Predicting Final Listing Prices in a Private Used Car Market

- using Random Forest Algorithm

Group 2

Ji-Soo Kim, Tae-Yoon Kim, Jun-Beom Lee, Jin-Joo Yang, Sung-Hyun Kim

Table of Contents

1. Introduction

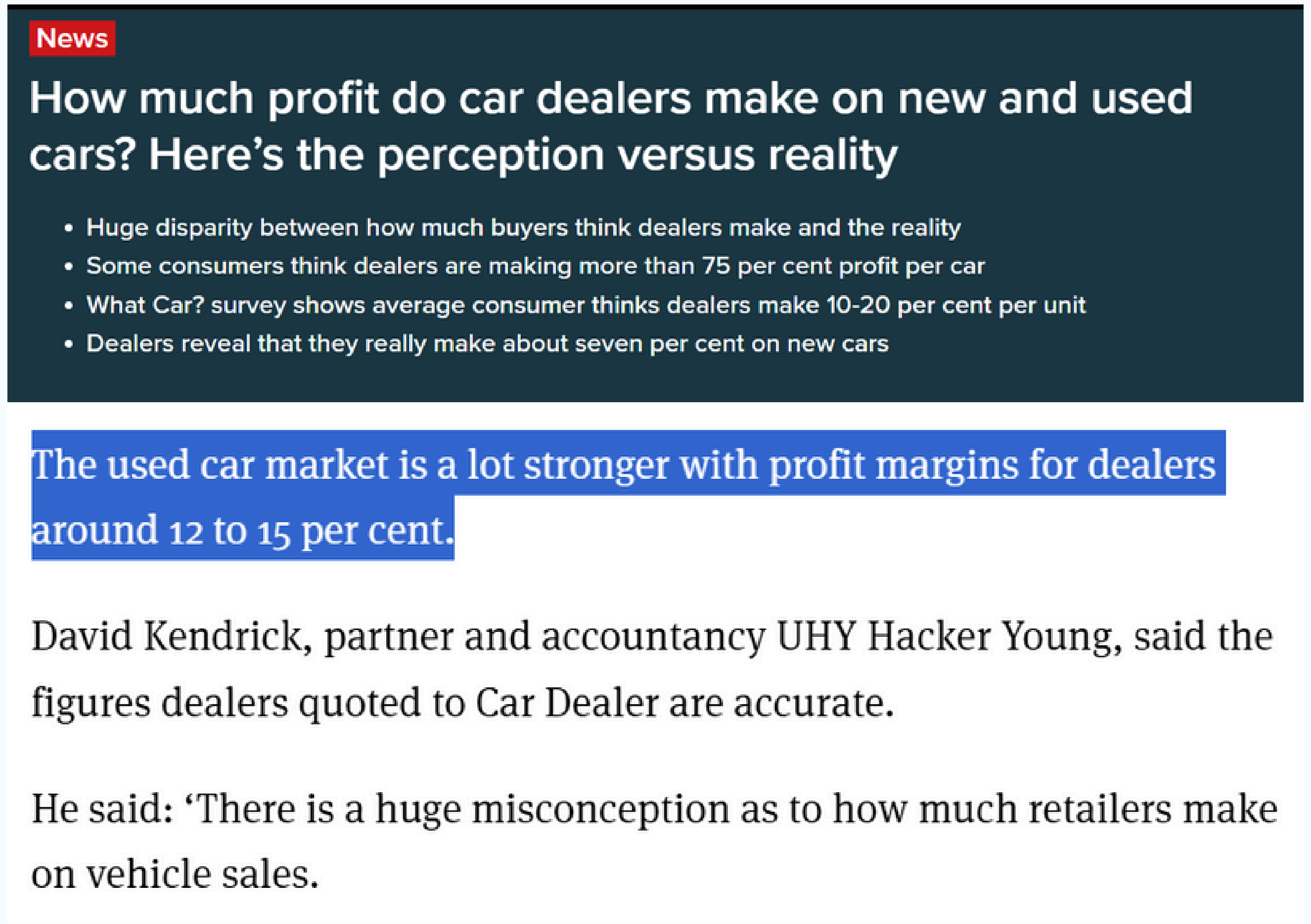
Car : Volkswagen - Golf

Bought : 2007

power : 245hp / Gasoline

...

1. Introduction



News

How much profit do car dealers make on new and used cars? Here's the perception versus reality

- Huge disparity between how much buyers think dealers make and the reality
- Some consumers think dealers are making more than 75 per cent profit per car
- What Car? survey shows average consumer thinks dealers make 10-20 per cent per unit
- Dealers reveal that they really make about seven per cent on new cars

The used car market is a lot stronger with profit margins for dealers around 12 to 15 per cent.

David Kendrick, partner and accountancy UHY Hacker Young, said the figures dealers quoted to Car Dealer are accurate.

He said: 'There is a huge misconception as to how much retailers make on vehicle sales.

Dealers benefit from the difference between the selling price and the buying price of a used car

➡️ **Sellers** want to maximize the selling price to obtain a greater profit, and still manage to sell the car

1. Introduction

BAIN & COMPANY Industries Consulting Services Digital Insights About Careers Explore

Brief

The Outlook for the European Used Car Market

Growth opportunities may be closer than they appear.

By Roch Baranowski, Eric Zayer, Klaus Stricker, and Ingo Stein
10 min read

With a total volume of 429 billion euros in 2021, the market for used cars in Europe is similar to the market for new cars. Of the 32 million used cars sold in Europe in 2021, some 44% were sold privately in consumer-to-consumer (C2C) transactions, while 56% were sold by professional retailers, which tend to focus on higher-class, younger cars. Furthermore, used car margins typically beat new car margins for dealers. Overall, the used car market is growing healthily, with a CAGR of 7% between 2015 and 2021.

Private Transactions are not easy to investigate with official statistics

➡️ **Buyers** struggle to determine if the listing price on the used car site is accurate or overpriced😭

1. Introduction

Let's Clarify the concept!

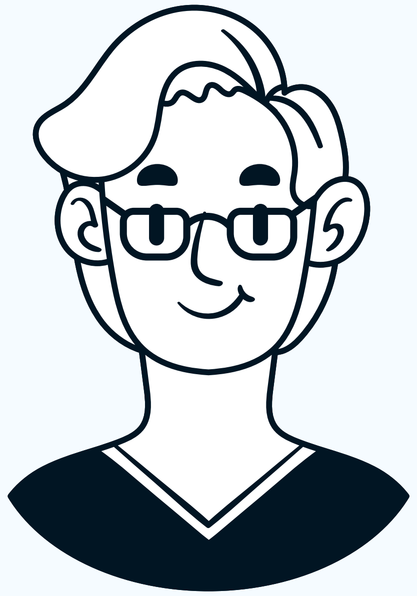
1. Listing price : the price that seller has set on the used car site

