Multimodality and Stacking Ensemble Models in Demand Prediction

Project Report for 15.095 at MIT in Fall 2024

Ellie Yang
ellie_y@mit.edu

Steve Zhou
stevez02@mit.edu

Abstract

Classified advertisement platforms, such as Avito, serve as crucial facilitators of online commerce, connecting millions of users daily. This project aims to predict the likelihood of a successful deal by leveraging multi-modal data from Avito's dataset, which includes tabular, text, and image data. Using machine learning models such as LightGBM and ensemble techniques, we explored the integration of diverse data modalities to improve predictive accuracy. Advanced feature engineering, including embeddings from NLP models like FastText, SpaCy, and TFI-DF, and image embeddings from ResNet50, combined with techniques such as stacking and Bayesian hyperparameter tuning, demonstrated the complementary power of different data types. The best-performing model, which integrated tabular, text embeddings, and image embeddings, demonstrated a significant improvement in predictive accuracy, highlighting the effectiveness of leveraging multi-modal data for demand prediction in dynamic online marketplaces.

1. Introduction

Classified advertisement platforms play a crucial role in facilitating online commerce, serving millions of users who list and purchase items daily. However, the success of a product listing on such platforms depends on a variety of factors, from descriptive titles and appealing images to competitive pricing and user location. It remains a crucial challenge for both the platforms and the sellers to predict the likelihood of a listing's success in such dynamic marketplaces.

In this project, we aim to develop a predictive model for the probability of a successful deal using an integrative approach that leverages the diverse data modalities associated with each listing, using a dataset provided by Avito.com, Russia's largest classified advertisement website. This project explores the potential of multi-modal machine learning to understand the factors influencing deal probability and consumer demand in general. We also experimented with boosting and stacking ensembles to improve the accuracy of our models further.

Key challenges include handling tabular data of predominantly categorical features, feature extraction from Russian texts, and handling large-scale image data. Our methodology involves feature engineering at different levels, advanced NLP models for large-scale applications, and transfer learning using state-of-art computer vision neural networks. Moreover, we automated the hyperparameter tuning of different models using Optuna's Bayesian optimization framework to ensure efficient and fair comparison of the models.

The results of this project can offer sellers actionable guidance on optimizing their listings and provide Avito with advice to improve user experience. It also paves the way for future research into multi-modal demand prediction in other online marketplaces. As such dynamic online marketplaces gain increasing popularity in the global economy, our work will facilitate fair and efficient transactions, contributing to the vision of frictionless markets.

2. Data

The dataset includes over 1.5 million listings from Avito.com between March 14 and 27, 2017. A listing is identified by a unique item ID, and Avito provides tabular information about the listed item and the listing user. In addition, the dataset also contains the title and description of each listing, as well as the primary image of the listing, if exists. We randomly partitioned the entire dataset into a 70-30 split, where 70% of the data is used for training and 30% is used for testing our analysis models.

The target variable in this dataset is "deal_probability", which represents the likelihood that a given listing will be sold. This variable is a continuous value ranging from 0 to 1, where a higher value represents a higher probability of deal. Avito's internal model provides this probability, and the goal of our analysis model is to minimize the root mean squared error of our predictions in the testing dataset.

2.1. Tabular Data

The tabular data includes a variety of structured features that we group into two main categories: product-specific attributes and user-related attributes.

Product-specific attributes include variables such as “parent category name” and “category name”, which describe two levels of product categories defined in Avito’s advertisement business, “param_1, _2, _3”, which defines three category-specific parameters of the product, and “price”, which is the listing’s monetary value. Additionally, the “region” and “city” of the listing are included to capture the geographic location of the listing, which may affect buyer interest due to regional variation in demand as well as the practicability of in-person transactions.

User-related attributes include features like “user ID”, “user type”, which distinguishes between a private seller and a corporate seller, and “item seq number”, which measures the listing frequency of the seller. For transparency and interpretability, we removed features for which Avito did not provide documentation. The raw dataset contains a total of 12 tabular features.

Of the tabular features, most are categorical except the “price” and “item seq number”, necessitating the engineering of new features and the user of appropriate label encoders before fitting the model. Our exploratory data analysis ensured that there were no missing values in price or the response variable.

