Book Ratings Prediction for amazonkindle

Shiqi Wang
DSI

Dec 6, 2023

GITHUB LINK

Recap

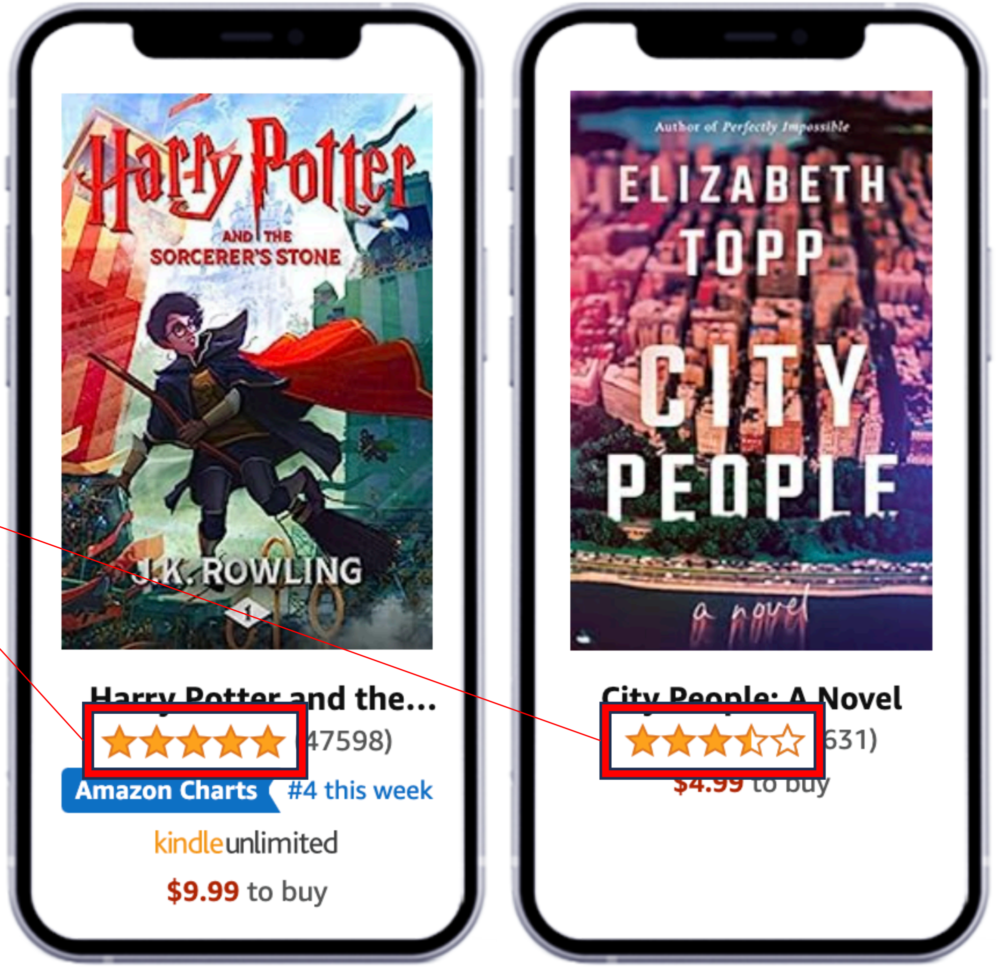
Data Info

- **Background**:
  - Amazon Kindle store is an online e-book e-commerce platform
  - A part Amazon’s retail website
- **Amazon Kindle e-book dataset**:
  - Kaggle
  - Scraped publicly available data
  - Collected in October 2023
  - About 130k observations

