Predicting Advertisement Demand on Avito Platform

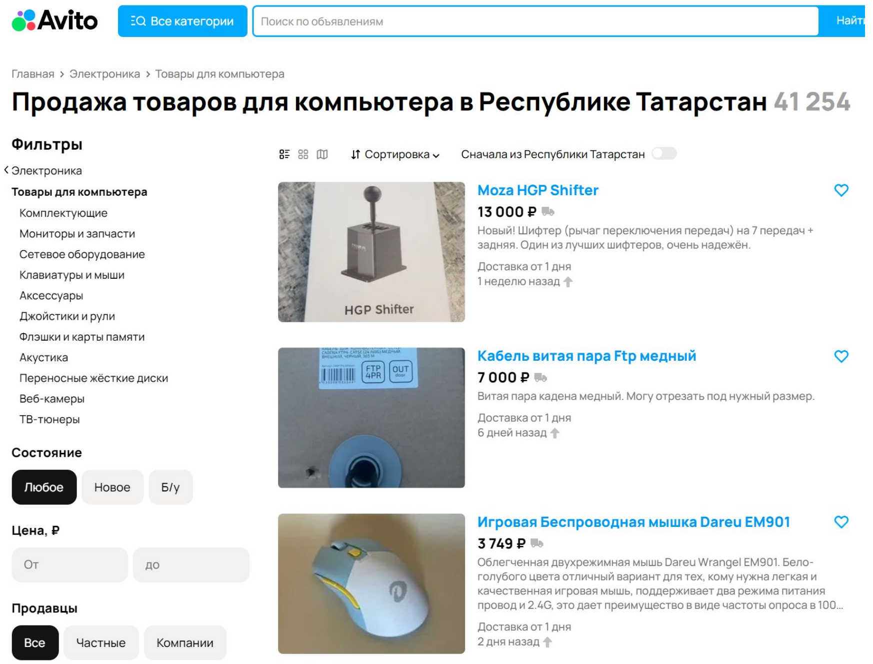
Artur Zagitov, Ildar Zalialiev, Artem Nazarov

Business problem

Sellers often struggle to predict demand for their products.

This leads to:

- Underpricing and lost revenue
- Overpricing and low buyer interest
- Frustration for both buyers and sellers



Продажа товаров для компьютера в Республике Татарстан 41 254

Сначала из Республики Татарстан

Moza HGP Shifter
13 000 ₽
Новый! Шифтер (рычаг переключения передач) на 7 передач + задняя. Один из лучших шифтеров, очень надежён.
Доставка от 1 дня
1 неделю назад

Кабель витая пара Ftp медный
7 000 ₽
Витая пара кадена медный. Могу отрезать под нужный размер.
Доставка от 1 дня
6 дней назад

Игровая Беспроводная мышка Dareu EM901
3 749 ₽
Облегченная двухрежимная мышь Dareu Wrangel EM901. Бело-голубого цвета отличный вариант для тех, кому нужна легкая и качественная игровая мышь, поддерживает два режима питания провод и 2.4G, это дает преимущество в виде частоты опроса в 100...
Доставка от 1 дня
2 дня назад

Business understanding

Business Objectives:

- Improve ad listing efficiency.
- Reduce underpriced and overpriced ads.
- Enhance user satisfaction and platform engagement.

ML Objectives:

- Develop a machine learning model to predict ad deal probability.
- Achieve a Root Mean Squared Error (RMSE) of 0.25 or lower on unseen data.

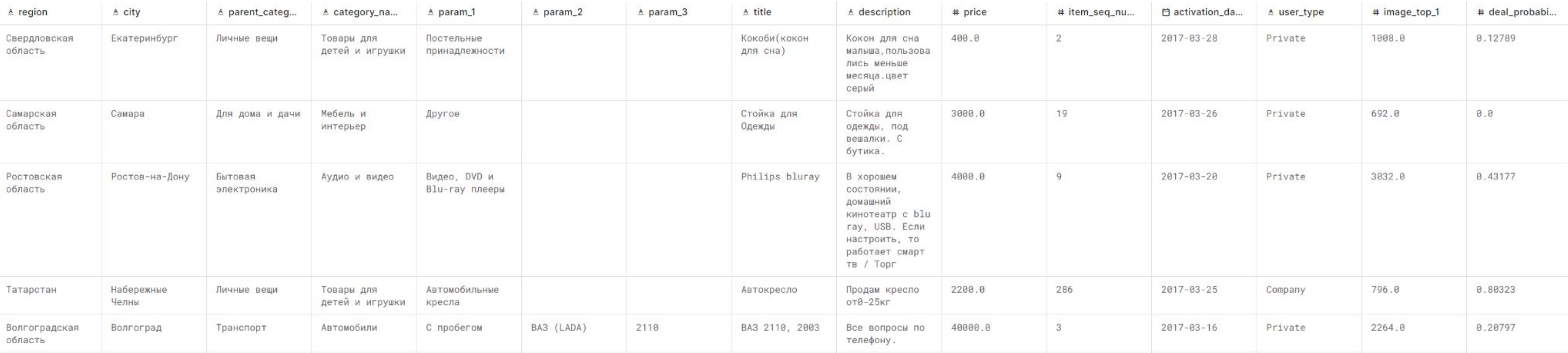
Success Criteria:

Business: Increased transaction volume and user engagement.

ML: RMSE below 0.25 on a held-out test dataset.

