Hotel room price prediction

Team: Dragonflies

Contents

Ideation

Literature Review

"Machine Learning Predicting Airbnb Prices"	• KNN algorithm applied • Find a few rooms that are similar to target room • Average the listed price for the ones most similar to target • Set listing price of target to this calculated average price
Zhang, K., Wang, K., Wang, X., Jin, C., & Zhou, A. (2015). Hotel recommendation based on user preference analysis. 2015 31st IEEE International Conference on Data Engineering Workshops, 134-138.	• Collaborative filtering (CF) and content-based filtering (CBF) applied • Use Expedia search data to represent user data combine collaborative filtering (CF) with content-based (CBF) method to overcome sparsity issue
Fang, Z., Yang, Z., & Zhang, Y. (2015). Collaborative Embedding Features and Diversified Ensemble for E-Commerce Repeat Buyer Prediction.	• Predict linkages of "search" and "property" • Build graph between "search" and "hotel property" • Classify "Search - Property" pair • Predict "Search - Property" linkage
Bollen, Johan et al. "Twitter mood predicts the stock market." J. Comput. Science 2 (2011): 1-8	• Social popularity to predict hotel prices

Dataset Analyzed

Source: **Kaggle**

Dataset: **Personalize Expedia Hotel Searches - ICDM 2013**

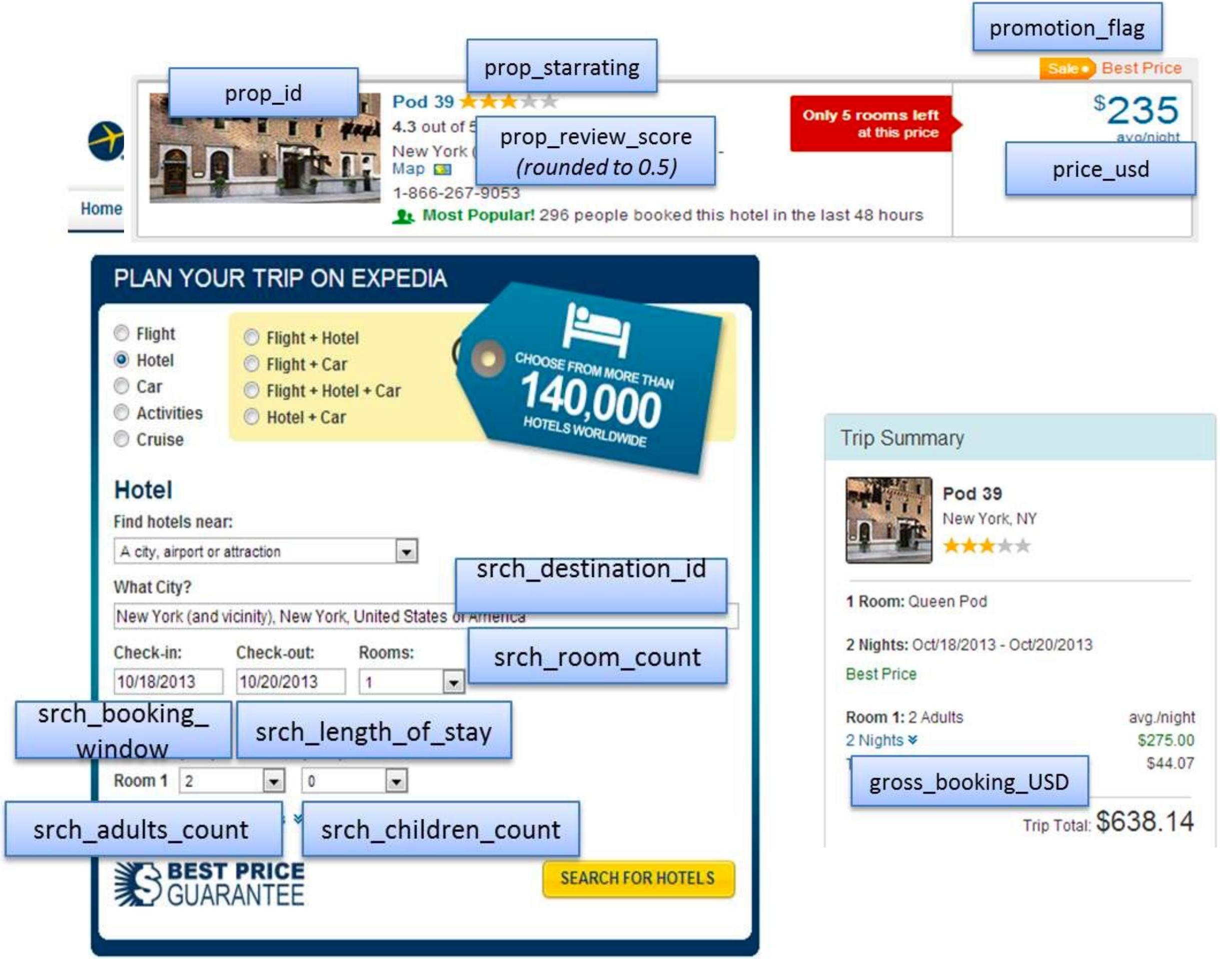
Data Size:

- **Train.csv (2.36GB)**
- **test.csv(1.5GB)**

Information:

- **Hotel characteristics**
- **Location attractiveness of hotels**
- **User's aggregate purchase history**
- **Competitive online travel agencies information**

Features overview



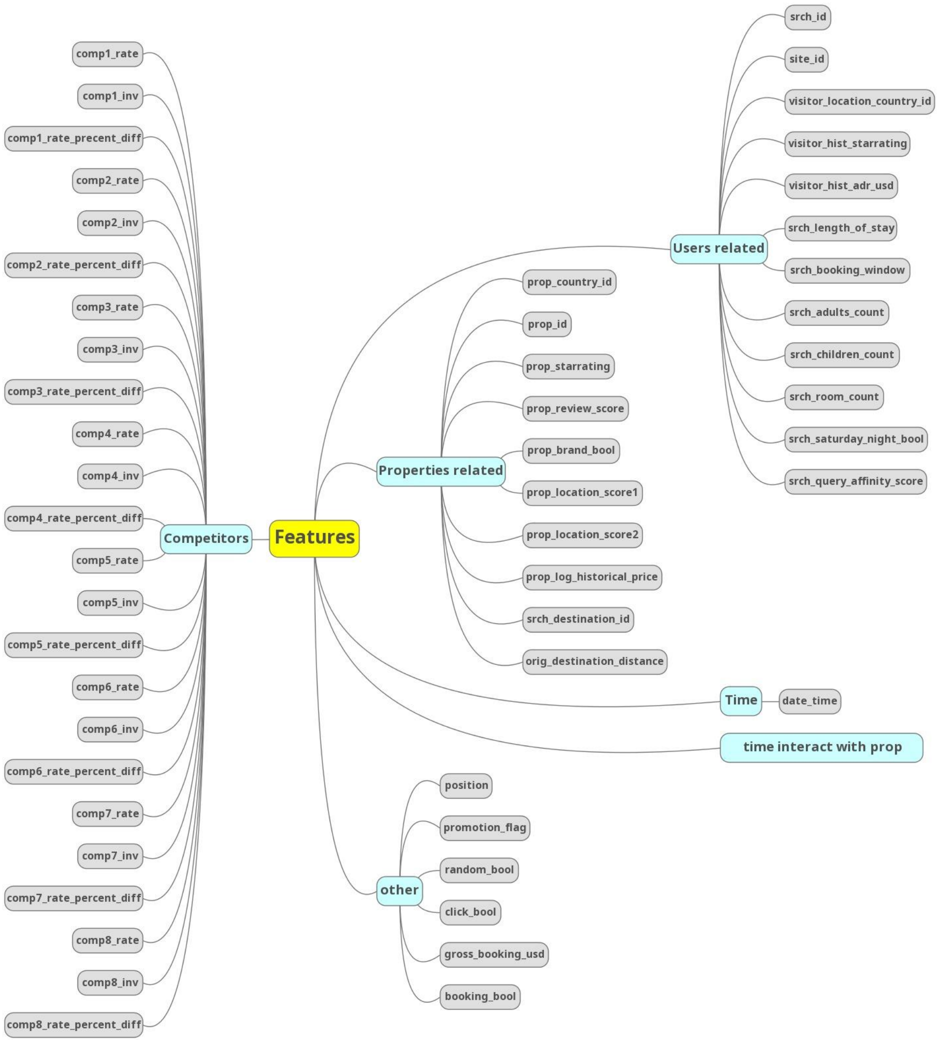
Features overview

Shape: 16,540,159 * 51 **Time:** 2012-11-01 to 2013-06-30

Countries: 174 **Cities:** 28416 **Hotel:** 140821 **Target:** Price_usd

srch_id	date_time	site_id	visitor_location_country_id	visitor_hist_starrating	visitor_hist_adr_usd	prop_country_id	prop_id	prop_starrating	prop_review_score
prop_brand_bool	prop_location_score1	prop_location_score2	prop_log_historical_price	price_usd	promotion_flag	srch_destination_id	srch_length_of_stay	srch_booking_window	srch_adults_count
srch_children_count	srch_room_count	srch_saturday_night_bool	srch_query_affinity_score	orig_destination_distance	random_bool	comp1_rate	comp1_inv	comp1_rate_percent_diff	comp2_rate
comp2_inv	comp2_rate_percent_diff	comp3_rate	comp3_inv	comp3_rate_percent_diff	comp4_rate	comp4_inv	comp4_rate_percent_diff	comp5_rate	comp5_inv
comp5_rate_percent_diff	comp6_rate	comp6_inv	comp6_rate_percent_diff	comp7_rate	comp7_inv	comp7_rate_percent_diff	comp8_rate	comp8_inv	comp8_rate_percent_diff