2.2. Text Data

The text data in the Avito dataset consists of two key fields: the listing’s title and description. These fields provide important unstructured information about each listing. The title typically contains a brief product summary in a few words, while the description provides more detailed information in a few sentences. We believe an ideal model should be able to gain insights into the product features and how well the sellers present their products from these texts. For example, listings with titles or descriptions that include keywords such as “new” or “urgent sale” can potentially solicit higher demand.

An initial review of the text data reveals considerable variability in the length and quality of the content, especially for the detailed description, which ranges from a few words to numerous detailed paragraphs. Moreover, both fields are written in Russian, which introduces a language-specific challenge for data preprocessing and analysis. This language barrier limits our choice of Natural Language Processing (NLP) techniques to only those trained specifically for Russian corpus or with multilingualism. We will discuss the selection and comparison of different models in the following sections.

2.3. Image Data

The Image data in the Avito dataset consists of raw product images uploaded by the sellers. Each listing is associated with up to one image. These images serve as a critical visual representation of the listing item, potentially influencing buyer interest. Since only 7% of listings lack an associated image, we believe extracting image features is essential for predicting deal probability. However, image data presents challenges in computational power.. The dataset includes images of varying sizes, orientations, and qualities, with a total size exceeding 50 GB on disk.

We thus randomly select 20% of listings in the training and testing set, focusing only on those with images, as a subset to evaluate image features. This subset provides a manageable data size to cope with the computational constraints while preserving the diversity of the original dataset. Considering the limitation in computational resources, we use pre-trained lightweight computer vision models to capture visual features of the associated features.

The reduction in sample size can introduce uncertain and undesirable effects on model performance. To ensure a fair comparison between models leveraging image data and those relying on tabular and text features, we train and tune the tabular and text-based models using the same subset of the training data that was created for image feature extraction. The models of different modalities are then evaluated using the same subset of testing data. This approach enables us to directly assess the contribution of image features to the predictive power of our models while avoiding potential biases arising from differences in dataset composition.

3. Methods

We selected LightGBM as the default model for this project. It is a gradient boosting method that offers faster training time and lower memory usage compared to traditional boosting methods. Its computational efficiency and predictive power make it well-suited for the vast size of the Avito dataset. Moreover, LightGBM is adaptable to different feature types. The Python package has native support for categorical variables and missing value handling, making it a natural choice for the tabular data in our project.

LightGBM offers an array of hyperparameters to be tuned, which gives us control over model complexity but also poses challenges in tuning. We employ Optuna, a Python package implementing Bayesian optimization frameworks for hyperparameter tuning. It uses a probabilistic approach to tune efficiently based on prior trials. It not only improves the model's predictive power but also significantly reduces the computational overhead compared to an exhaustive grid search. Using LightGBMs tuned by Optuna as the default predictive model offers a foundation for comparing the contribution of additional modalities.

3.1. Feature Engineering

We constructed different kinds of features to capture the characteristics of our data. By feature selection, we kept 34 tabular features to be put into future models, including 9 original features and 25 newly engineered features. Firstly, we created various combination features, such as region_city (combining region and city), and features that incorporate category and user information like region_category_user. Regarding the price variable, we performed a logarithmic transformation, binning, and constructed price-related statistical features at the category and city levels, for example, category_price_mean, to capture the distributional characteristics of prices. For text features, we extracted the lengths of titles and descriptions, word counts, keyword presence indicators, as well as counts of digits and newline characters. We also added a missing value indicator feature to capture the completeness of the description field. The full list of tabular features can be seen in the Appendix 1.

3.2. Text Feature Extraction

Modern NLP models represent features extracted from texts using embeddings, which are dense vectors that encode the essential features of the unstructured data into a continuous vector space. For text features, we explore both the traditional frequency-based representation and the output embeddings by neural-network-based models, which encode linguistic nuances and semantic relationships into numeric features that LightGBM can easily incorporate.

Despite their richness, embeddings can still result in high-dimensional feature space, depending on the specific model. High dimensionality increases the computational requirements and increases the risk of overfitting. Thus, we apply Singular Value Decomposition (SVD). SVD aims to retain the most critical information using a smaller collection of orthogonal components, which are then used as numeric features. For each model, we manually choose the output dimensionality by inspecting how the ratio of variance explained by SVD features changes as the number of output dimensions changes.

We concatenate titles and descriptions of each listing as a single corpus of text to analyze. We begin with Term Frequency-Inverse Document Frequency (TF-IDF), a frequency-based encoding method. It outputs a sparse vector representation of the text features. For practicability, we selected only the top 500 Russian words in our data that have the highest TF-IDF score to prevent memory overflow.