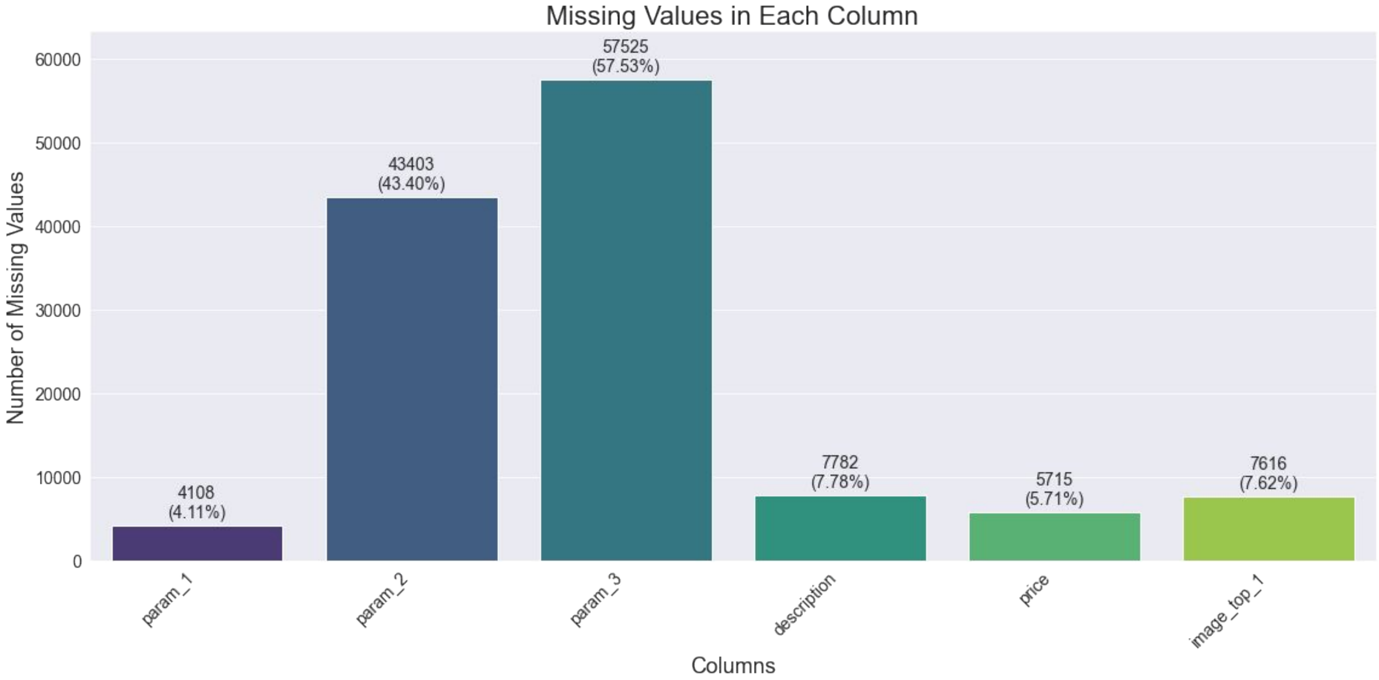
Data Description

- **Size:** 1.5 million records
- **Source:** Avito platform, accessed via Kaggle
- **Features:** Ad title, description, price, category, region, user type, and more.
- **Target:** Deal Probability - [0, 1]



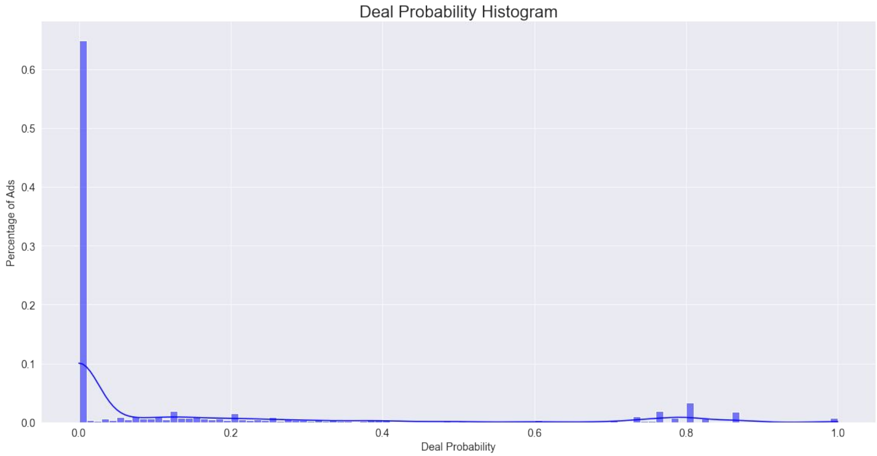
region	city	parent_categ...	category_na...	param_1	param_2	param_3	title	description	price	item_seq_nu...	activation_da...	user_type	image_top_1	deal_probabi...
Свердловская область	Екатеринбург	Личные вещи	Товары для детей и игрушки	Постельные принадлежности			Кокоби(кокон для сна)	Кокон для сна малыша,пользовались меньше месяца.цвет серый	400.0	2	2017-03-28	Private	1008.0	0.12789
Самарская область	Самара	Для дома и дачи	Мебель и интерьер	Другое			Стойка для Одежды	Стойка для одежды, под вешалки. С бутика.	3000.0	19	2017-03-26	Private	692.0	0.0
Ростовская область	Ростов-на-Дону	Бытовая электроника	Аудио и видео	Видео, DVD и Blu-ray плееры			Philips bluray	В хорошем состоянии, домашний кинотеатр с blu ray, USB. Если настроить, то работает смарт тв / Торг	4000.0	9	2017-03-20	Private	3032.0	0.43177
Татарстан	Набережные Челны	Личные вещи	Товары для детей и игрушки	Автомобильные кресла			Автокресло	Продам кресло от0-25кг	2200.0	286	2017-03-25	Company	796.0	0.80323
Волгоградская область	Волгоград	Транспорт	Автомобили	С пробегом	ВАЗ (LADA)	2110	ВАЗ 2110, 2003	Все вопросы по телефону.	40000.0	3	2017-03-16	Private	2264.0	0.20797