amazonkindle

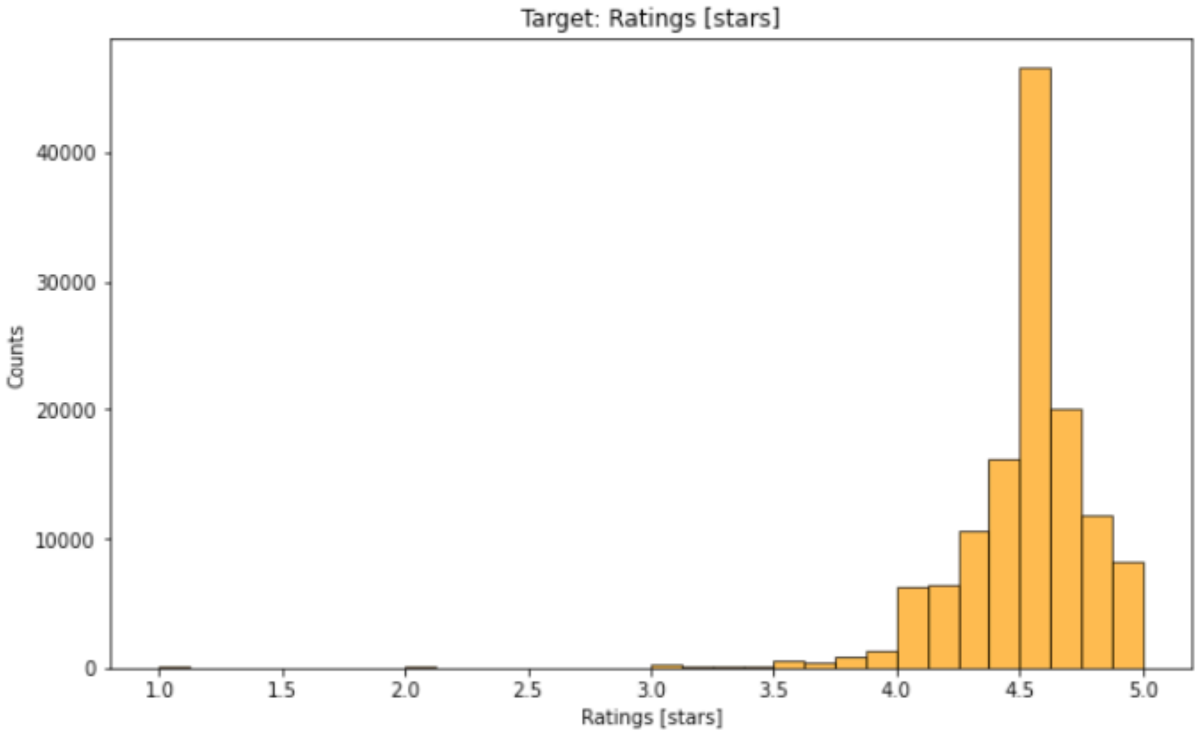
Question

- **Regression:** Predict how future customers will rate an e-book after purchasing.
- **Target Variable**: Average Ratings (Stars)
  - Continuous, range from 1 to 5
  - Rounded to 1 decimal place
- **Why matters?**
  - Ratings reflect customers' satisfaction about the purchase and the e-book
  - Result can be helpful for marketing and business strategies



Preprocessing

- Data shape: (129920, 11→ 90)
- **Missing values:**
  - Publisher: 7% missing
  - Published Days, Published Month, Published Year: 37% missing
  - ”most_frequent” imputer
- **Stratifying:**
  - Left-skewed continuous target variable
  - Assign target into bins and stratify base on bins



Cross Validation

Splitting and Pipeline 1

1) **Iterate Over 3 Random States**

   2) **Data Splitting with Stratification**:

      - For each random state, the data is split into X_other, Y_other (80%) and X_test, Y_test (20%).
      - Stratified based on y_binned, to maintain the proportion of each class.

   3) **Stratified K-Fold Cross-Validation Setup**:

      - A StratifiedKFold object is created for 4 splits.
      - X_other, Y_other is split into X_train, Y_train (75%)and X_val, Y_val (25%).

   4) **Model Training and Hyperparameter Tuning**:

      - In each fold, the function trains the model using a pipeline that includes the preprocessor (one-hot, standard scalar, ordinal) and machine learning algorithm (ML_algo).
      - GridSearchCV: perform hyperparameter tuning based on the provided param_grid.

Splitting and Pipeline 2

5) **Evaluation of Model Performance**:

- The best model from the grid search is evaluated on the X_val, Y_val for each fold.
- The RMSE is calculated for model performance comparison.

6) **Selection of the Best Model**:

- The model with the lowest RMSE in the cv is selected as the best model for each random state iteration.

7) **Testing and Scoring**:

- The best model from each random state iteration is used to predict and score the test set.
- The model’s performance on the test set is evaluated using RMSE and R^2 scores.

8) **Results Compilation**:

- Test RMSE and R^2 scores, and best models, are compiled and returned from the function for each random state.