The second model is SpaCy's pre-trained Russian models. SpaCy is a popular library that provides pre-trained models for multiple NLP tasks in different languages. We experiment with their small and medium Russian models, which output contextual embeddings of lengths of 98 and 300, respectively.

The third model is FastText, a library developed by Facebook AI Research, known for its ability to create robust word embeddings by incorporating subword information. This approach is particularly advantageous for morphologically rich languages like Russian, where words often exhibit complex inflectional patterns and derivations. In our analysis, we built a FastText model trained on tokenized titles and descriptions, configured with a vector size of 100 and a context window of 5. FastText also pre-trained word vectors for Russian based on Common Crawl and Wikipedia, and we also experimented with this pre-trained model.

3.3. Image Feature Extraction

To extract meaningful visual features from images, we use a pre-trained headless ResNet 50 model to generate embeddings of length 2048 for associated images. ResNet is a convolutional neural network widely recognized for its performance in image recognition tasks. ResNet 50 is a relatively small version that we can compute for our subset in a reasonable time. These embeddings offer a compact, high-level feature representation for each image. Similar to text embeddings, we applied SVD on image embeddings to acquire a set of numeric features that can be incorporated by LightGBM.

3.4. Stacking Ensemble

We employ a stacking ensemble approach to further improve the predictive power of our models. Stacking is a method that combines predictions from multiple first-stage base models by using them as inputs to a second-stage meta-model, which then learns to optimize the final prediction. In theory, this approach leverages the strengths of individual models while mitigating their weaknesses. We empirically assess this approach using different choices of first-stage response variables.

4. Results

In this section, we present the performance of our models. We started with using text embeddings as additional features to boost tabular trees, following a systematic workflow. The process began with the generation of embeddings from text data with the model to be tested, and then, we apply SVD to manage dimensionality. The reduced embeddings were concatenated with the tabular features to create a unified dataframe, which was then used to train a LightGBM regressor, with hyperparameter tuning performed by Optuna. The model is then evaluated on the test data to get the out-of-sample RMSE.

4.1. Stacking models

To enhance prediction performance, we experimented with two stacking models. Both models leverage a two-stage stacking architecture where predictions from first-stage models are used as inputs to a meta-model. Each stacking model is designed to combine information from text embeddings (TF-IDF, SpaCy, and FastText) with tabular features for a robust prediction pipeline. To prevent information leakage during training and ensure reliable evaluation, we employed a 5-fold cross-validation strategy throughout the stacking process.

In Stacking Model 1, we created three first-stage models, each utilizing one of the text embedding methods to predict the deal_probability. The embeddings include TF-IDF with 5 SVD, SpacySmall with 40 SVD, and FastText with 42 SVD. Separate LightGBM models were trained on each type of embedding (Model 1,2 & 3 in the Figure). Each model outputs a prediction of deal_probability. Then for the meta model, we again utilized a LightGBM model. The meta model combines the predictions from the first-stage models along with tabular features to produce the final prediction of deal_probability.