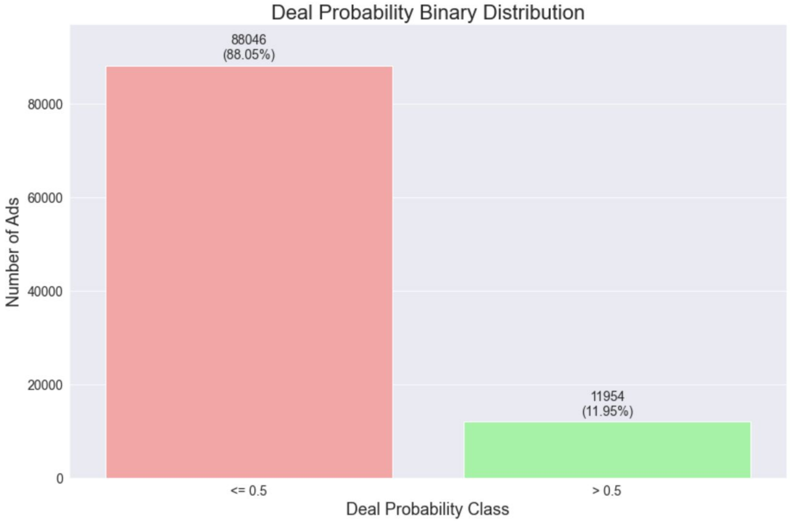
EDA - Missing values



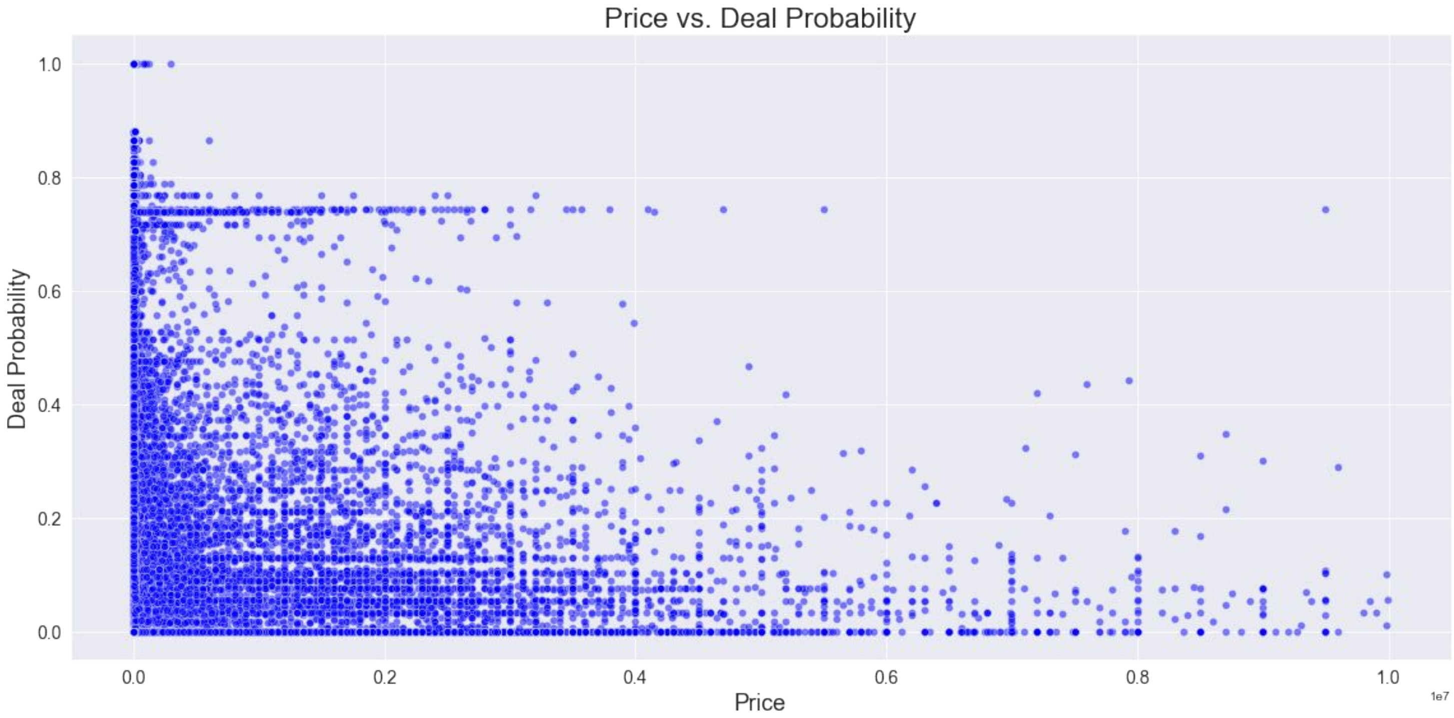
EDA - Target

- More than 66% of ad listings has deal probability of less than 0.05 (5%)
- More than 88% of ad listings has deal probability of less then 0.50 (50%)
- A small peak around 0.8, indicating a smaller number of ads with higher, deal probabilities.