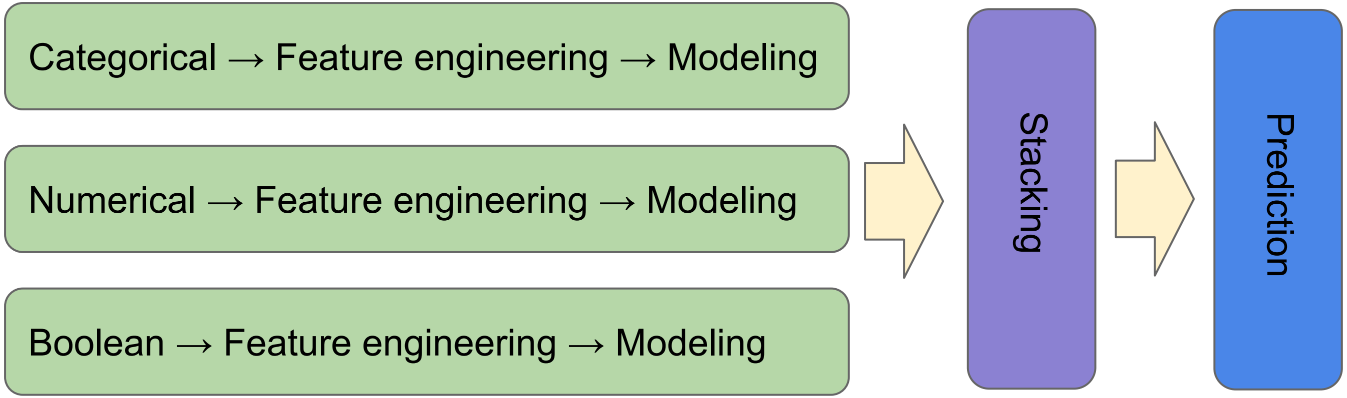
Features structure



Approach 1

- Separate variables by nature

- Pipeline structure

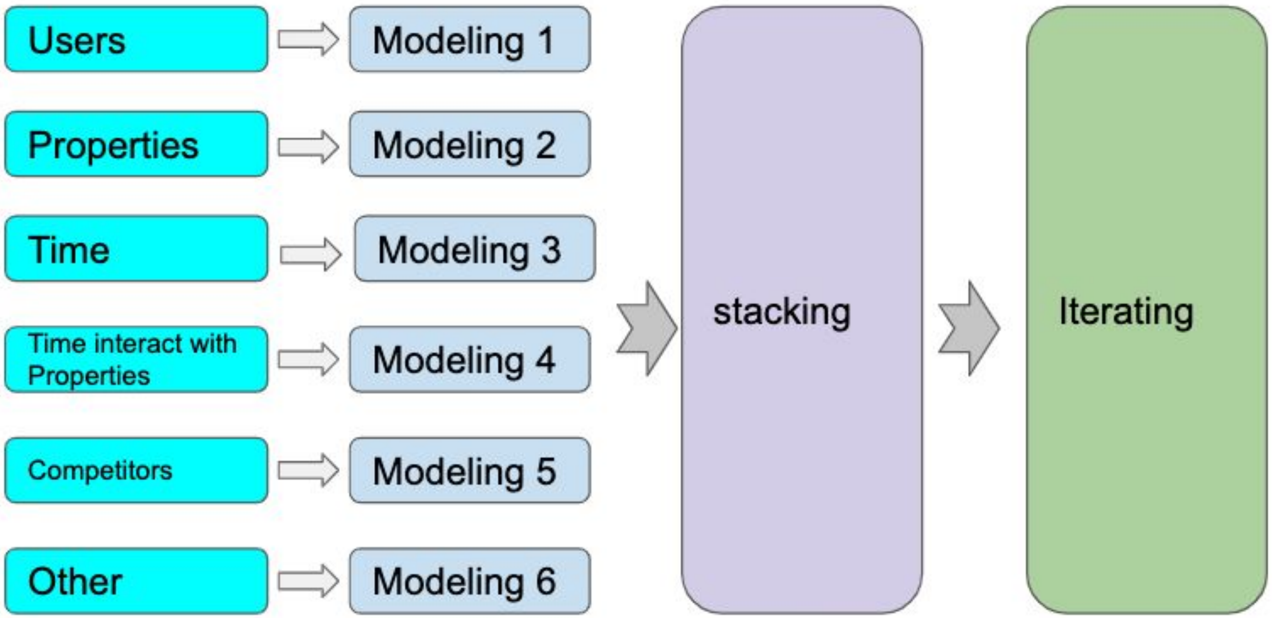


- Issues of approach 1
  - Hard to get the optimized result
  - Don't handle features intuitively
  - Difficult to tell features importance for each attribute
  - Don't integrate the "Date" feature

Approach 2

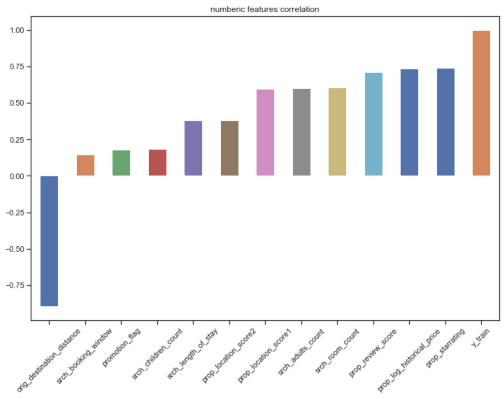
- Separate variables by relevance

- Pipeline structure



Features engineering

- Feature analysis summary from approach 1
  - 6 Categorical features (srch_id, site_id, visitor_location_country_id…)
  - 39 Numerical features (prop_starrating, prop_review_score, prop_location_score1...)
  - 3 Boolean features (prop_brand_bool, srch_saturday_night_bool, random_bool)