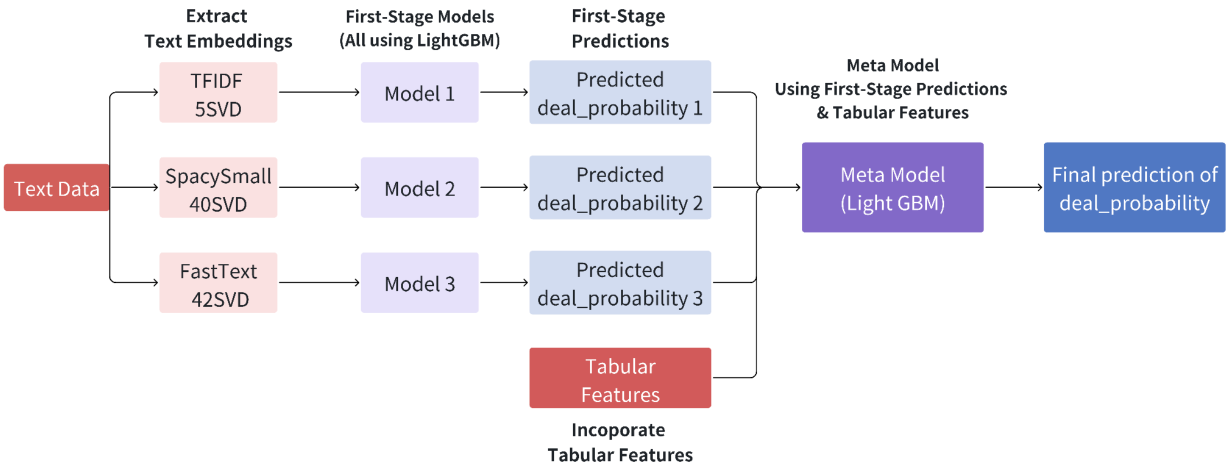

Figure 4.1. Illustration of Stacking Model 1

The second stacking model introduces a variation by including an auxiliary target in the first stage. The same text embedding extractions as Stacking Model 1 were applied, but in the first-stage only two separate LightGBM models were trained (Model 1 & Model 2 in the Figure). The first model predicts price using FastText SVD 42 embeddings, while the second model predicts deal_probability using

SpacySmall SVD 40 embedding. The meta model combines first-stage predictions (price and deal_probability), tabular features, and TF-IDF SVD 5 embeddings and utilizes LightGBM to predict the final deal_probability.

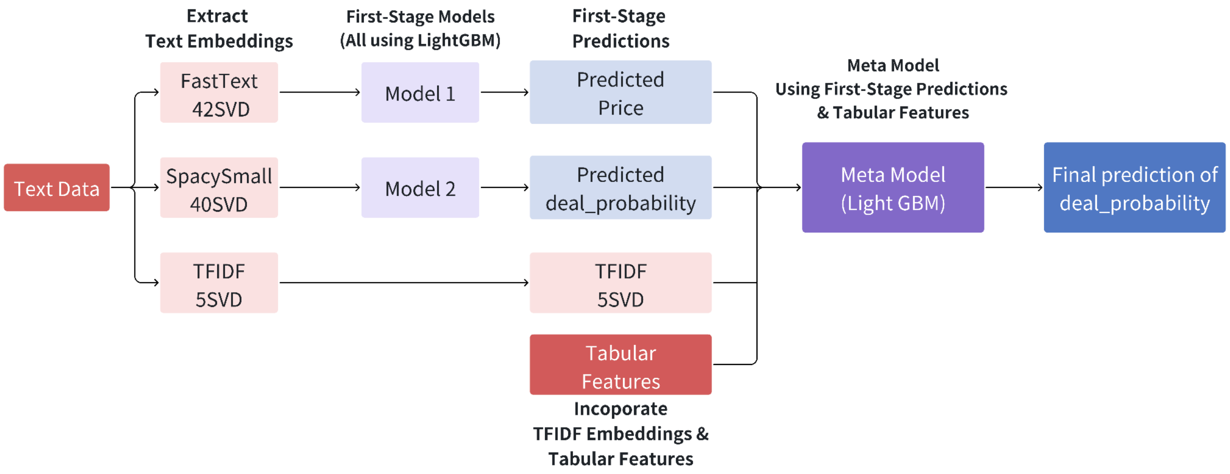

Figure 4.2. Illustration of Stacking Model 2

4.2. Accuracies Comparison

As the first step, we extracted embeddings using various language models, applied SVD to reduce their dimensionality, and integrated the reduced embeddings with tabular features into our final dataset. To evaluate their effectiveness, we recorded the RMSE at each stage of embedding utilization. A detailed performance summary is provided in the Appendix 2. Below, we highlight the performance of selected models to showcase and compare the results.

The table compares the performance of various models in terms of RMSE. The Tabular Baseline achieves an RMSE of 0.227894, serving as a benchmark.

Models using only text embeddings without tabular features showed significantly higher RMSEs compared to the Tabular Baseline, highlighting the critical role of tabular features in achieving better predictions. Among these models, our custom FastText embeddings outperformed others, achieving the lowest RMSE of 0.235021, closely followed by the FastText Pretrained model with an RMSE of 0.235337. Notably, the FastText family consistently outperformed other embedding methods by a margin of at least 0.01 in RMSE.

When text embeddings were combined with tabular features, RMSEs dropped significantly, emphasizing the complementary nature of embeddings and tabular data in improving model performance. Among these models, our custom FastText achieved the best performance with an RMSE of 0.226067, outperforming other models by at least 0.01 and surpassing the Tabular Baseline RMSE of 0.227894.

