Evolution of content moderation approaches for online classifieds: from action recommendations to automation

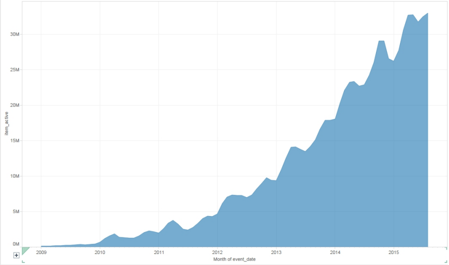
Ivan Guz, Vasily Leksin, Mikhail Trofimov, Alexandra Fenster

Avito.ru

23.09.2015

Contents

Introduction



- Classifieds become more and more popular
- Human moderation of all income flow of ads becomes unrealistic
- Complex approach for automatic moderation based on machine learning methods required

Data description

Each ad d_i is described by 6 groups of data:

- Title and description texts
- Placement of an ad in catalog - category and additional attributes
- Geographic location - region, city, district
- Requested ad price
- Provided images
- Contact information of the seller.

Based on this data vector of numeric features $\vec{f} = (f_1, \ldots, f_N)$ is constructed. Feature preparation logic is unique for each group of data.

Avito

Task definition

- Each individual ad is checked to comply with a set of rules
- We need to historical collection of ads predictive model for each reject reason
- It is required for each model to predict one number - reject probability $p \in [0, 1]$ for corresponding reason
- $D = (d_1, \dots, d_L)$ - historical collection of ads
- Each ad d_i is classified (belongs) to a single category, $\{c_i\}_{i=0}^{K-1}$ – possible item categories (Cars, Real Estate, Personal belongings, etc.)
- For each ad d_i we know human decision vector $\vec{y}_i = (y_1^i, \dots, y_r^i), y_j^i \in \{0, 1\}$

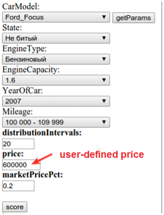
Avito

Prediction models overview

The following classes of algorithms are implemented in our system:

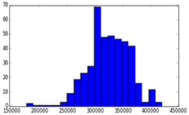
- Text classification models
- Wrong category models
- **Price prediction models**
- Duplicates models
- Image prediction models

Cars pricing model



"avgPrice": 324088.9952606635,
"priceStatus": 1,
"estimationSlice": {
    "State": "Не битый",
    "Mileage": "100 000 - 109 999",
    "EngineType": "Бензиновый",
    "YearOfCar": "2007",
    "EngineCapacity": "1.6 - 1.7"
},
"medianPrice": 325000.0

price is high



Given data:

- Set of possible parameters of cars
- Information about specific cars and prices

Task: Construct a query to the database, the result of which would contain not less than N objects that are close to original

Task definition

Lets

- F_i – partially ordered set of possible values of i-th car parameter
- Slice p – ordered set of k elements $(a_i, b_i), a_i \in F_i, b_i \in F_i, a_i \leq b_i$
- Entering the relation of embedded slices:
  $p_i \subset p_j : \forall m \in (1, \cdots, k) \quad a_m^i \geq a_m^j, \quad b_m^i \leq b_m^j$
- $X = \{((p_1, \cdots, p_k), y)\}$ - set of cars in the database, $y \in R$ – car price
- $T(p)$ – true price distribution for the parameters slice
- $S(p) : P \to 2^X$ - set of cars in parameters slice

We need to find:

- $\hat{p}(p) : \hat{p}(p) = min_{\hat{p}} Dist(T(\hat{p}), T(p)) \quad$ w.r.t. $\quad |S(\hat{p})| \geq N$

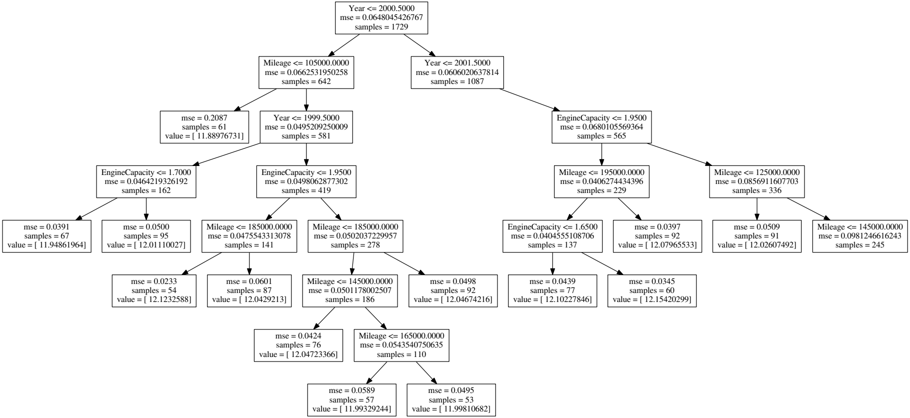
Data preparation

- Actual ads that were active on the site for more than a days and less than b days
- Last date of activity within last n days
- Not blocked by moderators
- Filter price biases
- Final sample: 2 035 437 ads

Model training

- Trained decision tree regressor for each car model with minimum leaf size equals to $M = 20$
- Cars that fall into the same tree leaf are similar because they have similar price and each leaf is defined by a set of rules on car characteristics which we identified as a slice we were looking for
- We selected best decision tree training method that minimized RMSLE on the training data.
- It could not overfit because we had restriction on a minimum leaf size

A fragment of decision trees



Model testing

We compared two models:

- Decision Tree Regressor
- Linear Regression with L1-regularization (Lasso)

Model name	RMSLE by car model	RMSLE entire
Decision Tree Regressor	0.297	0.268
Lasso	0.295	0.269

Probability of an incorrect price is determined by user-specified price deviation from predicted price.

Moderation automation

For each reject reason $j \in 1,\dots,r$ we trained the model m_j that predicts reject probability p_j^i for each ad d_i. Also for each reason j we need to define $\delta_j^a \in [0,1)$ - automatic allow threshold and $\delta_j^r \in (\delta_j^a, 1]$ - automatic reject threshold. Based on these definitions final automatic verification decision $M(d_i)$ should be taken using following logic:

$$M(d_i) = \begin{cases} \forall j : p_j^i < \delta_j^a & \Rightarrow \quad \text{Allow} \\ \exists j : p_j^i > \delta_j^r & \Rightarrow \quad \text{Reject} \\ else & \Rightarrow \quad \begin{array}{l} \text{Recommend to reject} \\ \text{for reason } j = \underset{j}{\operatorname{argmax}}\, p_j^i \end{array} \end{cases}$$

Questions

Thank you!