Car Selection Advisor & Price Predictor

Content

- WHY ?
- WHAT ?
- HOW ?

Our Team

- NAZ İREM BAZ
- BERK ÖZGÜR
- RIZA SEMİH KOCA

WHY?

The Problem

Choosing the best car among the choices given becomes a critical decision-making problem.

- Many features (fuel economy, comfort, life span, reliability, etc.) to consider
- Varying quality, features, and price
- Differing customer demands

WHAT?

CARVISOR asks customers their preferences and selects the best cars for them accordingly.
Moreover, CARVISOR predicts the optimal price for the car and therefore, can evaluate the ad.

The outcome of this decision-making process will be significant and beneficial for both customers and manufacturers.

This project will mainly focus on the following objectives:

- TO IDENTIFY THE CORRELATION BETWEEN PRICE AND OTHER FEATURES OF A VEHICLE AD.
- TO USE THE DECISION TREE ALGORITHM TO PREDICT THE OPTIMAL PRICES
- TO USE THE TOPSIS METHOD IN EVALUATING THE BEST CARS BASED ON SELECTED CRITERIA'S.
- TO PROVIDE THE RANK FOR EACH SELECTED CARS BASED ON THE CAR SEGMENT IN OTOEKSPER.COM

HOW?

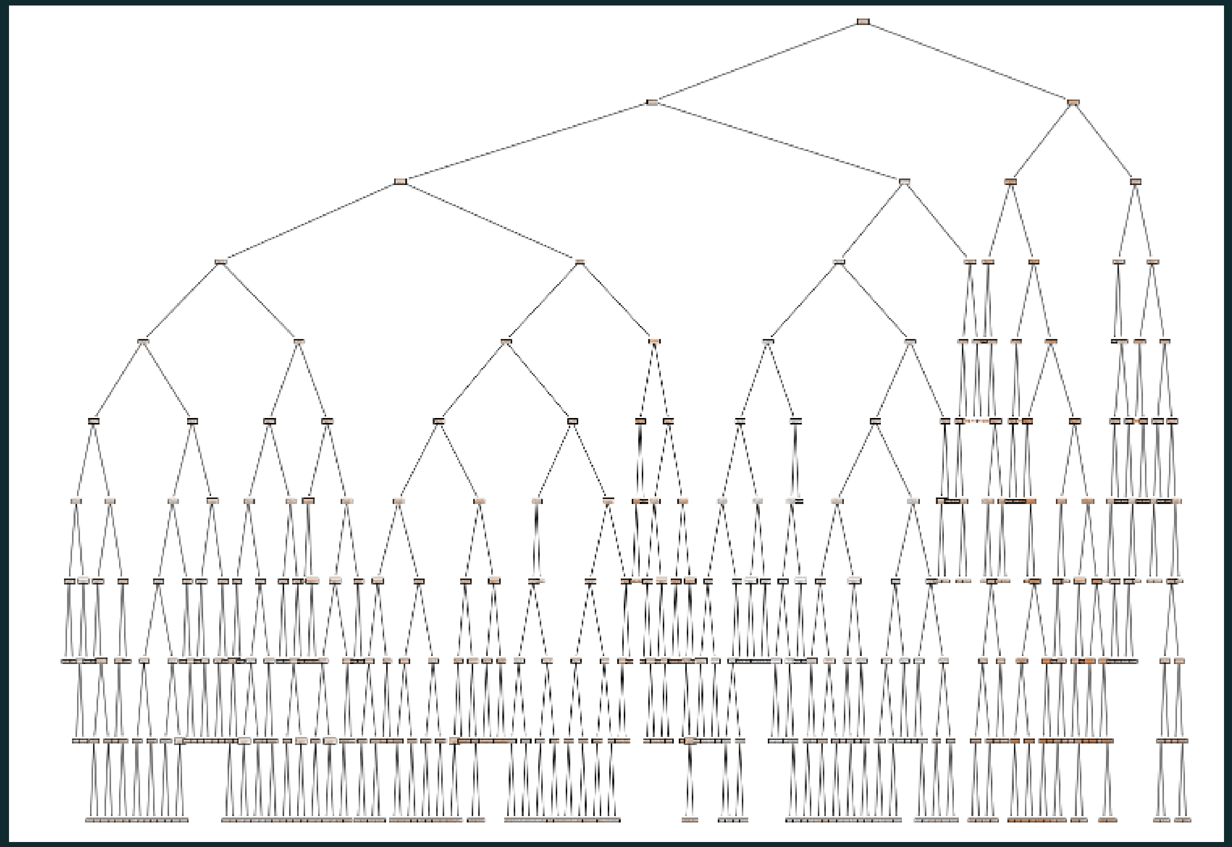
Data Gathering

To gather data from the second-hand car websites, we used web scraping. Web scraping gathers all the information about the currently available second-hand cars from the website *otoeksper.com*. The website is chosen purposely as it provides some information about the car that many other websites don't. We gathered the information of 804 second-hand cars to help building our model.

otoeksper İKİNCİ EL GÜVEN İSTER

Prediction Model: Decision Tree

1- Web Scraping via Python:
https://www.otoeksper.com.tr/ikinci-el?sayfa=



2- Data Manipulating:
Label encoding, feature selection

3- Model Selection
(Regression/Decision Tree/Random Forest/ XGBOOST)

4- Best Parameters Selection via Cross Validation(250 branch with lowest error score)

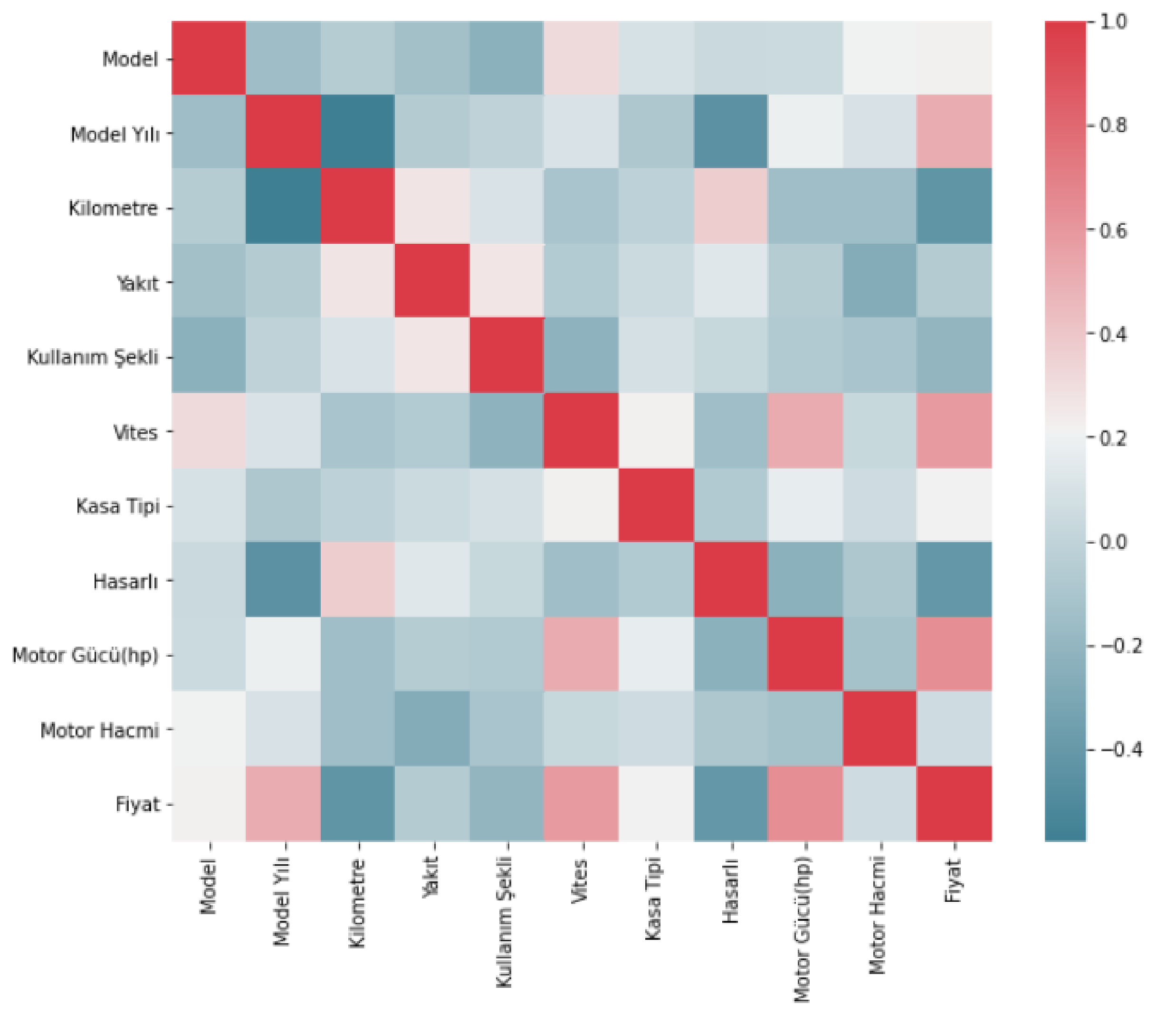
5- Model Implementation (Decision Tree)