The stacking models significantly improved compared to Tabular Baseline and the embedding-only models. In Stacking Model 1, where first-stage deal_probability was predicted using SpaCy, TF-IDF and FastText, and combined with tabular features in the second stage, achieved an RMSE of 0.226904. While it did not surpass the best-performing model that directly combined FastText embeddings with tabular features (0.226067 RMSE), it outperformed all other models, highlighting its value.

Table 4.1: Summary of Text and Tabular Models Performance in Complete Dataset

Tabular Baseline	.227894

RMSEs when Directly Using Embedding as Features for LightGBM

	SVD Dimension	RMSE - Embedding Only	RMSE - With Tabular
TF-IDF	5	.252066	.227283
SpaCy Small	40	.248780	.227110
SpaCy Medium	100	.246289	.227009
FastText	42	.235021	**.226067**
FastText Pretrained	50	.235337	.227278

RMSEs when Stacking Embedding LightGBM

First-stage Models	Second-stage Model	RMSE
deal_probability ~ SpaCy deal_probability ~ TF-IDF deal_probability ~ FastText	First-stage Preds + Tabular	.226904
price ~ FastText deal_probability ~ SpaCy	First-stage Preds + Tabular + TD-IDF	.227119

To incorporate image embeddings, we use a subset of the data due to computational constraints. As a result, we re-train models on these subsets to account for any potential performance loss due to the smaller sample size. This ensures a fair comparison of their performances on the subset testing data to evaluate the impact of adding image features. The Tabular Only model, which uses engineered features, achieves an RMSE of 0.232013, serving as the baseline. Adding FastText embeddings (not pre-trained) with SVD-reduced dimensions (42) improves the model when combined with tabular features, achieving an RMSE of 0.230863, significantly better than using embeddings alone (0.241487), underscoring the importance of integrating tabular data.

Similarly, incorporating image features extracted with ResNet50, reduced to 500 dimensions using SVD, achieves an RMSE of 0.241487 when used alone. However, combining image features with tabular data results in an improved RMSE of 0.230489, again demonstrating the complementary nature of different data types.

The best performance is achieved with the Tri-modality model, which combines tabular features, FastText embeddings (SVD 42), and image features (SVD 500). This configuration achieves the lowest RMSE of 0.229803, outperforming all other models and demonstrating the value of integrating multiple data modalities for robust and accurate predictions. This highlights that leveraging diverse information sources—text, images, and tabular features—can significantly enhance model performance.

Table 4.2: Summary of Performance in Subset Data for Image Embedding

	SVD dimension	RMSE - Embedding Only	RMSE - With Tabular
Tabular Baseline	-	-	.232013
Text (FastText)	42	.241487	.230863
Image (ResNet50)	500	.242552	. 230489
Tri-modality: Text + Image	-	-	**.229803**

5. Conclusion and Future

The results we present in this project highlight the effectiveness of multi-modal machine learning models in predicting the likelihood of success of product listings on Avito.com. By combining tabular features with unstructured text and image data, we demonstrated the complementary power of these modalities to enhance model accuracy. Through text embeddings generated by various NLP models, we capture the semantic features of text, and through image feature extraction with ResNet50, we abstract the image data into high-level vectors. Adding these features to LightGBM proved instrumental in deciphering the complex dynamics influencing deal probability. Furthermore, the stacking ensemble methods also demonstrated power in improving prediction accuracy, testifying to the potential of ensemble approaches in leveraging information from different sources.

We also propose several ways in which our approach can be further boosted for future work. Firstly, future researchers may want to expand feature engineering. While we focused on a carefully engineered set of features in this project, there remains untapped potential for exploring additional features that could potentially improve model performance. Secondly, we recognize the vast possibilities in stacking methods that we did not have resources to explore. Our current stacking approach was limited to leveraging text embeddings as first-stage inputs. Future work can involve exploring stacking strategies that incorporate image embeddings alongside text embeddings in the first stage. Stacking more stages or using other intermediate response variables for previous-stage models are also fields to be explored.

Lastly, scaling image analysis is an obvious direction for future work. Due to computational constraints, we restricted our analysis of image features to a subset of the dataset. Extending this analysis to the entire dataset will provide a more comprehensive understanding of the role of visual features in predicting deal probabilities.

By addressing these directions, future work pushes the boundaries of multi-modal learning, uncovering new insights and improving the predictive capabilities of models for online marketplaces. In this way, we can provide more actionable insights for sellers and platforms to optimize listings and improve user experiences.