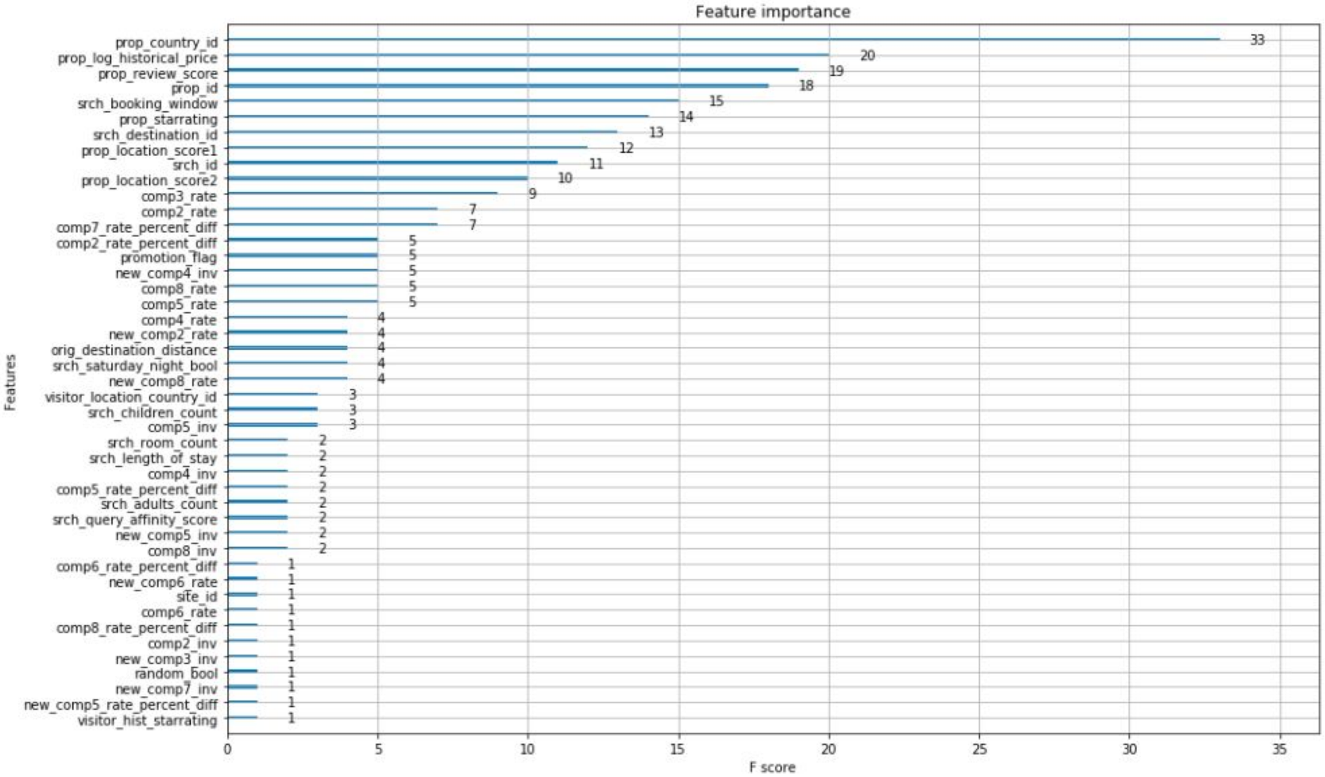
Numeric Features	prop_starrating, prop_review_score, prop_location_score1, prop_log_historical_price, srch_adults_count, srch_room_count, orig_destination_distance
Categorical & Boolean Features	srch_id, site_id, visitor_location_country_id, prop_country_id, prop_id, srch_destination_id prop_brand_bool, srch_saturday_night_bool, random_bool, promotion_flag