2. Final Listing Price : "Final registered price" after multiple adjustments

3. Selling price : the price that seller receives - the price that a seller sets in a used car market

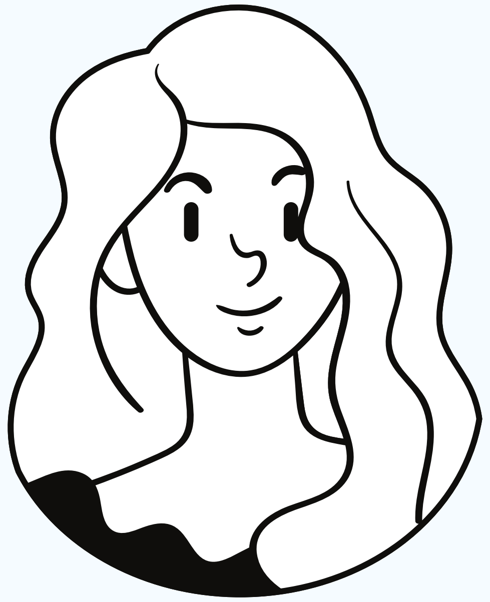
4. Buying price : the total price customer 'actually' pays for the product (may include taxes and other fees)

2. Scenario Example - Target Audience



Daniel Brühl - Seller

- I’m trying to sell my original car in order to buy a new car!
- But, I’m afraid of the dealer fraud.
- Can’t I just **sell this privately** at an **“Appropriate Price”**?



Maria Stein - Buyer

- I want to buy used car which is cheaper because I have a limited budget.
- I have to find a used car in private sale, without a dealer.
- How much is the **“common selling price”** of my dream car?

3. Our Dataset

Source of data : ebay or other sites where cars can be posted.

21 features

x

371,528 data samples

Vehicle Condition

- vehicleType
- yearOfRegistration
- gearbox
- powerPS
- model
- kilometer
- monthOfRegistration
- fuelType
- brand
- notRepairedDamage

car_age

Selling

- name
- seller
- price
- nrOfPictures
- postalCode

Crawling related

- dateCrawled
- dateCreated
- lastSeen

Determining the **last price** among duplicate data

3. Our Dataset - Data Preprocessing

dateCreated	dateCrawled	lastSeen	name	price	vehicleType	yearOfR	gearbox	pow	model	kilomet	month	fuelType	brand
2016-04-04 0:00	2016-04-05 0:37	2016-04-07 3:17	***_ACHTUNG_FORD_KA_MIT_NEU_TÜV_BIS_04/2018_ZU_VERKAUFEN_***	1350	kleinwagen	2002	manuell	60	ka	150000	8	benzin	ford
2016-04-05 0:00	2016-04-05 8:49	2016-04-07 11:17	***_ACHTUNG_FORD_KA_MIT_NEU_TÜV_BIS_04/2018_ZU_VERKAUFEN_***	1350	kleinwagen	2002	manuell	60	ka	150000	8	benzin	ford
2016-04-05 0:00	2016-04-05 6:55	2016-04-07 10:44	***_ACHTUNG_FORD_KA_MIT_NEU_TÜV_BIS_04/2018_ZU_VERKAUFEN_***	1350	kleinwagen	2002	manuell	60	ka	150000	8	benzin	ford
2016-04-04 0:00	2016-04-04 22:54	2016-04-07 2:17	***_ACHTUNG_FORD_KA_MIT_NEU_TÜV_BIS_04/2018_ZU_VERKAUFEN_***	1350	kleinwagen	2002	manuell	60	ka	150000	8	benzin	ford
2016-04-04 0:00	2016-04-05 1:55	2016-04-07 10:16	***_ACHTUNG_FORD_KA_MIT_NEU_TÜV_BIS_04/2018_ZU_VERKAUFEN_***	1350	kleinwagen	2002	manuell	60	ka	150000	8	benzin	ford
2016-04-04 0:00	2016-04-04 22:50	2016-04-07 2:17	***_ACHTUNG_FORD_KA_MIT_NEU_TÜV_BIS_04/2018_ZU_VERKAUFEN_***	1350	kleinwagen	2002	manuell	60	ka	150000	8	benzin	ford

Data with the most **recent dateCreated** date
= **final listing price**

- Datasample having Null data that has been deleted.
- Duplications have been deleted if some of the feature’s values were the same.
- Range of price was limited from 500 to 245,000.
- Range of powerPS was limited from 30 to 800.

4. Previous research analysis

Table 1: Performance Matrix of Linear Regression.

	Mixed Strategy	Label Strategy
Training MSE	43.73	54.26
Training R-squared	0.69	0.62
Test MSE	44.02	54.23
Test R-squared	0.70	0.63

Table 2: Performance Matrix of Decision Tree.

	Mixed Strategy	Label Strategy
Training MSE	0.02	0.02
Training R-squared	1.00	1.00
Test MSE	23.92	26.40
Test R-squared	0.84	0.82

J. He, "Predicting Vehicle Prices Using Machine Learning: A Case Study with Linear Regression," Applied and Computational Engineering, vol. 99, no. 1, pp. 35–42, Nov. 2024

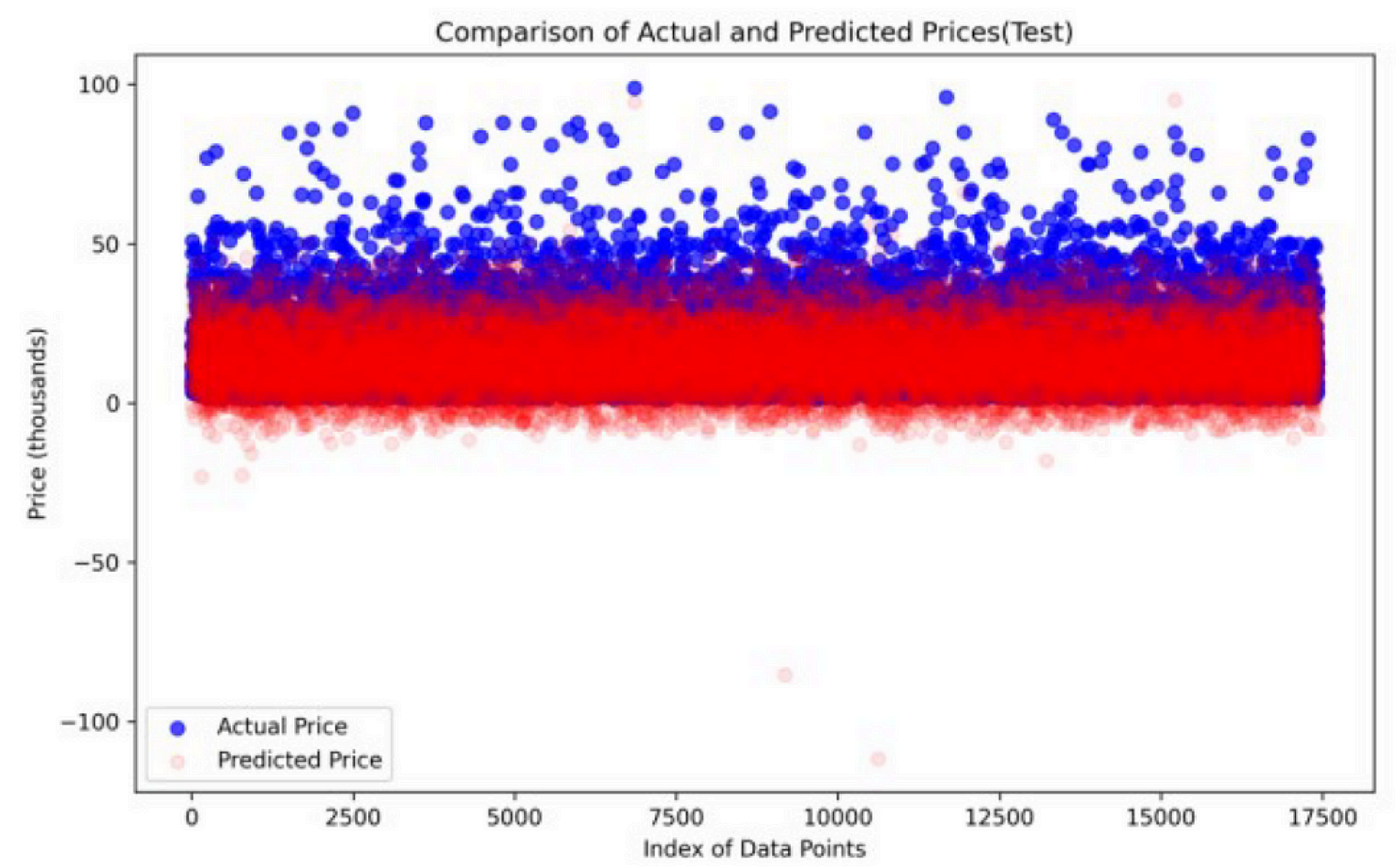
Overfitting -> random forest?

