Assignment 3: Final Report

Reinier Bos & Diego Sourbag
s3703177 & s3699420

1. Task Definition

The Home Depot search relevance task involves predicting how relevant a product is to a search query of a user. The goal is for us to train a model that gives a relevance score based on how well the product title and description match the search.

2. Data Exploration

We explored the Home Depot search relevance dataset to answer key questions and prepare for feature engineering. There are **54,667** unique products (distinct `product_uid` values) in the training data. The two most frequent products are 102893 and 101959, each being present a number of **21** times. The mean of the relevant scores is **2.382**, the median is **2.330**, and the standard deviation is **0.534**. **Histogram and boxplot of relevance distribution:**

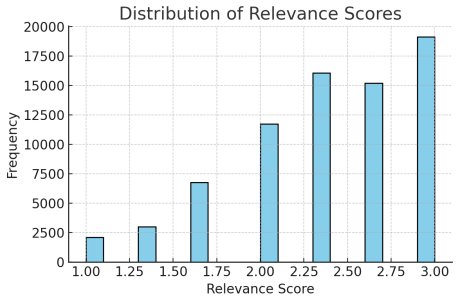

Figure 1: Histogram of relevance scores

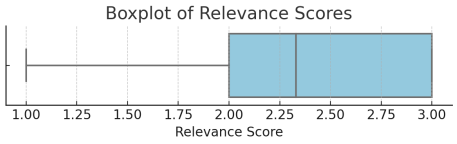

Figure 2: Boxplot of relevance scores

These plots show that relevance values are skewed towards higher scores, with most values between 2.0 and 3.0.

Top-5 most frequent brand names (from `attributes.csv`): The most frequent brand is the Unbranded products with 2,954 ocurrences. This is followed by Hampton Bay (1,723), KOHLER (1,389), Everbilt (1,381), and Home Decorators Collection (1,275).

3. Baseline Method

The baseline model implemented by Yao-Jen Chang. First, during **data preprocessing**, all text fields are converted to lowercase and stemmed using the Porter stemer from the NLTK library. For **feature engineering**, we used a small number of simple and interpretable features, including product title and product description.

Model: A Random Forest Regressor from scikit-learn is trained on the generated features. The model is trained on 80% of the data and evaluated on the remaining 20%. The evaluation metric is Root Mean Squared Error (RMSE).

4. Features and Hyperparameter Optimization

We engineered several new features and experimented with various regression models, to improve the baseline. Our features can be grouped into general, numeric, semantic, and cosine similarity categories. General features include the length of the query, word overlap counts. Numeric features are extracted from product information. These include measurements and quantities. Semantic features were derived using SpaCy to compute the number of matching entities. Cosine similarity was used to measure semantic alignment queries and between for example product titles and descriptions. This was done, because SpaCy is more suitable for shorter text comparison, and cosine similarity work better for longer texts, like product titles and descriptions

Preprocessing included converting all text to lowercase, stemming, and removing stop words. Numerical values and units were standardized using regular expressions.

In addition to Random Forest, we experimented with Gradient Boosting, HistGradientBoosting, Support Vector Regression (SVR), and k-Nearest Neighbors (KNN). For hyperparameter optimization, we used `RandomizedSearchCV`. We focused final tuning on HistGradientBoosting, which supports NaN values. Hyperparameter tuning returned improved results: the best RMSE for Gradient Boosting was 0.4764. While HistGradientBoosting achieved 0.4721

By combining all feature sets we achieved the best performance, without hyperparameter tuning (RMSE = 0.4747). This confirms the value of integrating multiple feature types to capture both lexical and semantic relevance.

5. Results

a) Baseline Results

Model	RMSE	Time (s)
RandomForest + Bagging	0.4831	6.94

Table 1: Baseline RMSE using RandomForest with Bagging

b) Results for Multiple Regression Models

Model	RMSE	Time (s)
Gradient Boosting	0.4784	12.40
HistGradientBoosting	0.4788	0.43
SVR	0.4931	41.87
KNN	0.5211	0.17

Table 2: Performance of different regression models

c) Results of Hyperparameter Optimization

Gradient Boosting is used for the data without numerical attributes, as the model does not handle 'NaN' well. Therefore, the hyperparamter optimization is also done with HistGradientBoosting, which had almost the same results in the initial regression model comparison. This model does handle 'NaN' values.

Model	Best RMSE
Gradient Boosting	0.4764
HistGradientBoosting	0.4721

Table 3: Best RMSE scores after hyperparameter optimization

d) Results for Different Feature Sets

This is a comparison of the different features sets used, and the performance of the model with all feature sets combined.

Feature Set	RMSE
General	0.4886
Numeric	0.5188
SpaCy	0.5169
Cosine	0.4958
All	0.4747

Table 4: RMSE Comparison Across Feature Sets using HistGradientBoostingRegressor

6. Feature Analysis

To analyze which features contributed most to the model's predictions, we used `permutation_importance()` from the `HistGradientBoostingRegressor`. The table below shows the top 5 features ranked by their mean importance.

Feature	Importance (Mean)	Std. Deviation
norm_overlap_title	0.020290	0.000694
len_of_query	0.019772	0.000665
word_in_title	0.018924	0.000882
cosine_title	0.016355	0.000760
cosine_description	0.009452	0.000594

Table 5: Top 5 most important features based on permutation importance

norm_overlap_title is the most important feature. This measures the normalized textual overlap between the search query and the product title. This is a strong signal of relevance because product titles often contain keywords.

len_of_query is the length of the user's query. This is important because shorter queries are more likely to be broader, while longer queries often indicate more specific intent.

word_in_title is the amount of words in the query that appear in the title. This makes it a strong indicator of relevance.

cosine_title measures the semantic similarity between the query and the product title using cosine similarity. This capturs that conceptual overlap, even when the words do not exactly match.

cosine_description does the same for the product description. This is less concentrated than the title, but it still offers conceptual signals which help the model to assess the relevance.

7. Conclusion

In this assignment, various regression models and feature sets have been analyzed. The best performance was achieved by `HistGradientBoostingRegressor`, especially when different attributes were retrieved from the data. These attributes include semantic similarity, textual overlap, and numeric attributes. Analysis of the feature importance showed that overlap and cosine-based comparison have the most impact on the prediction of relevance. Overall, combining multiple types of features and using a robust, tree-based model leads to improved accuracy in query-product matching tasks.