Features engineering

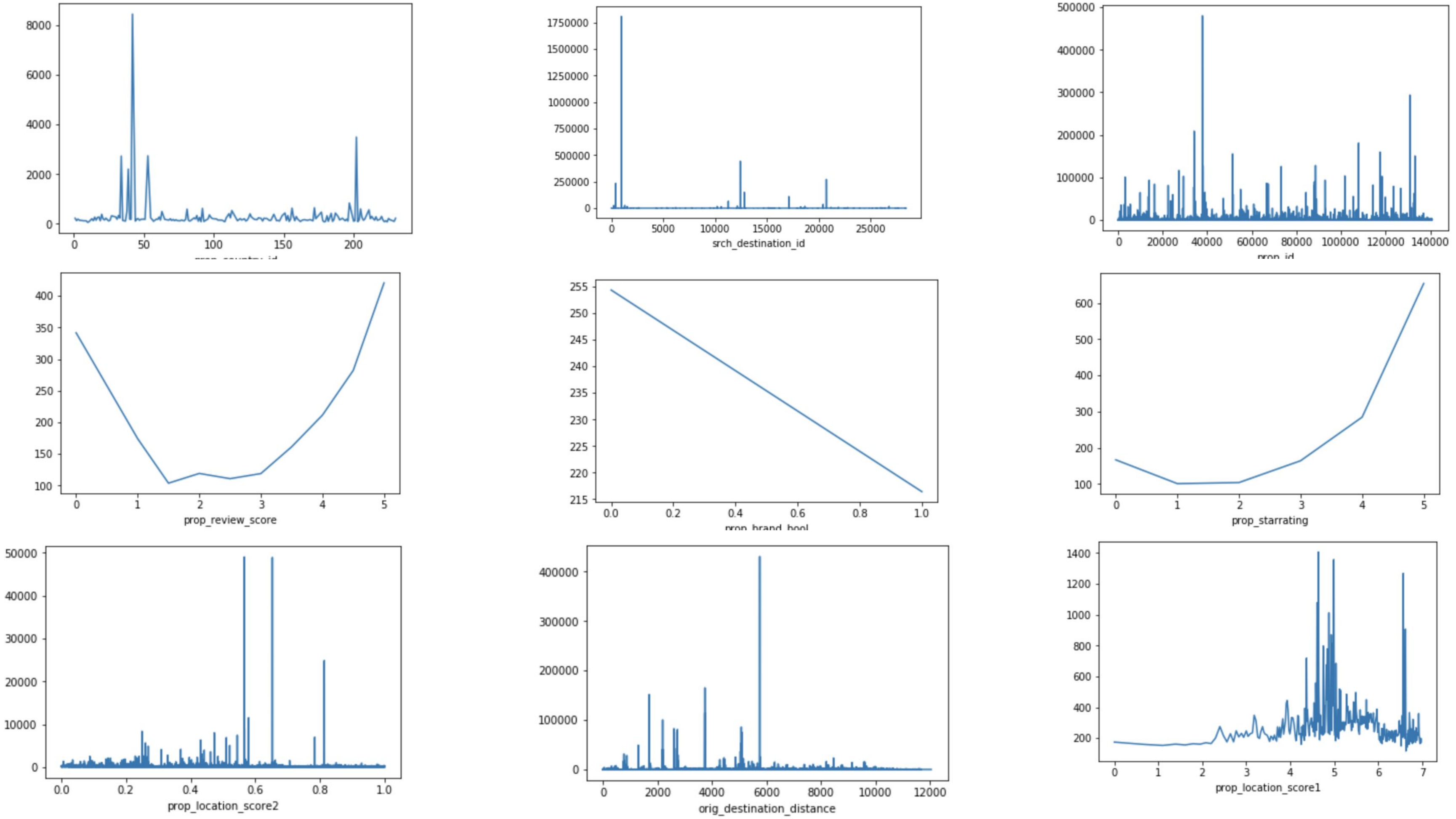
- Missing data handling:
  - > 50% NA in competitor features → drop
  - NA in numerical features → median
  - NA in categorical & boolean features → case by case
- Outlier value detection:
  - Hotel price too low → $0.2/night
  - Hotel price too high → $+5million/night
  - Considering filter out outliers later (for different countries, price gaps exist)
- Transfer categorical features to numerical features
  - Using popularity values instead of ID values
- Date feature:
  - Aggregate date by day, month, quarter → chose by day