Mixed strategy - using one-hot encoding & manual mapping(label-encoding)

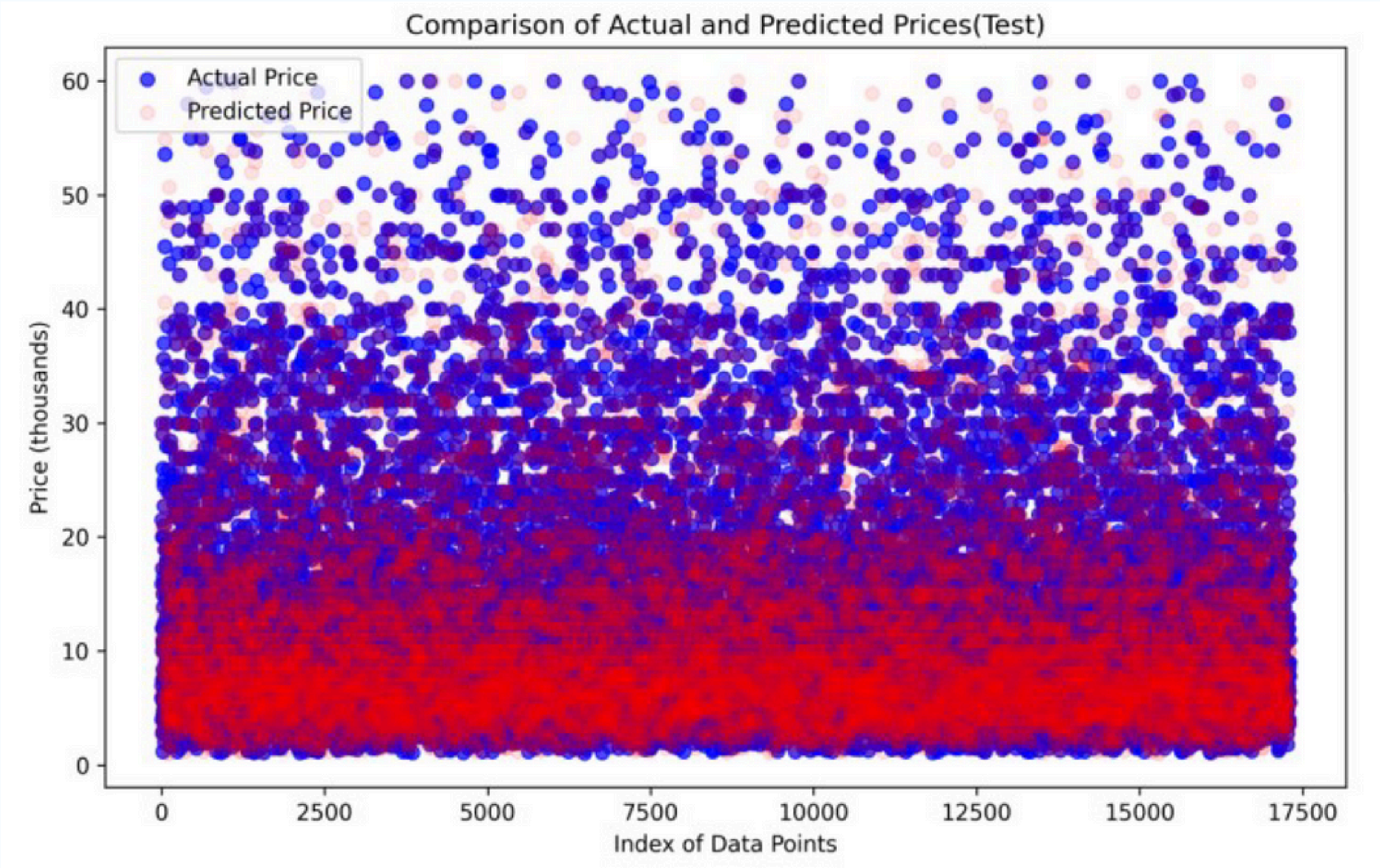
Label strategy - using only label-encoding

4. Previous research analysis

Distribution using **linear regression**



Distribution using **decision tree**



J. He, "Predicting Vehicle Prices Using Machine Learning: A Case Study with Linear Regression," Applied and Computational Engineering, vol. 99, no. 1, pp. 35–42, Nov. 2024

Low & medium prices：High Accuracy in both models

High price(luxury cars)：Accuracy declined

5. Model Works - Multiple Linear Regression (MLR)

 # Nominal values for MLR : Label encoding

Q. Why not One-hot Encoding?

A. Experienced the **Curse of Dimensionality**...😭

Fixed) Linear correlation > 0.5 : car_age, kilometer, powerPS, car_model, car_brand