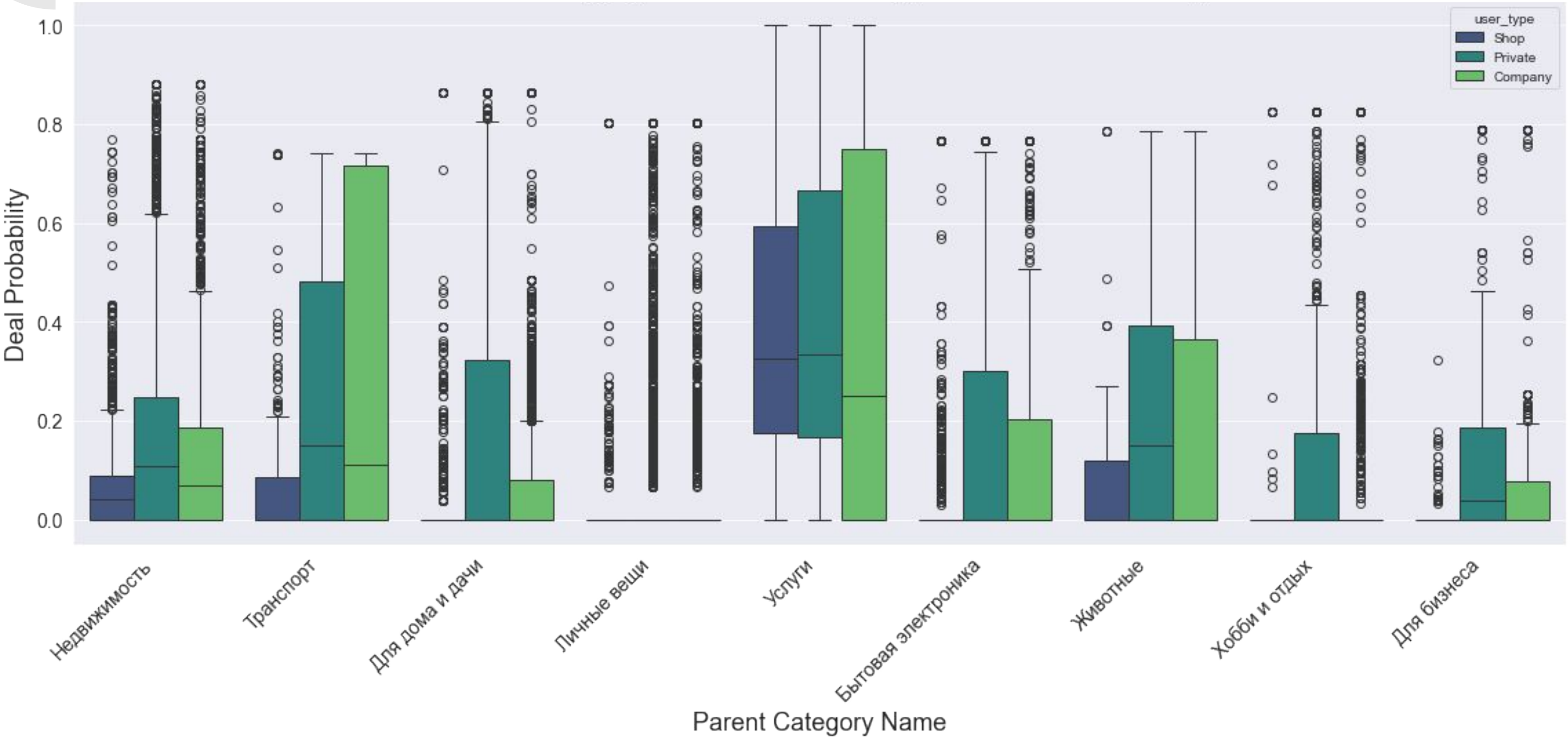




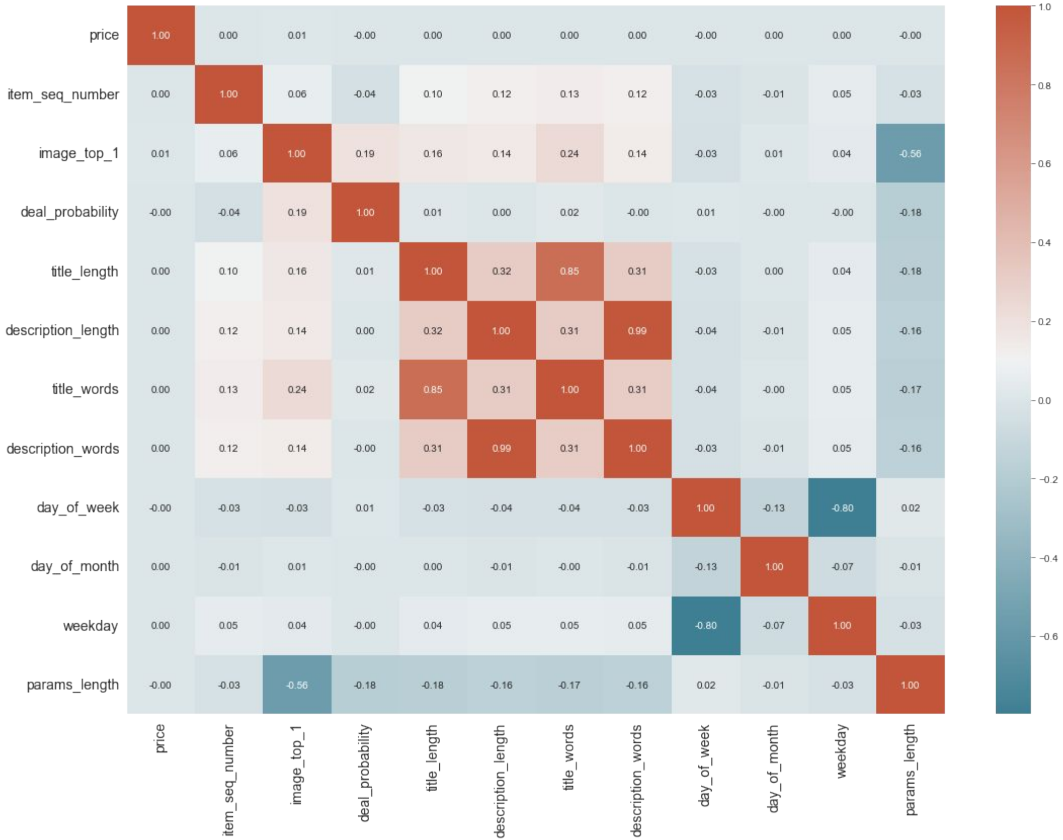
EDA - Price vs Target



EDA - User Type and Parent Category vs. Deal Probability



EDA - Correlations

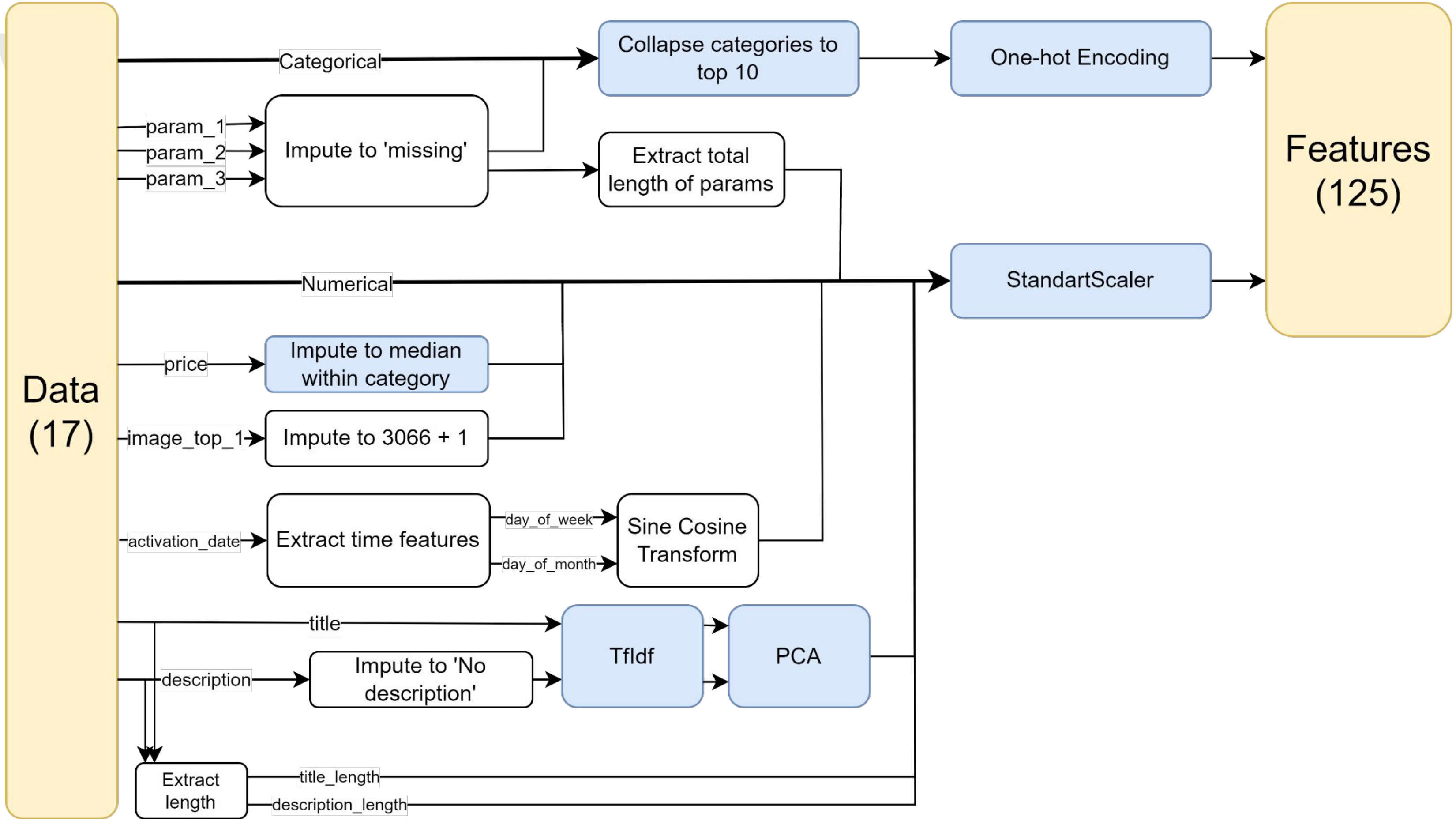


	price	item_seq_number	image_top_1	deal_probability	title_length	description_length	title_words	description_words	day_of_week	day_of_month	weekday	params_length
price	1.00	0.00	0.01	-0.00	0.00	0.00	0.00	0.00	-0.00	0.00	0.00	-0.00
item_seq_number	0.00	1.00	0.06	-0.04	0.10	0.12	0.13	0.12	-0.03	-0.01	0.05	-0.03
image_top_1	0.01	0.06	1.00	0.19	0.16	0.14	0.24	0.14	-0.03	0.01	0.04	-0.56
deal_probability	-0.00	-0.04	0.19	1.00	0.01	0.00	0.02	-0.00	0.01	-0.00	-0.00	-0.18
title_length	0.00	0.10	0.16	0.01	1.00	0.32	0.85	0.31	-0.03	0.00	0.04	-0.18
description_length	0.00	0.12	0.14	0.00	0.32	1.00	0.31	0.99	-0.04	-0.01	0.05	-0.16
title_words	0.00	0.13	0.24	0.02	0.85	0.31	1.00	0.31	-0.04	-0.00	0.05	-0.17
description_words	0.00	0.12	0.14	-0.00	0.31	0.99	0.31	1.00	-0.03	-0.01	0.05	-0.16
day_of_week	-0.00	-0.03	-0.03	0.01	-0.03	-0.04	-0.04	-0.03	1.00	-0.13	-0.80	0.02
day_of_month	0.00	-0.01	0.01	-0.00	0.00	-0.01	-0.00	-0.01	-0.13	1.00	-0.07	-0.01
weekday	0.00	0.05	0.04	-0.00	0.04	0.05	0.05	0.05	-0.80	-0.07	1.00	-0.03
params_length	-0.00	-0.03	-0.56	-0.18	-0.18	-0.16	-0.17	-0.16	0.02	-0.01	-0.03	1.00