Features engineering

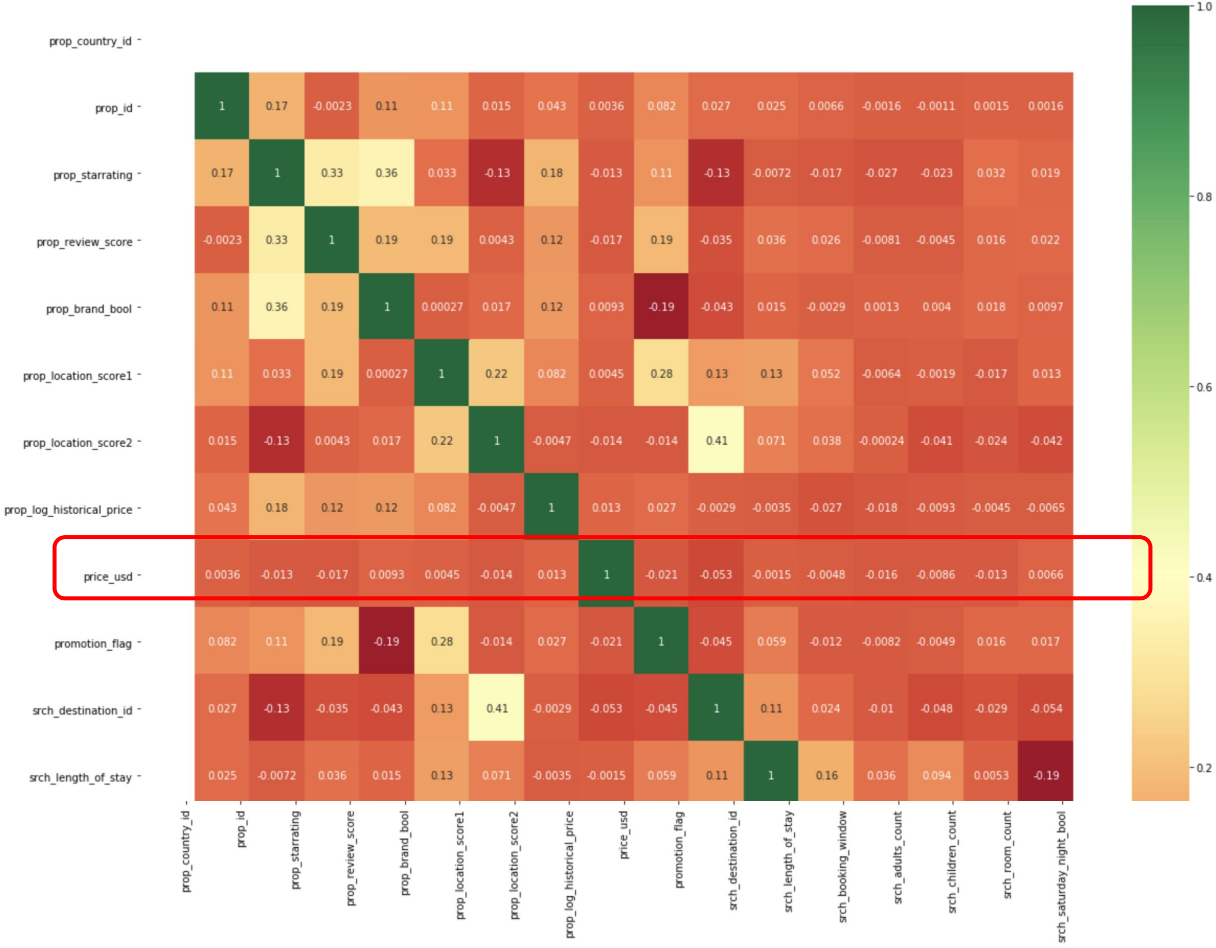
- ## Feature importance
  - Gradient Boosted Decision Trees (GBDT)
  - Feature importance: prop_country_id, prop_historical_price, prop_review_score, prop_id, srch_booking_windows, etc. (ranked in descending order by importance)