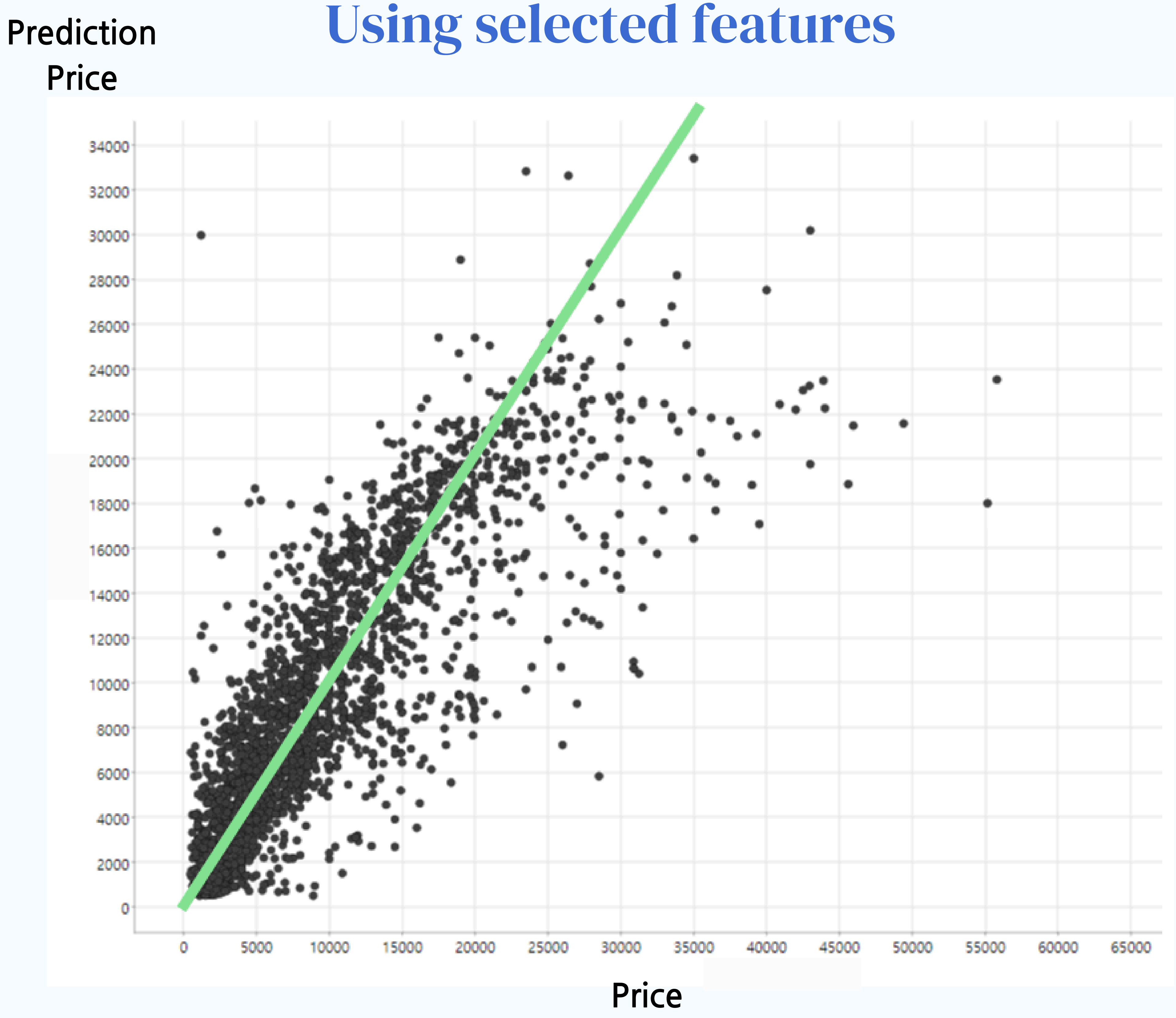
vehicle_Type(1), gearbox(2), fuel_Type(3), notRepairedDamage(4)

CASE	x	1, 3, 4	2, 3, 4	1, 2, 3, 4
R^2	0.63506842	0.648601815	0.656494454	0.655735274
mean absolute error	3174.356195	3132.950155	3118.916731	3124.028734
mean squared error	25281779.31	24440843.29	23588280.12	23578420.14
root mean squared error	5028.098976	4943.768127	4856.776721	4855.761541
mean signed difference	-5.4771E-10	-4.02865E-09	4.30216E-10	4.01539E-10
mean absolute percentage error	0.906901314	0.903172851	0.910689249	0.899603956
adjusted R^2	0.63506842	0.648601815	0.656494454	0.655735274

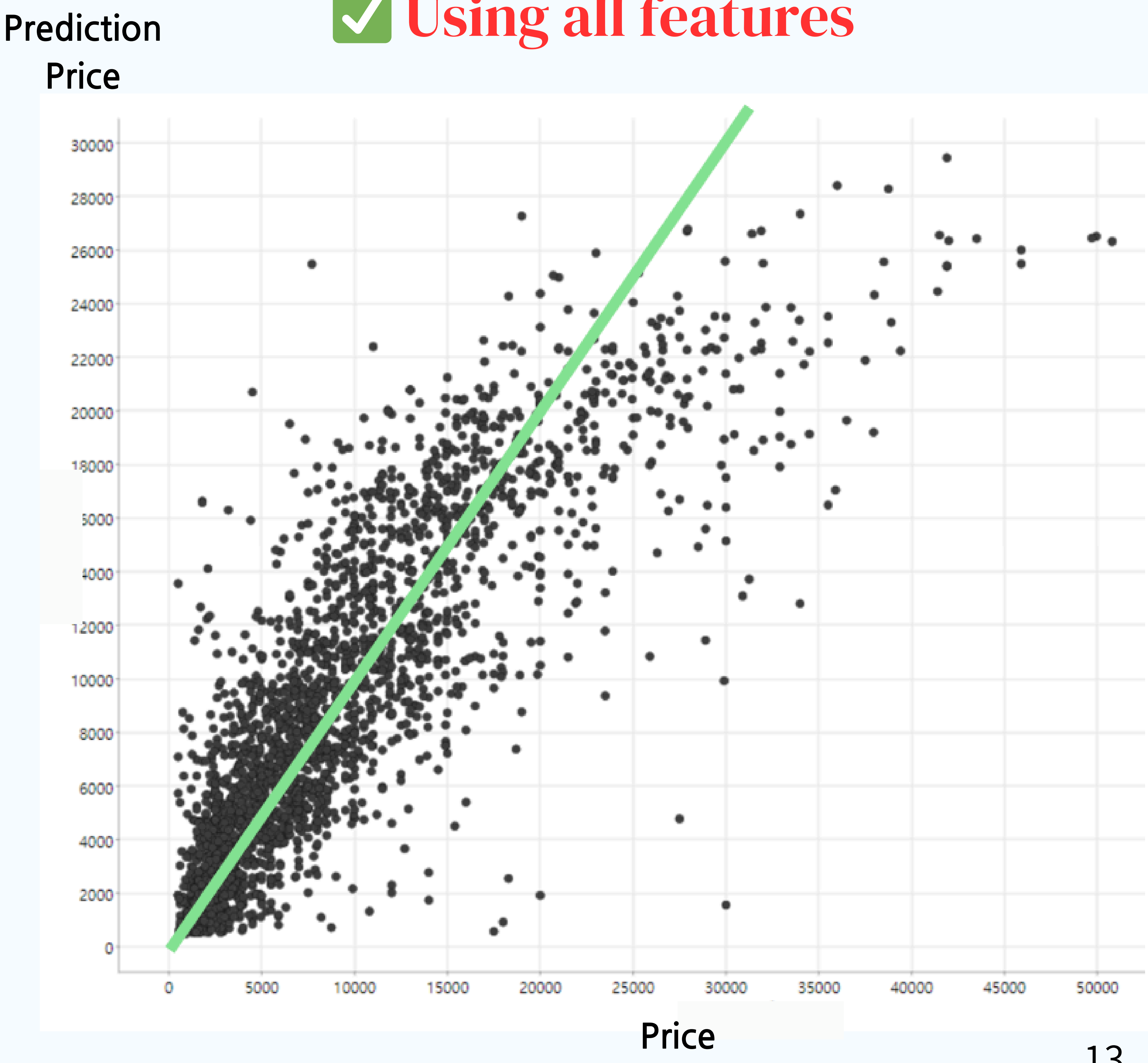
Using selected features **Using all features**

5. Model Works - Multiple Linear Regression (MLR)

Using selected features



✅ Using all features



5. Model Works - Random Forest

Q1. Why Random Forest?