ML Algorithms

Algorithm	Parameters
Linear Regression: **Lasso**	alpha(L1 regulation): [0.0001, 0.001, 0.01, 0.1, 1, 10]
Linear Regression: **Ridge**	alpha(L2 regulation): [0.01, 0.1, 1, 10, 100, 1000]
Linear Regression: **Elastic Net**	alpha: [0.0001, 0.001, 0.01, 0.1, 1, 10] l1_ratio: [0.0, 0.25, 0.5, 0.75, 1.0]
Random Forest	n_estimators: [10, 50, 100, 200, 300] max_depth: [3, 5, 10] max_features: [0.25, 0.5, 0.75, 1.0]
XGBoost	max_depth: [2, 3, 4, 5, 6] learning_rate: [0.01, 0.1,0.3] n_estimators: [200, 300] reg_alpha(L1 regulation): [0, 0.01, 0.1] colsample_bytree: [0.9] subsample: [0.66]

Results

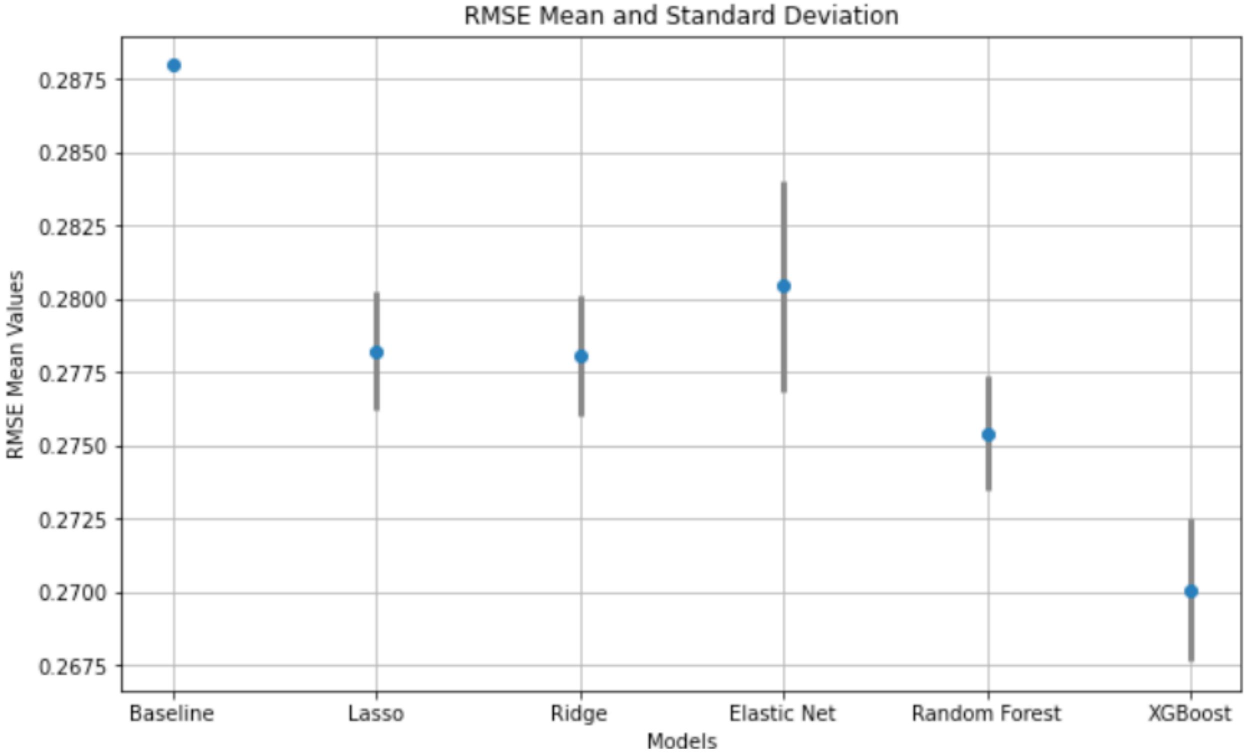
ML Algorithms

Algorithm	Parameters
Linear Regression: **Lasso**	alpha(L1 regulation): [0.0001, 0.001, 0.01, 0.1, 1, 10]
Linear Regression: **Ridge**	alpha(L2 regulation): [0.01, 0.1, 1, 10, 100, 1000]
Linear Regression: **Elastic Net**	alpha: [0.0001, 0.001, 0.01, 0.1, 1, 10] l1_ratio: [0.0, 0.25, 0.5, 0.75, 1.0]
Random Forest	n_estimators: [10, 50, 100, 200, 300] max_depth: [3, 5, 10] max_features: [0.25, 0.5, 0.75, 1.0]
XGBoost	max_depth: [2, 3, 4, 5, 6] learning_rate: [0.01, 0.1,0.3] n_estimators: [200, 300] reg_alpha(L1 regulation): [0, 0.01, 0.1] colsample_bytree: [0.9] subsample: [0.66]