Contributions

With regard to the technical tasks, both members made initial explorations in feature engineering and the selection of the baseline model. Ellie Yang finalized the complete set of tabular features and the baseline LightGBM model. She also performed experiments in building predictive models from TF-IDF and FastText embeddings. Steve Zhou performed experiments with SpaCy text embeddings and led the effort in image feature extraction using ResNet50. The effort in stacking models was initiated by Steve Zhou, and Ellie Yang performed experiments with different stacking designs presented here in the report.

With regards to the presentation and report, Steve Zhou drafted the Introduction, Data, and Methodology sections of the report as well as the tables, while Ellie Yang drafted the Results and Conclusion sections, as well as the data visualizations. The presentation and the slide deck are split into feature engineering, text embedding, and stacking models and results, which Ellie Yang completed, and secondly, introduction, data overview, image embedding, and conclusions that Steve Zhou compiled.

Appendix 1: Tabular Features and Engineering

Original Features	Price Statistical Features	Text-Related Features	Combination Features
region	category_price_mean	title_length	region_city
city	category_price_std	description_length	all_category
parent_category_name	category_price_skew	title_word_count	category_param_1
category_name	city_price_mean	description_word_count	region_category_user
param_1	city_price_max	title_has_keyword	city_category_user
param_2	city_price_skew	description_has_keyword	
param_3	price_to_category_mean	title_digit_count	
price	price_to_category_max	description_digit_count	
user_type	price_log	description_newline_count	
	price_bin	description_missing	
TOTAL: 9	TOTAL: 10	TOTAL: 10	TOTAL: 5
TOTAL: 34			

Appendix 2: Detailed Table of Text and Tabular Models

Model Type	**Configuration**	**RMSE**
Tabular Only	With Engineered Features	0.227894
	Stacked Engineered	0.227831
SpaCy Small Russian Embeddings	Text Embeddings Only	0.248780
	Tabular + Embedding	0.226968
	Tabular + Embedding SVD 40	0.227110
SpaCy Medium Russian Embeddings	Text Embeddings Only	0.246289
	Tabular + Embedding	0.226827
	Tabular + Embedding SVD 100	0.227009
TFIDF Embeddings	Text Embeddings Only	0.252066
	Embedding SVD 5	0.252892
	Tabular + Embedding SVD 5	0.227283
FastText Embeddings (not pretrained)	Text Embeddings Only	0.235021
	Embedding SVD 42	0.239110
	Tabular + Embedding SVD 42	**0.226067**
FastText Embeddings (pretrained)	Text Embeddings Only	0.235337
	Embedding SVD 50	0.263487
	Tabular + Embedding SVD 50	0.227278
Stacking	3 First Stage Model (SpaCy + TFIDF + FastText) + Tabular	0.226904
	2 First Stage Model (FastText + SpaCy)+Tabular+TFIDF	0.227119