A. It's an ensemble model of individual decision trees, which is :

- Good at catching **nonlinear relationships**
- Good at handling **categorical variables** well
- Can effectively reflect interactions between variables
- Automatically calculates feature importance -> **use important variables**

Q2. Why not MLP?

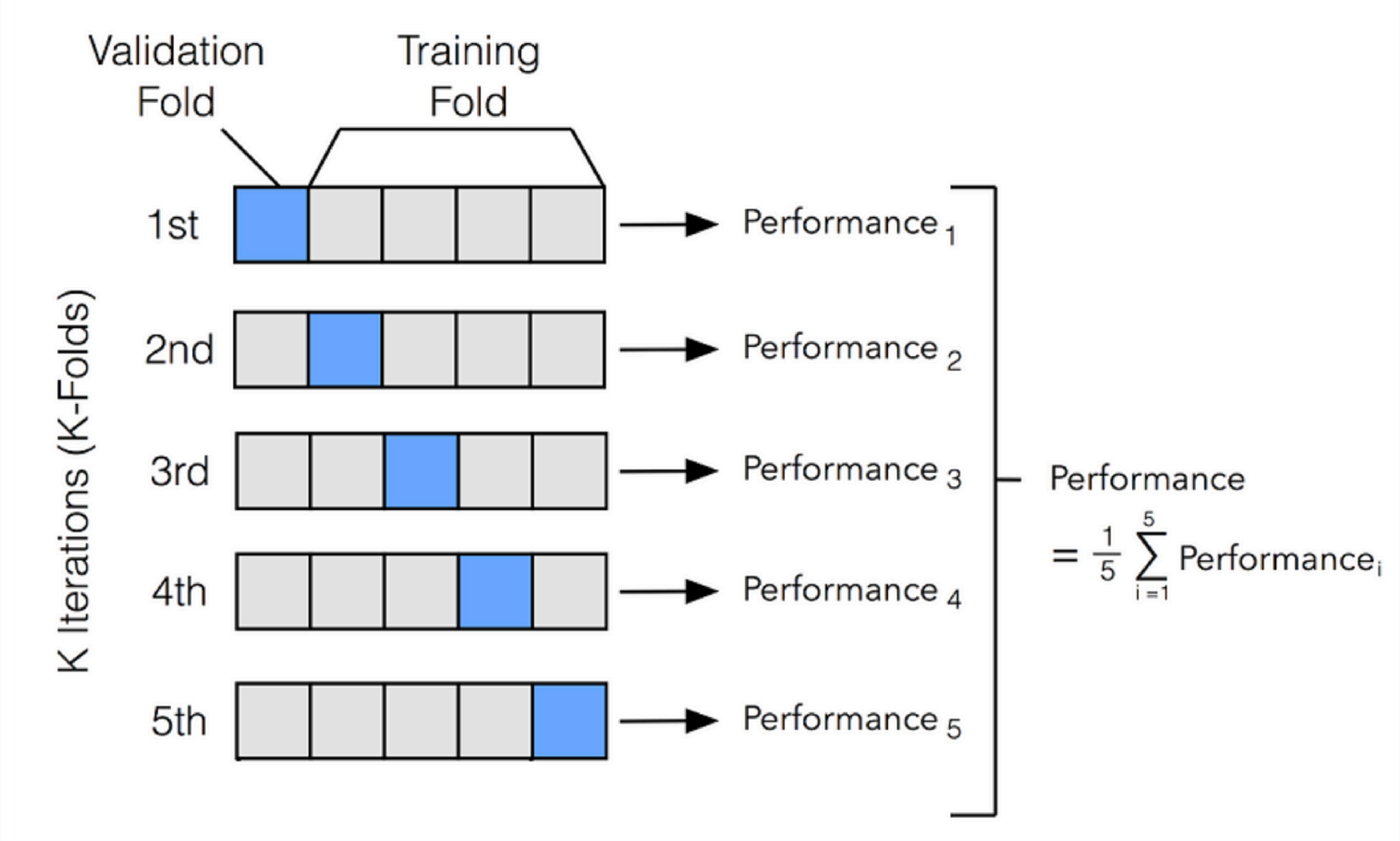
A. Without sufficient training data and adequate hyperparameter tuning, MLP may not capture nonlinearity well

- Requires **learning all its characteristics** & consideration of **non-critical variables** => Can lead to a lot of unnecessary learning!

5. Model Works - Random Forest

Random Forest Overfitting Prevent method:

K-fold validation



k-value: 5

80,000 / 5 = 16,000

Our dataset: about 80,000 data samples

➡️ set the k-value 5 to adjust for the sufficient amount of data to train the model

5. Model Works - Random Forest

⭐Target Encoding..?

Convert brands and models to average prices

car_brand	car_model	car_brand_enc	car_model_enc	price
volkswagen	golf	6819.045426	6024.54069	600
volkswagen	golf	6819.045426	6024.54069	1459
volkswagen	passat	6819.045426	6013.214716	5890
volkswagen	golf	6819.045426	6024.54069	4000
volkswagen	touareg	6819.045426	17980.71884	11900
volkswagen	beetle	6819.045426	8735.380631	5800
volkswagen	fox	6819.045426	2997.835766	7800
volkswagen	golf	6819.045426	6024.54069	19200
volkswagen	golf	6819.045426	6024.54069	1200
volkswagen	golf	6819.045426	6024.54069	3799
volkswagen	golf	6819.045426	6024.54069	8990
volkswagen	golf	6819.045426	6024.54069	1750
volkswagen	passat	6819.045426	6013.214716	5100