POC

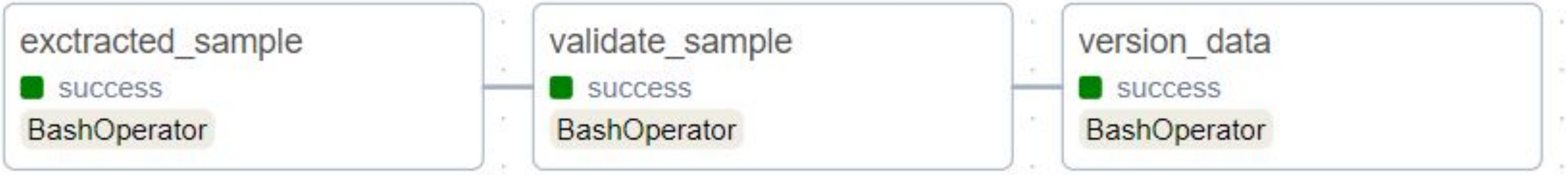
- Simple Linear Regression with our full preprocessing pipeline
- **RMSE**: 0.2633

Data preparation



Airflow Pipeline

Extract data pipeline

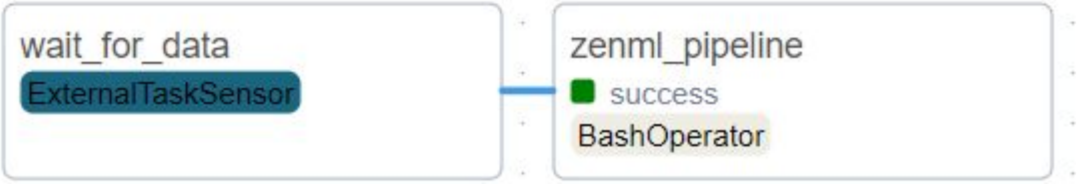


exctracted_sample
🟩 success
BashOperator

validate_sample
🟩 success
BashOperator

version_data
🟩 success
BashOperator

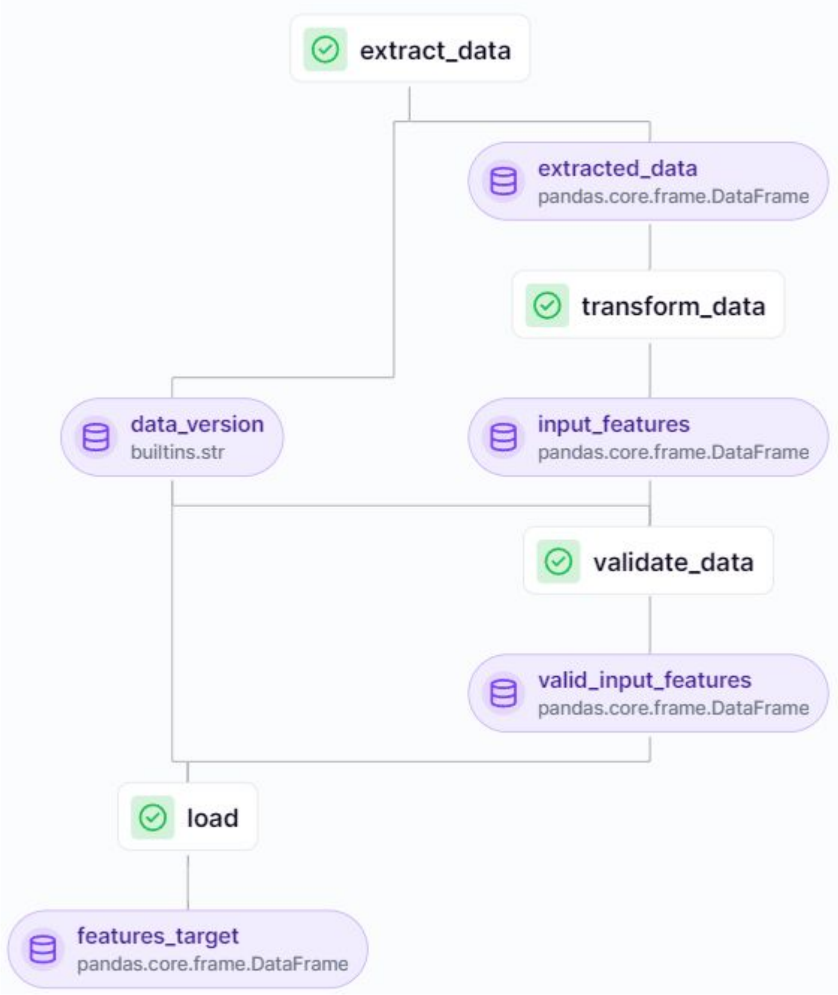
Preprocess data pipeline



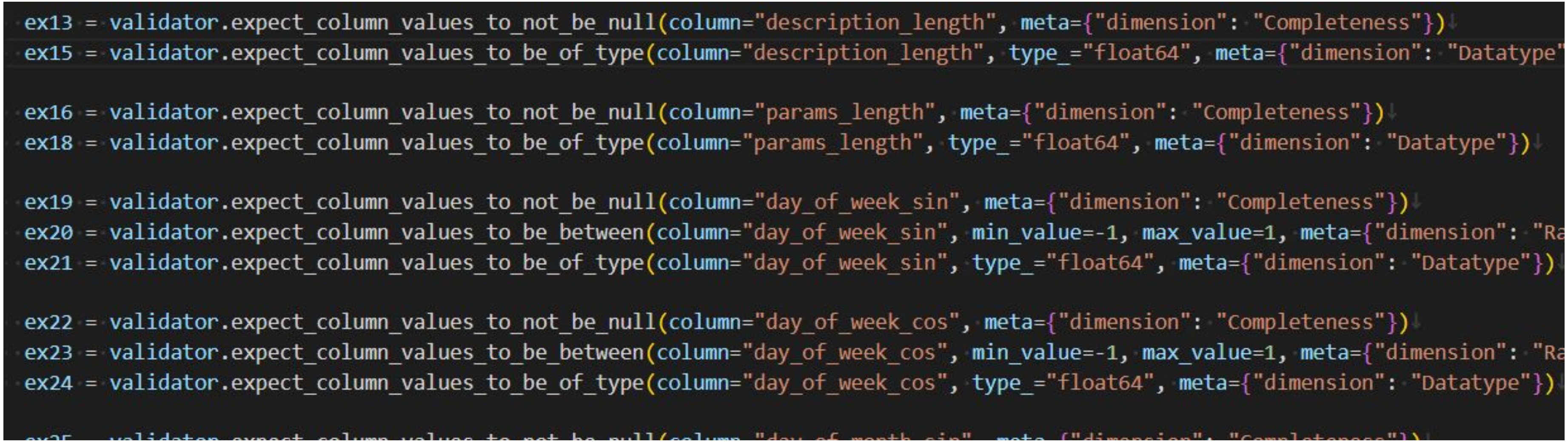
wait_for_data
ExternalTaskSensor

zenml_pipeline
🟩 success
BashOperator

ZenML Pipeline

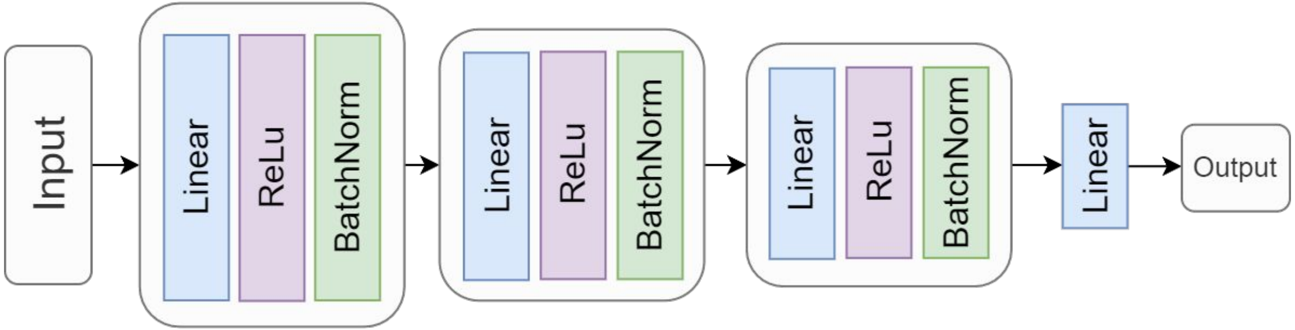


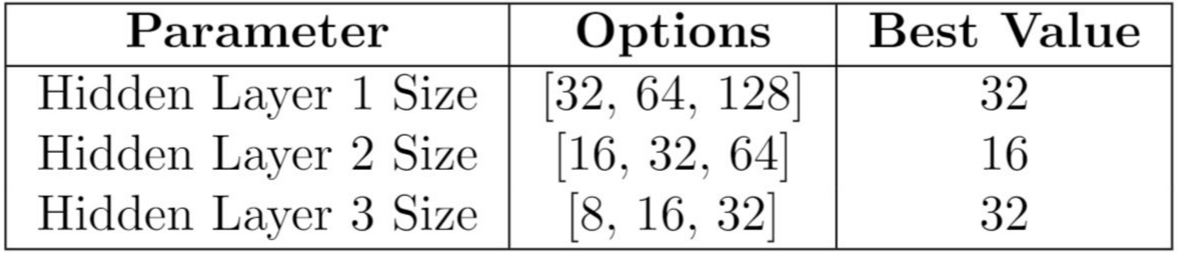
Expectations



```
ex13 = validator.expect_column_values_to_not_be_null(column="description_length", meta={"dimension": "Completeness"})
ex15 = validator.expect_column_values_to_be_of_type(column="description_length", type_="float64", meta={"dimension": "Datatype"

ex16 = validator.expect_column_values_to_not_be_null(column="params_length", meta={"dimension": "Completeness"})
ex18 = validator.expect_column_values_to_be_of_type(column="params_length", type_="float64", meta={"dimension": "Datatype"})