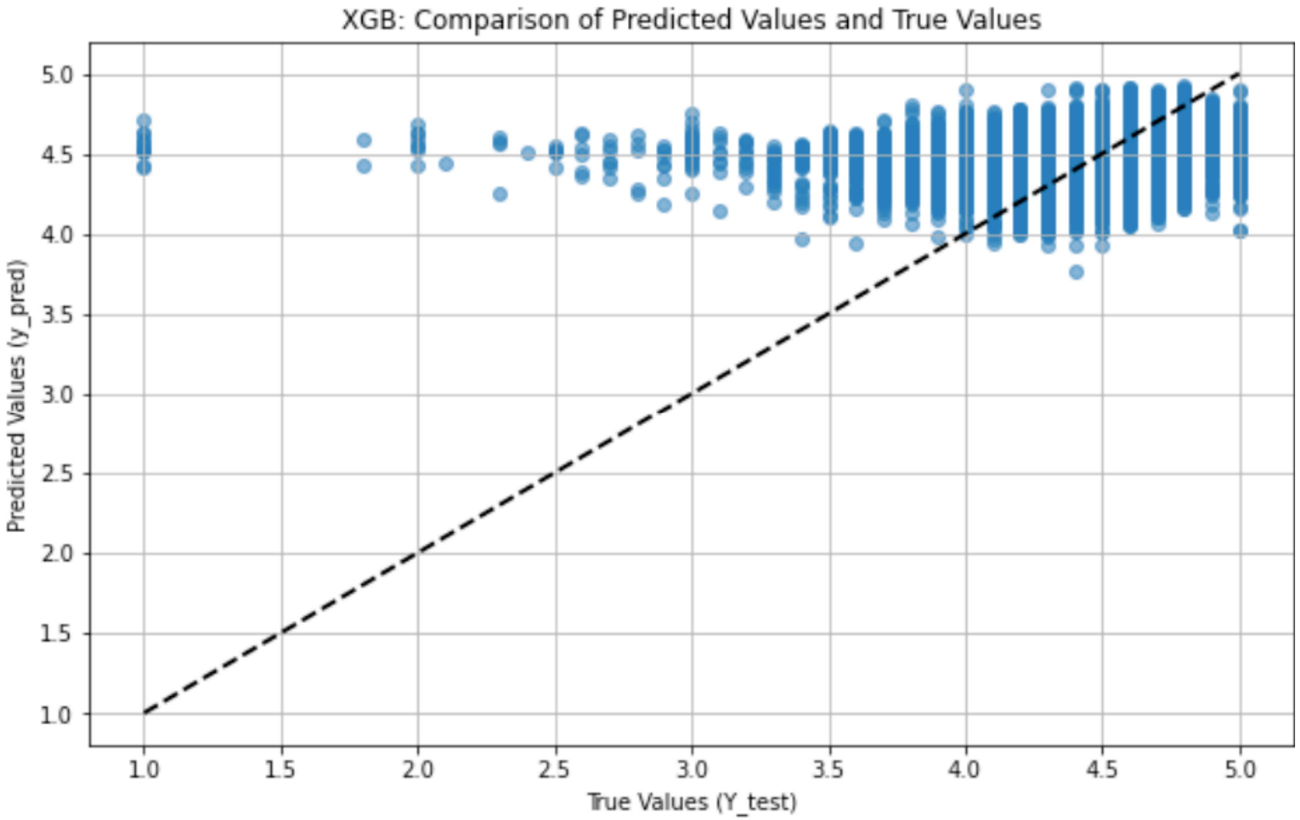
Model Performance 1

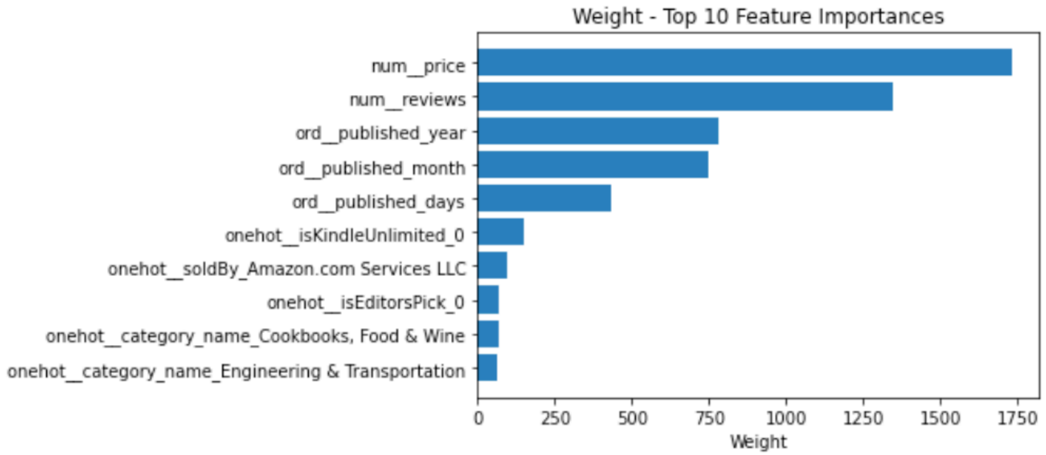
	Baseline	Lasso	Ridge	Elastic Net	Random Forest	XGBoost
RMSE: mean	0.288	0.278	0.278	0.280	0.275	0.270
RMSE: std		0.0020	0.0020	0.0036	0.0019	0.0024
R² : mean		0.0633	0.0647	0.0648	0.0835	0.1161