5. Model Works Random Forest- UnEncoded

train

min child node size

Tree Depth

tree depth	5	6	7	8	9	10	11	12	13	14	15
5	5073.83444	5087.78231	5085.24237	5044.77059	5051.80077	5036.44406	5022.2113	4940.41466	4976.87182	4931.20889	4969.2733
6	4647.40643	4637.04533	4648.1111	4645.80143	4610.97826	4596.37018	4576.86269	4579.94708	4568.39861	4547.30093	4551.12785
7	4286.65256	4291.06829	4299.23	4260.6506	4270.48117	4261.28607	4239.92579	4220.29771	4215.42981	4212.62722	4199.9638
8	3994.76992	3982.01599	3993.32079	3981.92781	3972.82208	3974.18785	3967.54015	3952.15442	3963.27206	3933.94239	3935.76502
9	3733.21848	3750.72231	3730.26372	3737.90289	3745.42784	3726.41265	3734.87151	3725.85411	3732.43839	3718.6797	3717.60498
10	3529.24836	3535.1556	3542.33743	3551.38359	3547.23559	3547.3813	3552.33057	3558.90025	3554.43746	3553.8165	3559.67442
11	3369.0179	3376.81598	3387.28097	3396.67537	3415.32879	3420.32519	3420.40687	3425.74612	3438.22344	3441.36099	3446.5621
12	3241.43168	3271.5179	3290.04411	3298.97172	3312.27157	3324.49011	3334.09417	3346.94211	3358.9221	3372.3531	3368.35463
13	3162.41366	3189.48057	3212.15548	3230.93479	3249.24921	3257.5106	3275.4621	3291.73784	3302.7549	3311.57033	3328.27638
14	3102.9982	3131.71174	3161.94235	3185.3263	3200.27528	3218.44648	3233.81834	3258.0882	3267.40402	3289.86268	3305.50835
15	3061.53513	3089.42134	3128.66421	3154.54102	3184.99155	3198.97043	3216.26569	3241.30883	3253.56628	3271.72241	3283.77644
16	3038.66982	3072.46229	3119.09173	3132.21421	3158.74277	3191.21771	3203.00865	3221.43159	3239.0058	3255.47362	3278.08264
17	3035.4788	3064.05504	3101.71589	3119.48314	3155.9946	3172.33532	3199.67726	3217.54133	3238.79261	3252.15388	3269.51797
18	3015.7388	3057.21754	3087.70831	3116.95178	3144.22538	3177.0075	3193.60756	3215.27044	3228.00326	3248.48157	3262.60705
19	3009.70148	3052.16202	3085.69582	3114.65252	3148.27473	3170.77217	3189.00195	3216.88805	3228.70332	3253.28522	3260.84319
20	3008.2095	3050.85801	3086.81845	3112.86636	3152.31314	3168.34755	3188.81149	3214.53585	3227.72573	3245.47777	3265.29016
21	3000.67154	3061.27253	3083.8327	3115.58255	3139.97258	3172.88853	3195.33989	3208.75505	3223.69141	3246.40279	3258.06007
22	3004.03584	3049.22275	3093.32643	3122.96191	3145.98031	3169.79873	3194.29973	3213.42431	3231.49244	3250.78816	3270.25276
23	3012.44444	3046.83626	3083.93522	3116.89608	3137.21002	3171.00817	3187.88489	3206.87074	3226.35983	3243.66767	3268.39063
24	3004.37493	3038.71688	3081.16252	3118.21506	3131.76292	3166.46486	3197.95679	3212.23677	3224.7554	3252.23578	3266.77799
25	3010.44305	3051.95382	3084.03992	3119.33657	3150.52851	3167.03154	3186.27009	3209.68453	3232.00133	3255.09312	3265.93106
26	3013.96995	3044.68477	3084.43694	3112.13656	3146.67059	3170.51559	3185.20054	3211.34698	3232.56138	3246.59824	3266.92676
27	3010.08922	3051.82265	3082.57837	3118.07079	3145.00023	3157.70852	3191.64799	3209.84602	3229.52443	3236.01902	3261.01255
28	3006.54145	3049.09129	3088.95345	3108.38729	3142.97688	3163.5209	3189.26531	3211.99551	3220.0202	3256.10877	3262.85876
29	3006.95744	3042.70042	3087.14048	3110.67243	3135.62192	3167.84852	3184.85411	3212.58475	3229.79439	3252.35345	3270.33652
30	3006.7741	3038.68252	3084.66253	3113.41696	3138.40477	3174.73642	3189.58678	3213.00669	3232.10213	3253.25515	3272.05503

test

min child node size

Tree Depth

tree depth	5	6	7	8	9	10	11	12	13	14	15
5	5093.47771	5109.62795	5106.24872	5063.58482	5071.26392	5054.3477	5043.28366	4963.52804	4998.25032	4950.50096	4991.23021
6	4680.67791	4671.16165	4681.16953	4682.34446	4644.18712	4629.01604	4614.12403	4614.52255	4601.31609	4579.42669	4582.98891
7	4337.76804	4344.19437	4354.50249	4311.49435	4320.8563	4316.31406	4286.92099	4270.53968	4266.54417	4261.78976	4252.52373
8	4080.16097	4059.46884	4075.90732	4052.88733	4038.49405	4047.4435	4037.21588	4017.67734	4031.99517	4006.34606	4001.82316
9	3837.02517	3859.02027	3838.91911	3835.71708	3840.61894	3817.44119	3822.23436	3807.10305	3816.35368	3806.80277	3807.53992
10	3673.5571	3670.74591	3672.08289	3673.5668	3660.69495	3666.66953	3663.45594	3664.05112	3664.04457	3654.97334	3656.51459
11	3534.26907	3535.05772	3543.1302	3543.93062	3551.09499	3558.6598	3545.59387	3548.77584	3556.45728	3561.73253	3555.82206
12	3443.73527	3449.84617	3455.00377	3456.33513	3460.00625	3465.71269	3478.35871	3487.25453	3493.19795	3492.19636	3490.78558
13	3373.40715	3391.27326	3392.86655	3401.365	3409.32018	3421.21469	3423.04055	3425.75864	3444.64247	3450.73978	3459.67467
14	3331.43707	3334.0127	3355.88946	3368.89879	3375.54895	3387.56466	3387.04664	3408.5984	3413.14728	3426.03688	3436.77229
15	3302.22637	3305.30342	3335.44474	3339.73756	3356.83001	3370.11938	3380.45864	3393.86067	3404.08069	3402.78334	3424.02275
16	3279.70356	3304.50721	3318.08862	3329.75259	3340.48747	3356.58512	3372.46615	3371.88733	3389.26525	3397.24868	3413.02665
17	3284.51907	3300.60914	3311.48173	3311.42345	3338.09722	3359.6985	3364.8549	3377.87255	3387.2519	3395.27921	3411.30409
18	3263.78099	3281.73814	3300.01274	3319.49509	3329.35701	3351.6599	3366.37679	3377.16981	3380.09196	3391.502	3404.53193
19	3264.46511	3287.39162	3303.72491	3311.02813	3337.71025	3353.58877	3350.53496	3376.65431	3382.56478	3395.14178	3400.77209
20	3271.96764	3294.80225	3307.26351	3313.44267	3332.46753	3347.96015	3354.7165	3369.40298	3383.27171	3393.95814	3401.44157
21	3270.48203	3299.8496	3300.03276	3312.96723	3330.9407	3348.24316	3354.99309	3364.59504	3374.62207	3398.13076	3391.29174
22	3264.17155	3275.9566	3314.87641	3326.66326	3329.93637	3346.28239	3362.87239	3367.725	3384.5016	3398.0871	3406.78118
23	3266.24493	3280.75883	3298.43235	3313.54974	3337.3832	3349.33532	3350.27213	3361.5134	3377.8521	3388.17196	3406.61434
24	3270.49148	3275.45112	3304.90942	3317.39314	3322.71257	3343.4402	3363.02126	3370.66467	3375.03526	3396.30248	3409.56203
25	3273.55845	3279.96193	3300.01667	3323.07215	3335.6864	3347.63704	3361.70353	3362.71317	3382.04699	3401.97518	3403.8082
26	3270.1202	3279.12595	3297.97358	3316.11863	3332.73596	3348.71876	3354.97296	3369.31893	3387.51886	3401.34329	3407.22835
27	3267.71373	3300.13122	3293.05863	3316.79336	3330.44077	3338.13372	3360.5385	3365.51531	3379.96571	3385.01038	3406.66746
28	3257.10923	3291.48586	3306.7723	3308.36474	3329.28972	3341.4195	3357.96373	3369.85321	3380.10069	3398.77737	3402.7839
29	3261.13332	3294.11766	3303.18545	3309.24626	3330.24269	3345.36613	3353.69786	3373.73174	3382.046	3397.29149	3411.1507
30	3267.25544	3276.41766	3306.67593	3321.46739	3334.02764	3348.43867	3358.96828	3374.52829	3383.48752	3399.58586	3412.41225

↓ ↑

Limit number of levels (tree depth)

Number of tree levels to be learned.