*few data: after manipulation 589

The Implementation

```python
a = ['Model', 'Model Yılı', 'Kilometre', 'Yakıt', 'Kullanım Şekli', 'Vites','Kasa Tipi', 'Hasarlı', 'Motor Gücü(hp)', 'Motor Hacmi']
a = ['Jeep Grand Cherokee Grand Cherokee 3.0 V6 CRD Limited Otomatik',18,189.000,1,1,1,1,1,75,1.149]
a = [5,18,189.000,1,1,1,1,1,75,1.149]
a = np.array(a) # convert to a numpy array
a = np.expand_dims(a, 0) # change shape from (8,) to (1,8)
dt_model.predict(a) # voila!
```

```
array([265000.])
```



TOPSIS Method

- TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) is the multi criteria decision analysis method we used for deciding on which car to pick.
- It is a method of compensatory aggregation that compares a set of alternatives by identifying weights for each criterion, normalizing scores for each criterion and calculating the geometric distance between each alternative and the ideal alternative, which is the best score in each criterion.
- Among the other multi-criteria decision making methods, TOPSIS is selected as both the qualitative and quantitative can be used with this method.
- TOPSIS is useful with choice and ranking problems, which is what we are looking for.

Step 1

Create an evaluation matrix consisting of m alternatives and n criteria, with the intersection of each alternative and criteria given as x_{ij}, we therefore have a matrix $(x_{ij})_{m \times n}$.

Step 2

The matrix $(x_{ij})_{m \times n}$ is then normalised to form the matrix $R = (r_{ij})_{m \times n}$, using the normalisation method

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{k=1}^{m} x_{kj}^2}}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n$$

Step 3

Calculate the weighted normalised decision matrix

$$t_{ij} = r_{ij} \cdot w_j, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n$$

Step 4

Determine the worst alternative (A_w) and the best alternative (A_b):

$$A_w = \{\langle \max(t_{ij} \mid i = 1, 2, \ldots, m) \mid j \in J_- \rangle, \langle \min(t_{ij} \mid i = 1, 2, \ldots, m) \mid j \in J_+ \rangle\} \equiv \{t_{wj} \mid j = 1, 2, \ldots, n\},$$

$$A_b = \{\langle \min(t_{ij} \mid i = 1, 2, \ldots, m) \mid j \in J_- \rangle, \langle \max(t_{ij} \mid i = 1, 2, \ldots, m) \mid j \in J_+ \rangle\} \equiv \{t_{bj} \mid j = 1, 2, \ldots, n\},$$

where,

$J_+ = \{j = 1, 2, \ldots, n \mid j\}$ associated with the criteria having a positive impact, and

$J_- = \{j = 1, 2, \ldots, n \mid j\}$ associated with the criteria having a negative impact.