Appendix 3: Visualization of Model Performance

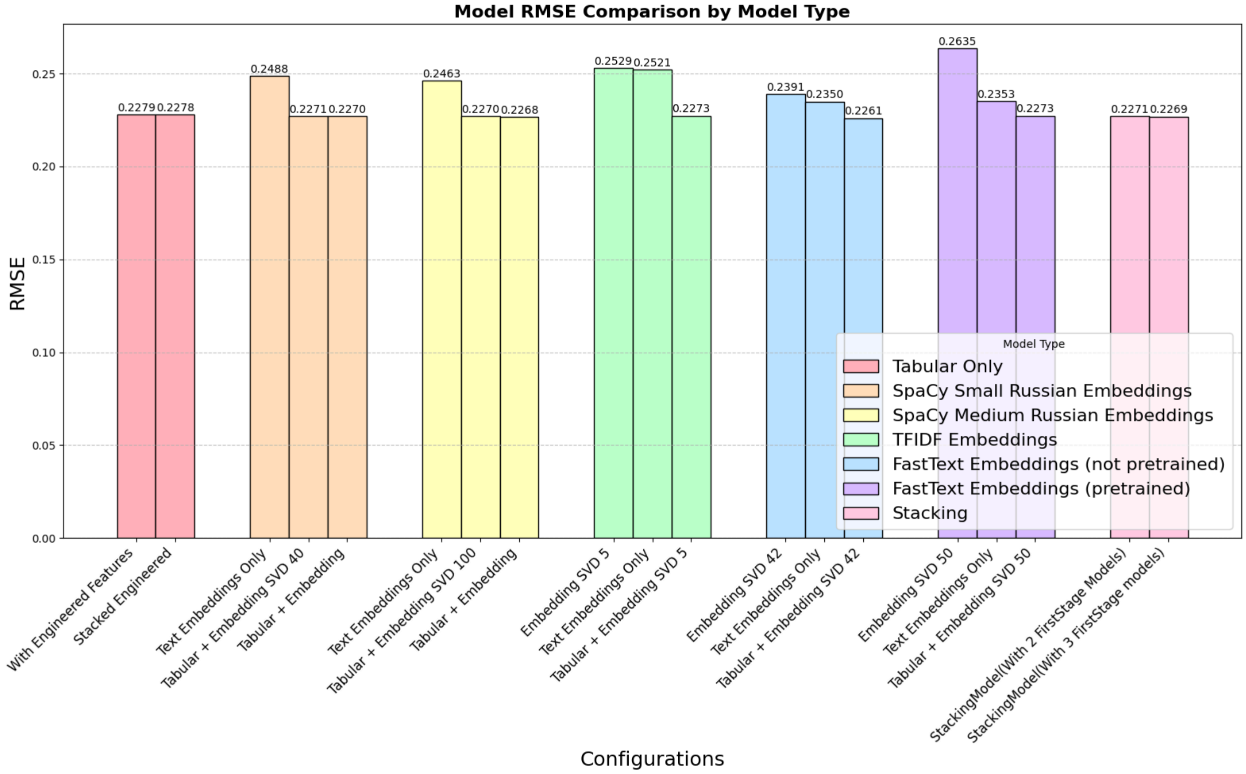

Figure 1. Model RMSE by Model Type (Tabular & Text) in Complete Dataset

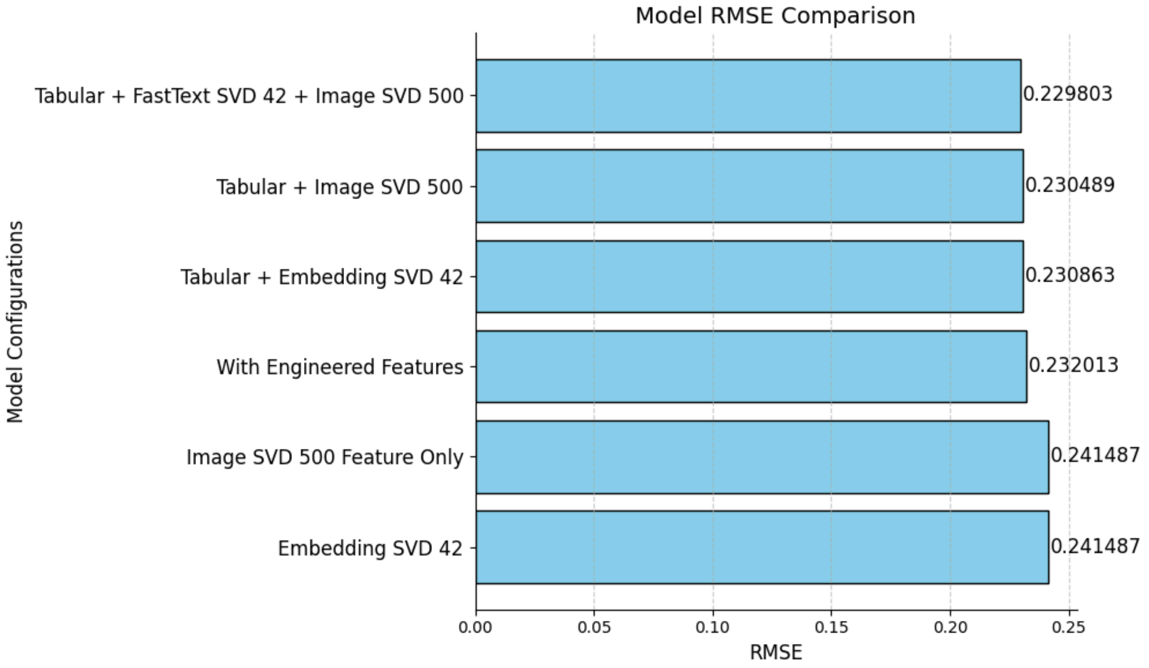

Figure 2. Model RMSE in Subset Data for Image Processing (Tabular, Text, Image)