Minimum child node size

Minimum number of records in child nodes

Limit number of levels (tree depth)	Minimum child node size
28	5

5. Model Works Random Forest- Encoded

train

min child node size

Tree Depth

tree depth	5	6	7	8	9	10	11	12	13	14	15
5	4818.07788	4798.85384	4792.85414	4775.32751	4764.28058	4742.17724	4742.51217	4690.05119	4647.50655	4658.67812	4633.00104
6	4392.93992	4357.27641	4337.42951	4309.14333	4309.2915	4290.12658	4265.98161	4271.79674	4240.64881	4212.90987	4190.66529
7	3962.47287	3959.12396	3964.03747	3943.95223	3924.96359	3912.83526	3892.23794	3873.57289	3864.7435	3834.5223	3837.85617
8	3615.67101	3611.92015	3607.19632	3599.9575	3597.45639	3591.51558	3574.75282	3554.36416	3548.49978	3540.75289	3527.01958
9	3320.461	3324.27579	3319.473	3317.10312	3314.25298	3300.66986	3310.96554	3298.26513	3296.89361	3280.0808	3281.21153
10	3072.03911	3085.60565	3091.87808	3093.40777	3098.443	3096.71324	3100.16233	3103.25928	3096.27408	3105.62084	3101.14879
11	2884.01905	2894.23212	2907.1694	2926.18873	2928.72416	2939.04727	2946.34289	2951.67555	2957.57513	2960.9933	2964.09787
12	2728.4797	2756.05917	2777.7173	2787.16041	2804.7805	2817.35007	2830.40446	2846.32437	2858.65539	2863.32102	2877.72524
13	2616.76784	2646.73326	2673.27862	2697.59475	2719.31031	2739.15407	2756.18451	2773.42057	2785.37907	2801.08408	2817.13227
14	2524.75486	2566.79032	2601.79697	2631.36178	2656.49242	2681.3843	2700.66439	2720.64494	2738.45098	2763.10525	2781.53019
15	2469.03421	2510.10725	2546.58926	2580.03393	2611.66597	2639.82879	2661.06136	2687.38118	2705.80698	2734.2942	2753.82785
16	2421.45888	2468.12544	2507.33748	2547.36125	2583.46461	2613.32648	2640.47883	2663.98161	2689.90183	2731.5917	2743.5147
17	2382.11862	2433.07078	2482.06718	2524.37609	2563.27235	2594.22814	2624.09708	2651.22838	2677.2645	2714.06204	2739.65057
18	2360.7219	2412.93245	2465.29366	2510.85002	2547.89885	2582.83706	2615.0743	2642.23821	2668.1532	2710.626	2729.51151
19	2339.81381	2400.38974	2457.79444	2499.14523	2542.81312	2578.59453	2609.49987	2636.05411	2663.6239	2706.16098	2730.63889
20	2325.91565	2391.59623	2449.13783	2495.46479	2534.61455	2571.13674	2605.31407	2635.34916	2661.68412	2701.81394	2727.44473
21	2318.11397	2388.4458	2446.42632	2491.93836	2534.25664	2569.32505	2606.45372	2632.39052	2660.08156	2698.72619	2714.79999
22	2317.55012	2385.32872	2445.48215	2491.37193	2532.9214	2570.17797	2604.67802	2631.24698	2659.87061	2691.09292	2716.00959
23	2315.87833	2384.92538	2440.538	2491.39055	2531.88079	2570.35492	2601.80294	2632.29522	2660.13868	2691.29136	2714.37427
24	2312.52883	2381.13618	2440.53687	2485.79238	2532.38722	2573.76748	2601.56388	2632.41991	2660.89471	2681.93376	2703.59646
25	2311.83551	2385.63372	2443.20856	2492.03475	2535.91523	2570.08029	2603.44054	2634.31059	2656.65218	2683.09484	2710.2458
26	2312.89418	2383.84581	2442.61071	2489.49499	2532.42853	2573.11506	2600.12228	2632.60757	2657.39894	2685.6635	2708.55304
27	2315.08513	2382.17366	2441.78087	2487.00377	2532.81006	2569.53474	2598.91402	2634.23275	2659.71217	2681.86574	2708.07703
28	2313.69156	2384.11963	2438.6795	2490.28429	2535.57383	2568.32382	2603.00395	2629.13752	2659.615	2681.56985	2704.24507
29	2306.37113	2383.39346	2441.64606	2490.79696	2530.22183	2568.51047	2603.53321	2633.94294	2655.93796	2682.45182	2706.62498
30	2311.77449	2384.12248	2438.00112	2492.45556	2530.31913	2568.97137	2602.78685	2630.62556	2658.50297	2683.03325	2707.22513