ex19 = validator.expect_column_values_to_not_be_null(column="day_of_week_sin", meta={"dimension": "Completeness"})
ex20 = validator.expect_column_values_to_be_between(column="day_of_week_sin", min_value=-1, max_value=1, meta={"dimension": "Ra
ex21 = validator.expect_column_values_to_be_of_type(column="day_of_week_sin", type_="float64", meta={"dimension": "Datatype"})

ex22 = validator.expect_column_values_to_not_be_null(column="day_of_week_cos", meta={"dimension": "Completeness"})
ex23 = validator.expect_column_values_to_be_between(column="day_of_week_cos", min_value=-1, max_value=1, meta={"dimension": "Ra
ex24 = validator.expect_column_values_to_be_of_type(column="day_of_week_cos", type_="float64", meta={"dimension": "Datatype"})

ex25 = validator.expect_column_values_to_not_be_null(column="day_of_month_sin", meta={"dimension": "Completeness"})
```

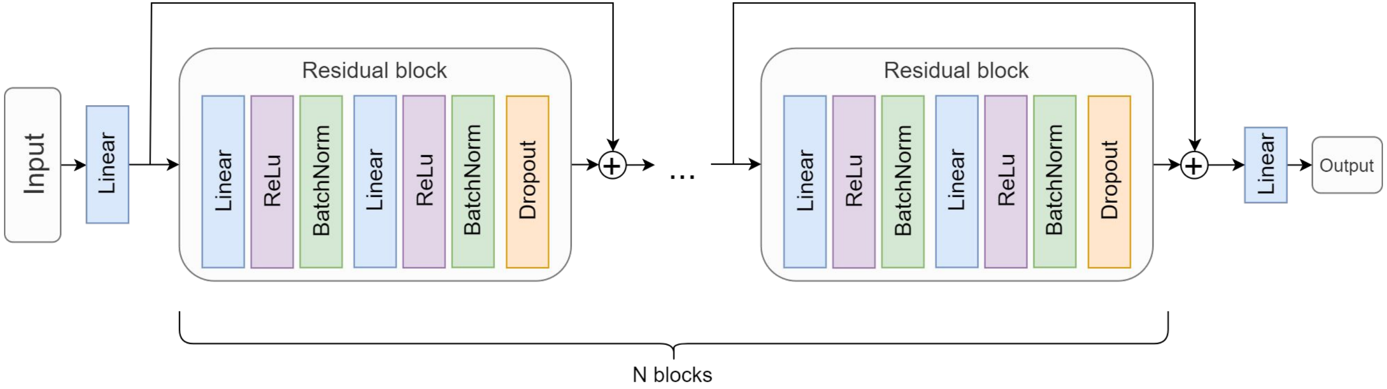
Model engineering - MLP





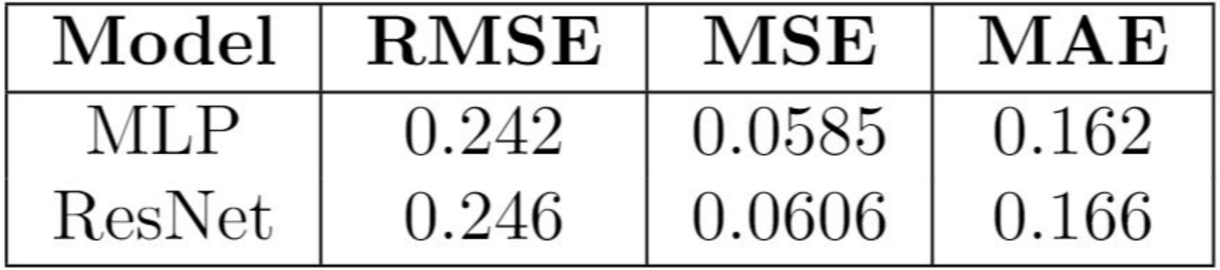
Parameter	Options	Best Value
Hidden Layer 1 Size	[32, 64, 128]	32
Hidden Layer 2 Size	[16, 32, 64]	16
Hidden Layer 3 Size	[8, 16, 32]	32

