PREDICAR: A Tool For Car Price Prediction*

Max Dávid Karel
FIIT, STU

Peter Bokor
FIIT, STU

Ondrej Harnúšek
FIIT, STU

Tomáš Vrtal
FIIT, STU

Branislav Baláž
FIIT, STU

Richard Letanec
FIIT, STU

Veronika Vejčíková
FIIT, STU

ABSTRACT

There are tens of thousands of used cars for sale online in Slovakia right now. Buying a used car can be a great way to get a good car for a reasonable price. However, an inexperienced customer may be lost among the many technical parameters and end up with not a very good deal at all. This is where our product comes in. We monitor top used-cars markets and use machine learning to help customers decide if the car they consider buying has a reasonable price and what other similar offers they might like to look at first.

Our proposed solution is a one-page web application with a simple and intuitive user interface. The user enters a link of a car ad and is shown overview of the car's characteristics and our estimate of what the reasonable price should be for a car with such attributes. Moreover, they get a list of similar cars displayed to consider and possibly choose from.

The application is built on a microservice architecture. Each microservice runs in a separate docker container. Our frontend solution is a Vue.js[1] application. Backend is implemented in Python with the Flask framework. We use MongoDB for storing crawled ads. Machine learning with word2vec embeddings and SVM are used for NLP entity recognition of car features from fulltext data. For finding similar ads, deep learning is used.

KEYWORDS

price prediction, used cars, web scraping, data analysis, machine learning, NLP

INTRODUCTION

Many people prefer buying a used car instead of a new one due to financial reasons. However, the used-car market is full of opportunistic sellers who sell cars for more than the appropriate price for the given parameters. Regular customers who aren't experts in the car domain need to rely on someone else's advice about whether the car they consider buying has reasonable price. Otherwise, they risk buying overpriced.

We provide a solution for this problem. There are currently 5 big and many more small online markets which provide a platform for private and corporate advertising of used cars for sale. We monitor the market and collect ad data of cars that are being sold. Our growing database allows us to help customers make data-driven decisions when buying a used car. When the user enters a url of a car they consider buying, we provide them with the price we calculated as appropriate for the car. We compare the prices and give them information about how many similar cars currently on the market are cheaper or more expensive than the selected one. We also show them similar cars on the market for them to consider.

ARCHITECTURE

The application utilizes a microservice architectural style. It uses docker containers on the server.

Main components of the application as displayed on Fig. 1 are:

- Frontend - web user interface, single-page Vue.js application
- Manager - handles requests from frontend and manages communication among other components of the application
- Database - MongoDB database, has all the crawled and parsed ads as well as the queue for crawler ad requests and processing
- Predictor - component for price prediction, uses deep learning
- Crawler - collects data from used-car markets
- DB updater - keeps the ads up to date, passes DB and inserts ads into queue to be re-visited to know when they are sold or changed
- Index Explorer - generates markets' index pages and ads to queue

[1] https://vuejs.org/

ISBN .

*http://predicar.pokope.sk

- Queue Runner - component for interaction with queue, gets records from queue and passes them to parsers
- Text Processor - handles NLP processing of fulltext ads to extract attributes from them

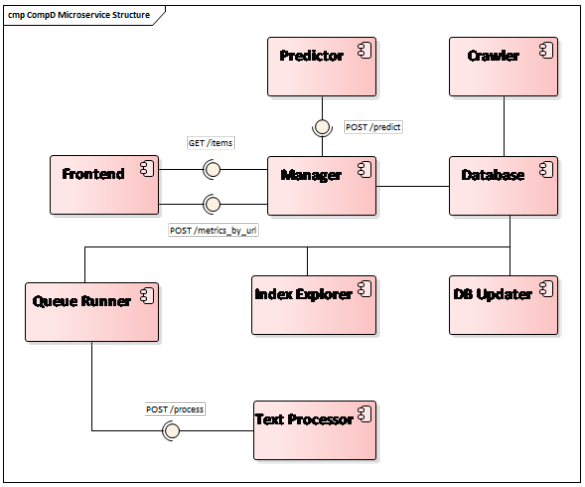

Figure 1. Predicar component structure view

DATA COLLECTION

At the moment we monitor 4 major Slovak online markets for advertising used cars: autobazr.eu, autovia.sk, auto.bazar.sk, auto.bazos.sk. We implemented four parsers, one for each online market, and a crawler service which constantly and in parallel collects data from these four web sites.

The crawler service consists of three components Database Updater, Index Explorer and Queue Runner, and uses custom made queue. Database updater regularly pushes old ad pages from the database to the queue in order to update them. Index Explorer generates index pages for the online markets. Queue Runner pulls pages from the queue and calls the parsers to parse the pages. The parsers then parse the data from the pages and insert them into database.

GETTING ATTRIBUTES FROM FULLTEXT ADS

Online markets which specialize only for cars have ads displayed in a structured way. That is not the case for auto.bazos.sk where users can write the ad in fulltext form. This makes the ad data unsuitable for our machine learning purposes. If we want to estimate the price of a car we need to know its attributes like mileage, engine and fuel type, year when the car was produced etc.

We trained custom Named Entity Recognition model which currently recognizes 10 attributes from fulltext ads: brand, model, production_year, power_kw, fuel, mileage, engine_size, specific_type, horsepower, transmission_type. Each parsed fulltext ad from auto.bazos.sk is processed with text processor and attributes which have been identified in in the text are saved into the database in structured manner.

To represent the words for this gensim word2vec model [1] is used. It was trained on all our crawled fulltext data. These vector representations of each word plus previous and following word along with some additional orthographic features are used in SVM classificator to predict if the word is any of the known entities. Our method is successful achieving 96% average prediciton accuracy for 11 classes (10 attributes and class 'nothing').

FRONTEND

User interface is minimalistic and simple to use. Main page contains instruction steps with input area and button for search and analysis. There are also most recently crawled ads displayed under this.

Car Detail

Ad detail for an analyzed car contains structured data about it and its photos which we load from the online car markets. On the side there is a box with actual price and predicted recommended price calculated based on the data about similar cars. The customer can clearly see the prices difference and info-bar which notes how many cars have higher or lower price with appropriate color for how good or bad the price is.

Similar Cars

Under the car's description there is a list with cars which our model marked as the most similar. Basic informations about the car are visible along with its price and photo. User can click on the car and then the selected car is being analyzed and its whole car detail with price predictions and similar cars is displayed.

FINDING SIMILAR CARS AND PREDICTING PRICE

We use deep learning for finding similar cars. Our model uses triplet network architecture to learn the similarities between currently selected car and cars in database. Based on features calculated from the car it queries the DB for similar cars. The features can be learned in different ways - by similar brand and model, look, engine, or wear, and thus the model can be used as a personalised recommender.

For calculating price only similarity for brand and model is considered and from those similar cars the price is calculated by averaging.

REFERENCES

[1] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50. `http://is.muni.cz/publication/884893/en`.