Step 5

Calculate the L^2-distance between the target alternative i and the worst condition A_w

$$d_{iw} = \sqrt{\sum_{j=1}^{n} (t_{ij} - t_{wj})^2}, \quad i = 1, 2, \ldots, m,$$

and the distance between the alternative i and the best condition A_b

$$d_{ib} = \sqrt{\sum_{j=1}^{n} (t_{ij} - t_{bj})^2}, \quad i = 1, 2, \ldots, m$$

where d_{iw} and d_{ib} are L^2-norm distances from the target alternative i to the worst and best conditions, respectively.

Step 6

Calculate the similarity to the worst condition:

$$s_{iw} = d_{iw} / (d_{iw} + d_{ib}), \quad 0 \leq s_{iw} \leq 1, \quad i = 1, 2, \ldots, m.$$

	TOPSIS	VIKOR	COPRAS	MULTIMOORA	PROMETHEE-GAIA	AHP
Type of normalization	Vector normalisation (square root of sum (L2 normalization)	Linear normalization (L1 normalization)	Vector normalization (sum)	Vector normalization (square root of sum)	Normalization is performed automatically	Vector normalisation (sum)
Suitability	Choice problems, ranking problems	Choice problems, ranking problems	Choice problems, ranking problems	Choice problems, ranking problems	Choice problems, ranking problems, description problems (GAIA)	Choice problems, ranking problems, sorting problems (AHPsort)
Inputs	Ideal and anti-ideal option weights	Best and worst option weights	Best and worst option weights	Best and worst option weights	Indifference and preference thresholds weights	Pairwise comparison on ratio scale (1–9)
Outputs	Complete ranking with closeness score to ideal and distance to anti-ideal	Complete ranking with closeness score to best option	Complete ranking	Complete ranking	Partial and complete ranking (pairwise outranking degrees)	Complete ranking with scores
Preference function	Distance metric (Euclidean distance, Manhattan distance, Tchebycheff distance)	Distance metric (Manhattan distance)	Min Max	Min Max	Usual, Linear, U-shape, V-shape, Level, Gaussian	
Approach	Qualitative and/or quantitative	Quantitative	Quantitative	Quantitative	Qualitative and/or quantitative	Qualitative
Ranking scale	0 to 1	Positive values	Positive values	Positive values	−1 to 1	0 to 1
Best alternative	Max value	Min value	Max value	Max value	Max value	Max value
Consistency levels	no restrictions	no restrictions	no restrictions	no restrictions	7±2	9
Software	MS Excel, Matlab, Decerns	MS Excel	MS Excel	MS Excel	Visual Promethee, Decision Lab, D-Sight, Smart Picker Pro	MS Excel, MakeItRational, ExpertChoice, Decision Lens, HIPRE 3+, RightChoiceDSS, Criterium, EasyMind, Questfox, ChoiceResults, 123AHP, DECERNS

The Implementation

	Plaka	Kilometre	Yakıt Sarfiyatı	Araç Büyüklüğü	Motor Hacmi	Fiyat	Araç Genel Sağlık Puanı	Model Yaşı	Distance Positive	Distance negative	Topsis Score	Rank
228	35 RN 125	32172	6.6	1	1368.0	100000	7.000000	27	0.100939	1.418178	0.933554	1.0
230	20 AEV 156	3122	5.2	4	1598.0	213950	6.272727	2	0.113281	1.434755	0.926823	2.0
231	20 AEV 971	3089	5.2	4	1598.0	213950	5.909091	2	0.113618	1.434653	0.926616	3.0
563	34 ESJ 876	4	5.7	3	1368.0	208500	6.545455	1	0.114843	1.438860	0.926084	4.0
571	38 AFF 612	8297	5.7	3	1368.0	207950	6.545455	1	0.115229	1.427481	0.925307	5.0

THANK YOU...