Model engineering - ResNet



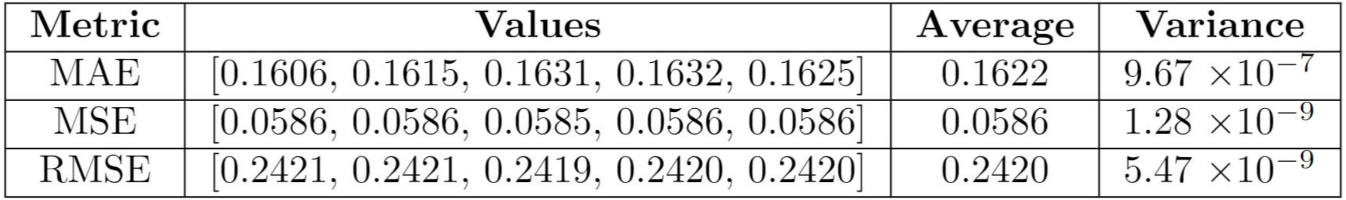
Parameter	Options	Best Value
Embedding Size	[16, 32, 64, 128]	128
Number of Residual Blocks	[1, 3, 5, 10]	3
Dropout Rate	[0.35, 0.5, 0.75]	0.75

Model Performance



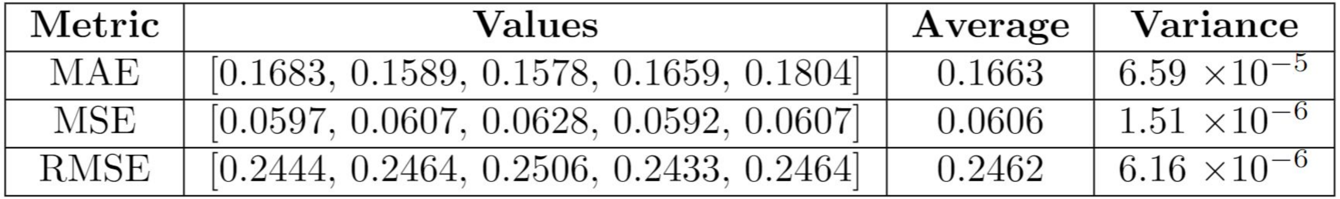
Model	RMSE	MSE	MAE
MLP	0.242	0.0585	0.162
ResNet	0.246	0.0606	0.166

Result Reproducibility



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}

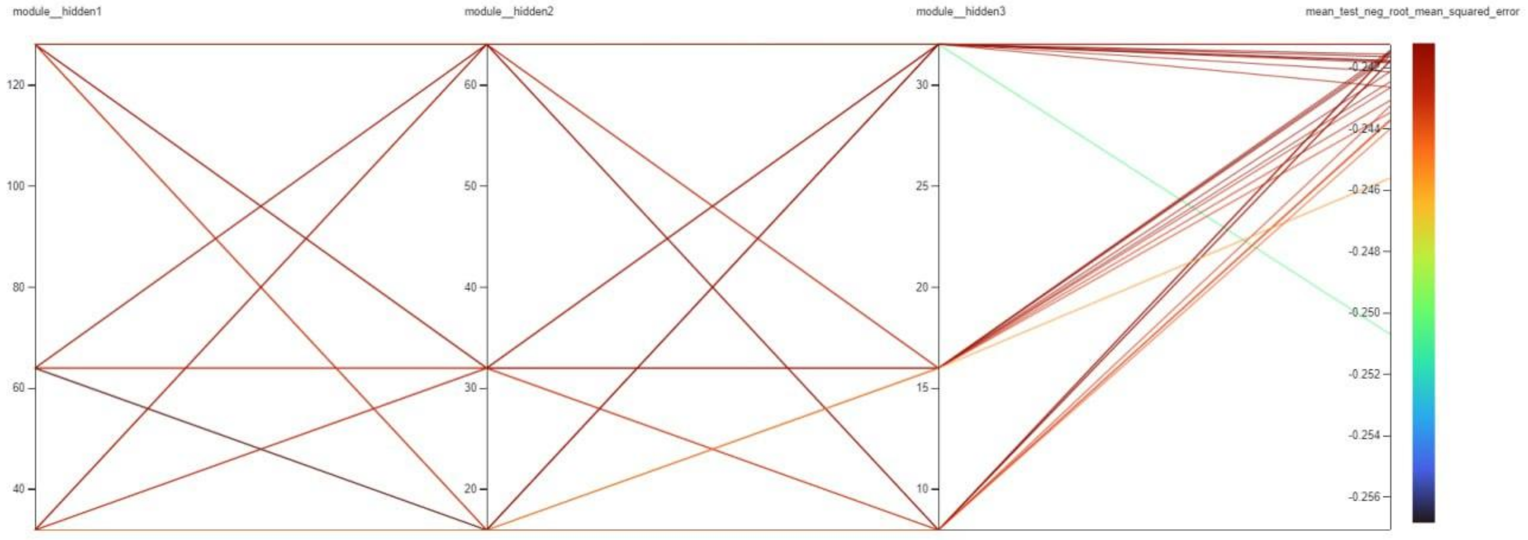
Table 4: MLP Test Metrics for Different Seeds