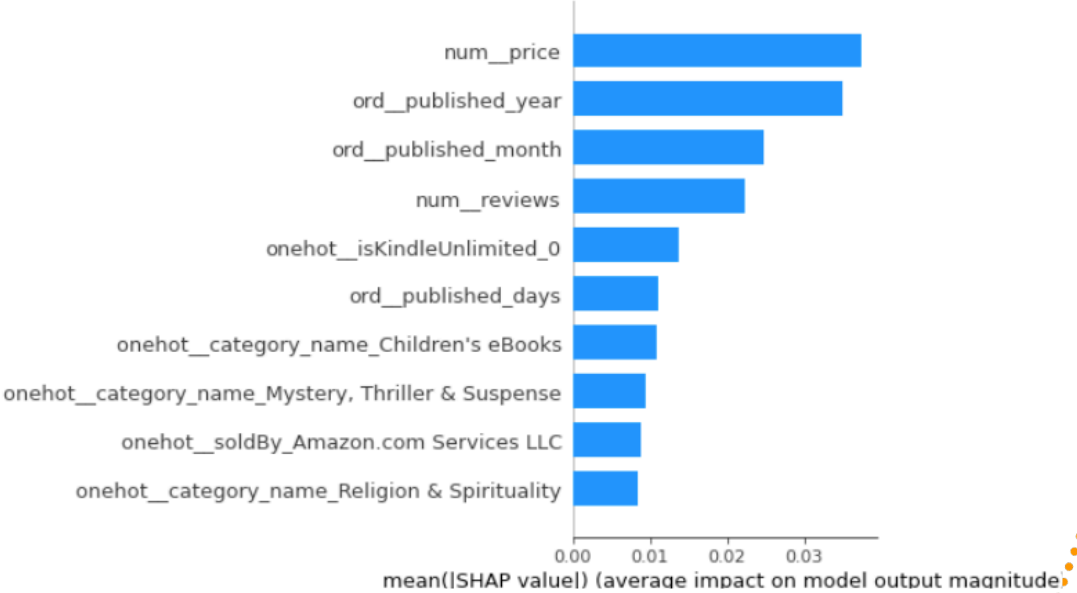
- **Performance**
  - XGBoost has the best performance, Random Forest second
  - Non-linear models may be better
  - Both Lasso and Ridge outperform Elastic Net
- **Overall very low R²**
  - May be problematic, but does not necessarily imply bad model



