics for both models across these seeds are presented in Tables 5 and 6.

Metric	Values	Average	Variance
MAE	[0.1606, 0.1615, 0.1631, 0.1632, 0.1625]	0.1622	9.67×10^{-7}
MSE	[0.0586, 0.0586, 0.0585, 0.0586, 0.0586]	0.0586	1.28×10^{-9}
RMSE	[0.2421, 0.2421, 0.2419, 0.2420, 0.2420]	0.2420	5.47×10^{-9}

Table 5: MLP Test Metrics for Different Seeds

The MLP demonstrates very low variance in performance across different seeds, with an average RMSE lower than that of ResNet. While ResNet has slightly higher variance, it still remains relatively low.

4.6.3 Experimental Documentation

Our experiment tracking involves detailed logs of all model training and validation steps, ensuring that each model iteration is fully documented and reproducible.

Metric	Values	Average	Variance
MAE	[0.1683, 0.1589, 0.1578, 0.1659, 0.1804]	0.1663	6.59×10^{-5}
MSE	[0.0597, 0.0607, 0.0628, 0.0592, 0.0607]	0.0606	1.51×10^{-6}
RMSE	[0.2444, 0.2464, 0.2506, 0.2433, 0.2464]	0.2462	6.16×10^{-6}

Table 6: ResNet Test Metrics for Different Seeds

5 Model Evaluation

5.1 Model Validation Report

The trained MLP model was evaluated on a test dataset (10,000 samples) not used during training. The MLP model achieved the following performance:

- **MAE:** 0.1622
- **MSE:** 0.0586
- **RMSE:** 0.2420

These results indicate that the model meets the project's initial success criterion of achieving an RMSE of 0.25 or lower.

5.1.1 Giskard Validation and Vulnerability Analysis

While the overall performance is promising, Giskard validation revealed some areas for potential improvement within the model's predictions. These areas highlight specific data slices where the model's performance slightly deviates from the global average:

1. Category-Specific Performance Insights:

- **Children's Products and Toys:** The model exhibits a moderately higher MSE (+91.21%) for ads classified as "Children's products and toys" compared to other categories. This suggests difficulty capturing the nuances of pricing and demand within this category, possibly due to factors like brand loyalty, seasonality, or a wider price range.
- **Transport:** Similarly, the "Transport" category displays a higher MAE (+53.56%) compared to the global average. This discrepancy might be due to the model's inability to account for variations in vehicle condition, brand reputation, or fluctuations in the used car market.

2. Feature Engineering Refinement Opportunities:

- **PCA Components:** Minor performance variations are observed within specific ranges of certain PCA components derived from the title and description data (`title_pca_7`, `description_pca_13`, `description_pca_9`, `description_pca_12`, `description_pca_2`, `description_pca_3`, and `description_pca_15`). This sensitivity to engineered features suggests that the PCA transformation might not be optimally capturing the underlying information or that these components are correlated with other features in a way that introduces bias. A more complex approach to text data may be needed, such as usage of RNNs, to fully capture the underlying information.

3. Further Exploration of Missing Value Imputation:

- **Missing Parameters:** Ads with missing values in `param_2` and `param_3`, imputed as 'missing,' exhibit slightly higher MSE values (+35.37% and +28.39% respectively) compared to ads with complete data. This suggests that the model might be interpreting the 'missing' category as a strong signal, potentially leading to biased predictions for ads lacking this specific information. Maybe an alternative method of imputation might solve the issue.
- **Item Sequence Number:** A minor increase in MSE (+30.1%) is observed within a specific range of the `item_seq_number` feature. This issue requires further investigation to understand if there is a genuine relationship between the order in which a user posts ads and ad demand, or if it represents a false correlation.

5.2 Discussion

The model validation results demonstrate that the MLP model achieves the pre-defined success criteria for RMSE, indicating its potential for providing valuable demand predictions on the Avito platform. The Giskard validation, while highlighting areas for potential improvement, does not present significant concerns.

5.3 Deployment Decision

Based on the overall positive model validation results and the manageable vulnerability issues, we recommend proceeding with the deployment of the MLP model.

6 Model Deployment

6.1 Deployment Strategy

The deployment of the advertisement demand prediction model is a critical step in realizing its practical value for Avito's users. Our deployment strategy focuses on providing both an API endpoint for programmatic access and a user-friendly interface for direct interaction.