Volatility of property features

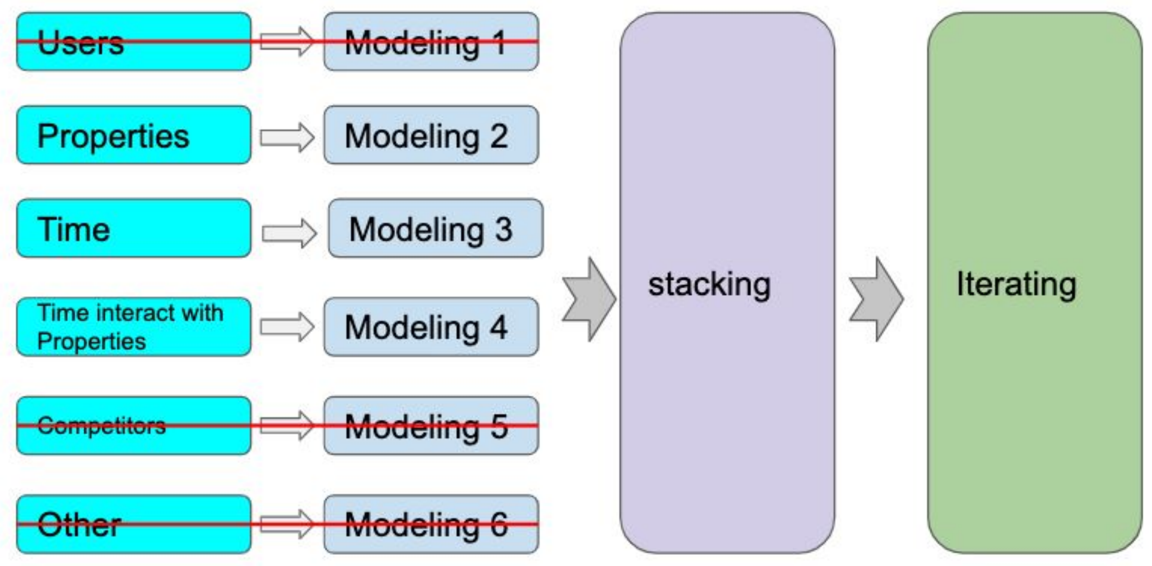


Heatmap of correlated features



Modeling

- Pipeline mode change to



- Key features

  Prop_id, prop_country_id, srch_destination_id, prop_starrating, prop_brand_bool, prop_review_score, prop_location_score, prop_log_historical_price, srch_length_of_stay

- Aggregate data by day

Modeling

- Select and align prop_id with time features
  - prop_id : 116942
  - 242 items
  - High correlation score features: srch_saturday_bool, srch_length_of_stay, srch_booking_window, prop_log_historical_price, prop_location_score1, prop_location_score2

- Property features modeling result

	Linear Regression	Ridge	ElasticNet	Random Forest	Decision Tree
Training RMSE	16.1	17.2	18.3	7.9	2.9
Validation RMSE	19.4	18..8	19.3	19.4	22.8
Test RMSE	2408.3	41.9	17.2	22.2	22.34

Modeling

Time feature modeling:

model 1: ARIMA time series model

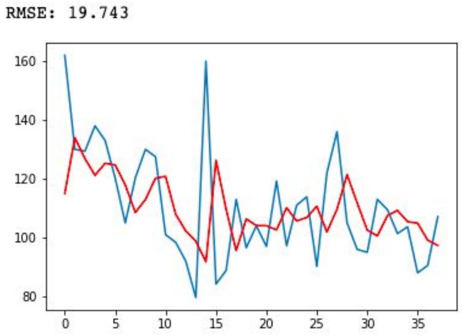
- train RMSE: 19.743
- test RMSE: 16.756

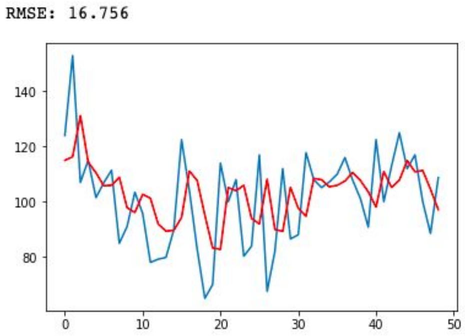
model 2: linear regression model
With extracted time-related features:
'month','day','quarter','week'

- train RMSE: 16.352
- test RMSE: 20.367

	date_time	price_usd	day	week	month	quarter
0	2012-11-01	105.000	1	44	11	4
1	2012-11-02	105.000	2	44	11	4
2	2012-11-03	127.140	3	44	11	4
3	2012-11-04	91.380	4	44	11	4
4	2012-11-05	91.605	5	45	11	4

price data len: 242





Modeling

- stacking model 1 and model 2:

Using prediction results from the previous two models as new features, and fit a second-layer regression model

2nd layer modeling	train_RMSE	test_RMSE
XGBoost (with hyperparameter tuning)	18	17

Modeling

- Stacking property and time modelings together:

time + property modeling:

	train RMSE	test RMSE
linear regression	12	21
random forest	5	23
XGboost (with param tuning)	22	18

- Work Ongoing:
    - Using cross validation to avoid overfitting (K = 5)
    - Set property selection threshold: 75 percentile popularity, get the corresponding popularity value = 234
    - Select all property id by using the threshold
    - Tuning model by iterating all qualified property id dataset to get optimized performance
- Future work:
    - Remove outliers: set price threshold = >$10 and <$100,000
    - Parameter tuning for algorithms
    - More feature engineering (e.g. the impact of competitors)

Take away

Takeaways of the project:

- Hotel price is predictable
- Hotel price is related with hotel's attributes and its fluctuation vary by time
- Feature engineering is the key
- Multiple layers modeling to integrate the regression model and time series model

Thanks!