Model Performance 2







Feature Importance

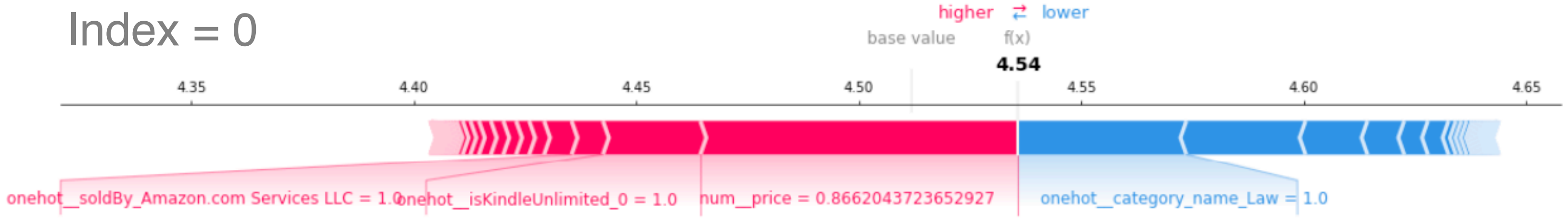
➢ **Global**

- **Top features:**
  - **“price”:**
    - Most influential across both measures.
    - The price of an e-book is a strong predictor to its rating.
  - **“reviews”, “published_year”:**
    - Appear in top 5 both measures.
    - Association with the e-book popularity or temporal trend.
  - **”isKindleUnlimited”**
    - Appears in top 10 across both measures.
  - **Book categories and publisher**
    - Some specific book categories and publishers also appears to be in the top features
    - Do not align in both measures.

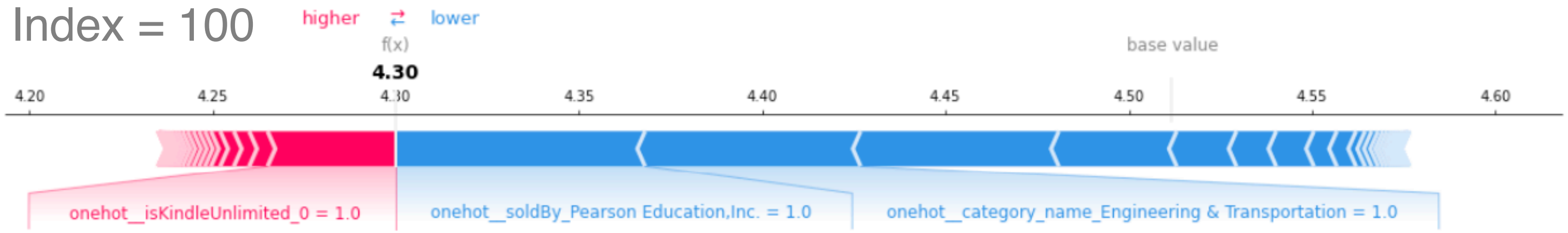
Feature Importance

➢ **Local**

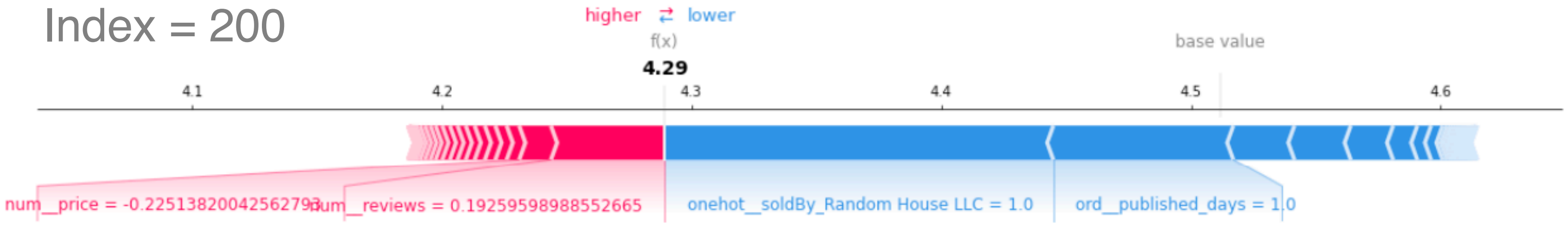
Index = 0



Index = 100



Index = 200



- **Positive influence**
  - “price”
  - “isKindleUnlimited”
  - “reviews”
- **Negative influence**
  - Some specific categories and publishers

Outlook

Future improvement

- **Data collection**:
  - Try to add in more features: current original data only has 9 usable features
  - Combine multiple datasets
- **Preprocessing:**
  - Temporal data has 37% missing values
  - Find better inputer
- **Hyperparameter tuning**:
  - Try more combinations
  - Increase n-estimator for XGBoost: currently only 300

Thanks!

Questions?

Reference

https://www.kaggle.com/datasets/asaniczka/amazon-kindle-books-dataset-2023-130k-books/data

https://scikit-learn.org/stable/index.html

https://en.wikipedia.org/wiki/Kindle_Store

Github Link (in case title page link failed)

https://github.com/ccwxp116/Data1030Project.git