6.1.1 API Endpoint Deployment

To ensure scalability and maintainability, we have chosen to deploy the model as an API endpoint using two distinct approaches:

1. **Docker Deployment:** MLflow, a platform for managing the end-to-end machine learning lifecycle, is used to package the trained MLP model, along with its dependencies, into a Docker image. This containerized approach simplifies deployment and ensures consistency across different environments. The MLflow deployment offers a REST API endpoint, allowing for integration with other services or applications.
2. **Flask API Deployment:** For direct and localized deployment, we developed a custom API using Flask, a lightweight web framework in Python. This API wraps the trained model and exposes an endpoint to process incoming prediction requests. While this approach allows for greater customization and control over the API's behavior, it may require additional configuration and management compared to the containerized MLflow deployment.

6.1.2 Gradio UI for User Interaction

To improve user experience and provide direct access to the model's predictions, we developed a graphical user interface (GUI) using Gradio, a Python library for building machine learning demos. The GUI enables users to input information about their advertisement listing, such as title, description, price, and category. Upon submission, the GUI interacts with the deployed model API (either using API from Docker or Flask) to retrieve and display the predicted deal probability.

6.2 Inference Hardware

The model can be deployed on a relatively modest hardware configuration. The size of the model does not exceed 200 KB and can be inferenced on CPU without the need of GPU. Our deployment environment consists of:

- Processor: Intel Core i5
- Memory: 16 GB RAM
- GPU: NVIDIA RTX 3050 with 4GB VRAM

6.3 Model Evaluation Under Production Conditions

To ensure the model's accuracy in a deployment setting, we performed an additional evaluation using a new sample of 10,000 advertisement listings unseen during model training. These evaluations were conducted by using the MLflow Docker containerized model.

Model	RMSE	MSE	MAE
MLP	0.2415	0.0583	0.1618

Table 7: Model Performance in Production Environment

The results in Table 7 indicate that the model maintains consistent performance in production setting and that deployment was correct, with RMSE values closely aligned with those observed during the model validation phase (Chapter 5).

6.3.1 Meeting Business Success Criteria

The consistent RMSE scores below the target threshold of 0.25 demonstrate that the deployed model successfully meets the project's primary business success criterion of providing accurate demand predictions.

6.3.2 Economic Success Criteria

While the model's accuracy in predicting deal probability may be an indicator of its potential economic impact, direct measurement of its effect on Avito's revenue requires further monitoring and analysis of user behavior post-deployment. This analysis should focus on:

- **Transaction Volume:** Tracking changes in the number of successful transactions on platform after the model's deployment.
- **User Engagement:** Analyzing how sellers are utilizing the demand predictions, including any observed changes in pricing strategies, listing quality, and overall engagement with the platform.

7 Conclusion

This project addressed the challenge of predicting demand for online advertisements on the Avito platform, aiming to enable sellers with insights to optimize their listings and improve their chances of successful transactions. Through a systematic approach consisting of data understanding, preparation, model engineering, evaluation, and deployment, we successfully developed and implemented a robust machine learning solution.

Key Achievements:

- Developed and deployed a Multi-layer Perceptron (MLP) model that accurately predicts advertisement deal probability, achieving an RMSE consistently below the target threshold of 0.25.
- Implemented an efficient and reproducible machine learning pipeline using industry-standard tools like DVC, Apache Airflow, and ZenML.
- Validated the model and the data using Giskard and Great Expectations.
- Deployed the model through both a containerized MLflow API and a custom-built Flask API.
- Created a user-friendly Gradio interface that allows sellers to directly interact with the model.

Future Improvements:

- Investigate and incorporate additional features, such as image data and user-specific information, to potentially enhance the model's predictive power further.
- Explore more complex architectures, like Recurrent Neural Networks (RNNs) or Transformer, to capture information from the text data more efficiently.
- Monitor the model's performance in the production environment and analyze its impact on key business metrics, such as transaction volume and user engagement.

References

[1] S. Rai, A. Gupta, A. Anand, A. Trivedi, and S. Bhadauria, "Demand prediction for e-commerce advertisements: A comparative study using state-of-the-art machine learning methods," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, 2019.

[2] M. Liu, S. Cai, Z. Lai, L. Qiu, Z. Hu, and Y. Ding, "A joint learning model for click-through prediction in display advertising," *Neurocomputing*, vol. 445, pp. 206–219, 2021.