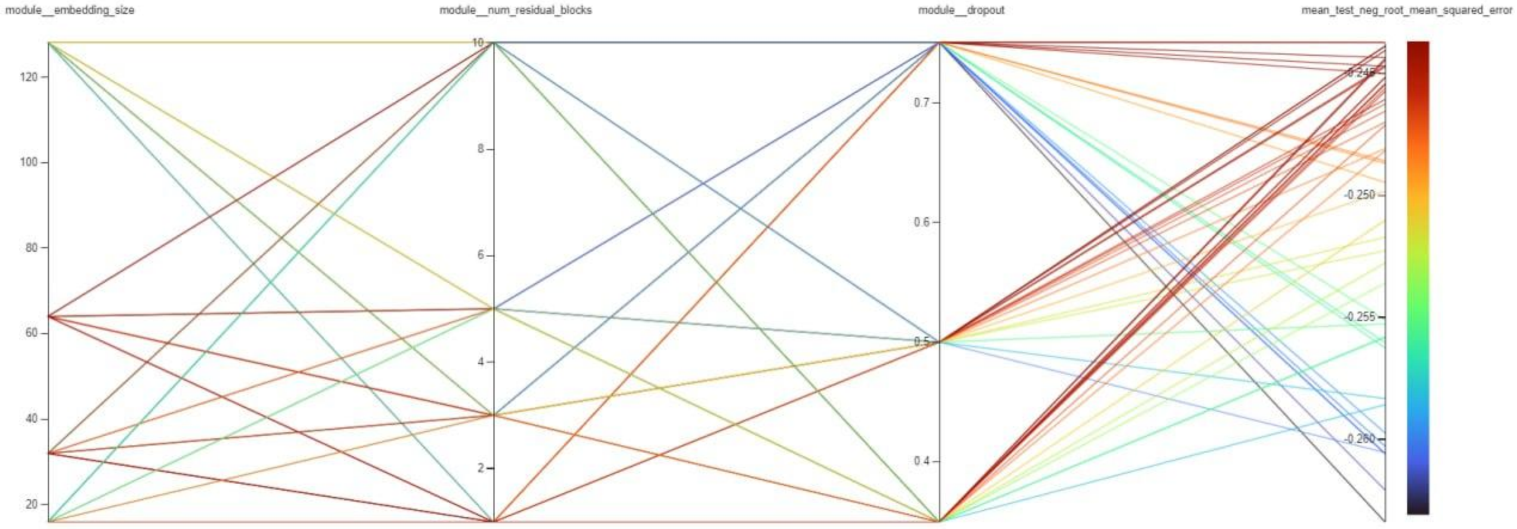
Metric	Values	Average	Variance
MAE	[0.1683, 0.1589, 0.1578, 0.1659, 0.1804]	0.1663	6.59×10^{-5}
MSE	[0.0597, 0.0607, 0.0628, 0.0592, 0.0607]	0.0606	1.51×10^{-6}
RMSE	[0.2444, 0.2464, 0.2506, 0.2433, 0.2464]	0.2462	6.16×10^{-6}

Table 5: ResNet Test Metrics for Different Seeds

MLP - Parallel Plot



ResNet - Parallel Plot



Model validation

- **Performance Test:** RMSE with a threshold of 0.25
- **Success Criteria:** Achieve an RMSE below 0.25 on unseen data.
- **Result:** The MLP model successfully met the validation criteria.
- **Decision:** Proceed with model deployment.

Validation report of the champion model

https://drive.google.com/file/d/1a9NFoW3XWYYuDmQSyJDjCMXN_W-w-K3a/view?usp=sharing

Model deployment

PyTorch SKORCH

Deployment Options:

- **Docker:** Packaged the model and dependencies into a Docker image.
- **Flask API:** Developed a custom API for more direct and localized deployment.

User Interface:

- **Gradio:** Built an intuitive GUI for users to input ad details and receive demand predictions.

Inference Hardware:

- **Model Size:** 100 Kb
- **Hardware tested:** CPU: Intel Core i5, RAM: 16GB

Project plan

https://sharing.clickup.com/9012093001/g/h/8cjk829-412/85b8479c837f6cd

Project collaboration

Data engineer: Ildar Zalialiev

- Setup extraction pipeline
- Setup preparation pipeline using ZenML and Airflow
- Tests and Expectations

Data scientist: Artur Zagitov

- EDA
- Build data preprocessing pipeline
- Build and train models

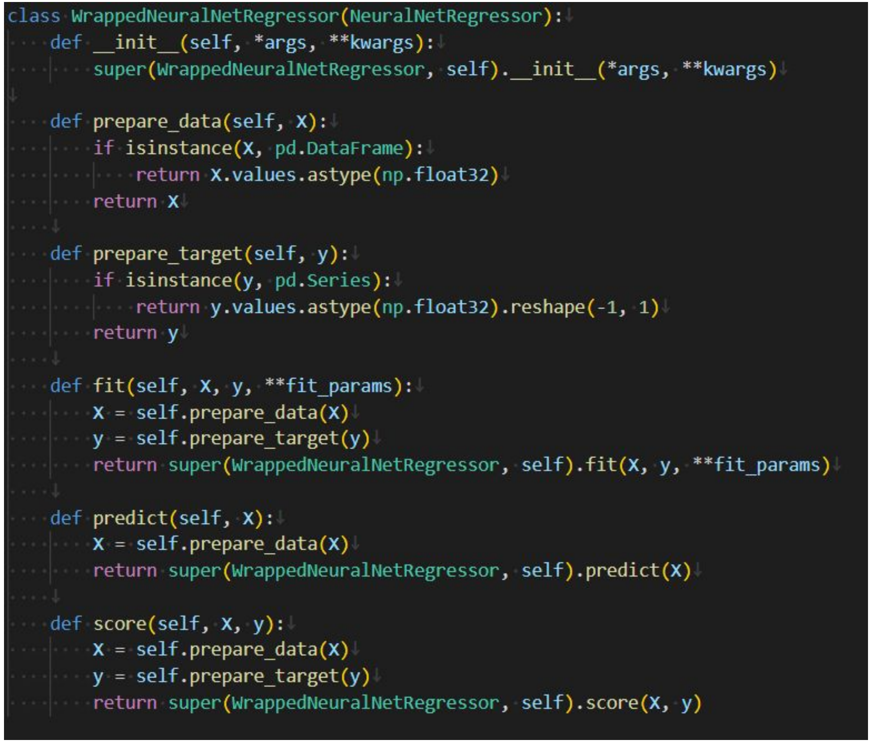
ML engineer: Artem Nazarov

- Model validation
- Model deployment
- Gradio UI

Tasks	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
Ildar Zalialiev	50%	50%	20%	20%	10%
Artur Zagitov	40%	40%	40%	30%	20%
Artem Nazarov	10%	10%	40%	50%	70%
Total	1	1	1	1	1

Challenges and lessons learned

- **Main technical challenge:** Deploying Pytorch+Skorch model to Docker
- Dependencies



```python
class WrappedNeuralNetRegressor(NeuralNetRegressor):
    def __init__(self, *args, **kwargs):
        super(WrappedNeuralNetRegressor, self).__init__(*args, **kwargs)

    def prepare_data(self, X):
        if isinstance(X, pd.DataFrame):
            return X.values.astype(np.float32)
        return X

    def prepare_target(self, y):
        if isinstance(y, pd.Series):
            return y.values.astype(np.float32).reshape(-1, 1)
        return y

    def fit(self, X, y, **fit_params):
        X = self.prepare_data(X)
        y = self.prepare_target(y)
        return super(WrappedNeuralNetRegressor, self).fit(X, y, **fit_params)

    def predict(self, X):
        X = self.prepare_data(X)
        return super(WrappedNeuralNetRegressor, self).predict(X)

    def score(self, X, y):
        X = self.prepare_data(X)
        y = self.prepare_target(y)
        return super(WrappedNeuralNetRegressor, self).score(X, y)
```

Demo