test

min child node size

Tree Depth

tree depth	5	6	7	8	9	10	11	12	13	14	15
5	4851.16918	4832.34124	4826.81993	4811.19276	4796.81424	4776.75688	4779.16594	4729.15366	4689.54189	4697.64515	4668.44441
6	4454.63149	4417.86793	4398.1396	4375.4893	4369.70538	4353.09279	4332.16683	4333.47893	4301.45592	4272.27792	4259.85356
7	4064.94039	4048.74285	4063.12367	4032.78813	4021.74805	4008.93185	3991.69304	3962.81526	3960.32257	3932.02392	3922.7631
8	3768.03768	3766.31467	3755.72771	3745.90309	3754.38619	3712.88864	3709.29398	3691.22919	3675.63231	3672.7318	3640.62649
9	3526.9668	3515.26386	3513.03229	3483.67927	3479.60412	3479.40937	3483.74859	3464.15052	3461.37977	3460.97145	3443.22282
10	3332.66317	3329.85753	3324.28927	3325.68499	3316.94826	3309.7671	3301.65814	3297.88206	3284.00508	3287.46221	3292.35911
11	3176.63794	3178.36185	3168.24062	3178.18386	3174.49071	3155.59165	3153.31328	3165.96812	3176.83927	3182.39563	3166.91619
12	3075.42828	3069.59889	3064.41666	3075.23756	3085.95159	3088.55739	3072.51301	3078.27596	3088.20381	3121.5435	3113.09475
13	2985.68693	3005.85168	3010.94824	3025.45945	3024.28949	3021.45406	3024.85164	3034.88349	3042.75617	3054.64502	3056.2698
14	2945.00215	2947.55053	2970.63622	2966.67094	2980.52284	2974.06701	2985.58357	3004.49511	3009.46297	3024.0377	3043.88423
15	2918.66748	2907.10135	2937.9317	2940.45449	2946.61899	2971.13937	2972.80675	2993.41095	2990.92373	3020.70278	3019.2486
16	2881.53344	2896.65968	2920.3992	2924.8638	2927.22239	2965.55755	2957.76887	2970.30231	2983.0266	3007.47767	3013.36043
17	2868.62129	2892.04735	2890.37844	2921.39442	2920.30477	2927.22582	2956.93029	2968.47642	2984.27023	2991.68494	3005.43738
18	2864.21317	2859.69187	2870.46011	2906.15832	2920.94585	2950.66555	2961.23569	2961.32455	2980.06324	3006.77896	3000.80641
19	2855.49245	2877.11559	2883.79604	2903.78412	2917.77269	2929.46852	2947.24729	2943.65869	2973.19236	2990.87297	3004.50111
20	2848.76065	2877.54014	2895.93997	2916.20028	2927.28032	2925.32095	2939.00164	2961.69507	2969.65242	2967.12954	3016.57003
21	2842.443	2870.92814	2889.09988	2891.51461	2909.9663	2930.5388	2942.89817	2959.60763	2953.48103	2994.52101	3001.72957
22	2839.66467	2868.06346	2859.10157	2898.86929	2895.46923	2921.0529	2938.36629	2952.34466	2964.14046	2979.10852	3001.90824
23	2844.38335	2853.2714	2890.31164	2899.66415	2908.38896	2928.28769	2945.74195	2940.45841	2968.85892	2976.21413	2991.89435
24	2842.0512	2864.46708	2881.10804	2901.59236	2915.6707	2929.32418	2921.74622	2940.62856	2963.00353	2969.75063	2987.85942
25	2846.62885	2851.99017	2876.22392	2896.59154	2919.58425	2925.81108	2945.24938	2949.34277	2972.25105	2973.91624	2996.45474
26	2835.1087	2863.52067	2907.02675	2886.57563	2918.09569	2922.38848	2953.12682	2955.5384	2969.38356	2975.97283	2983.38757
27	2857.77756	2857.2184	2884.1374	2907.25164	2905.58348	2911.25448	2948.15367	2965.21411	2966.97566	2974.73742	2988.6198
28	2840.57728	2850.40479	2892.7461	2896.58592	2911.41308	2915.51807	2930.86261	2965.68711	2964.66637	2975.98544	2981.38231
29	2850.65981	2858.09361	2867.0724	2909.39039	2925.35934	2927.98936	2930.85388	2942.78304	2963.61115	2986.76765	2986.58599
30	2840.58614	2852.90251	2868.04554	2899.64578	2907.15824	2911.93102	2940.97828	2967.33532	2971.37519	2979.6776	2988.285

Limit number of levels (tree depth)

Number of tree levels to be learned.

Minimum child node size

Minimum number of records in child nodes

Limit number of levels (tree depth)	Minimum child node size
26	5

6. Model Results

Original data (un-encoded)

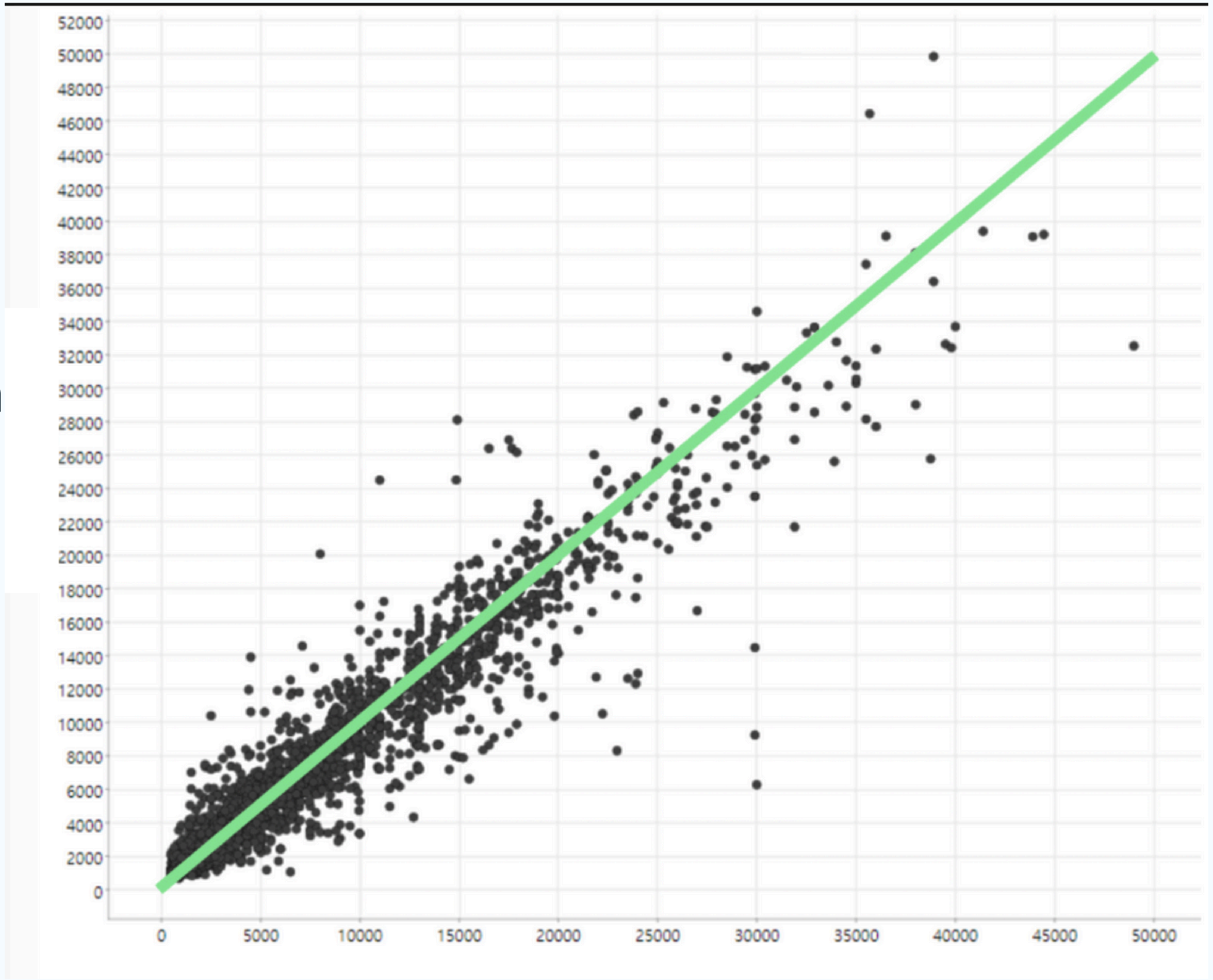
performance	prediction(price)
R^2	**0.847**
MAPE(Mean Absolute Percentage Error)	**0.455**
RMSE(Root Mean Squared Error) [€]	**3257.109**

Target Encoded data

performance	prediction(price)
R^2	**0.885**
MAPE(Mean Absolute Percentage Error)	**0.319**
RMSE(Root Mean Squared Error) [€]	**2833.331**

6. Model Results

Prediction Price



Price

Q. How to resolve big errors in RMSE?

A. Dividing into Low price / High price

⭐ **Criterion?**

6. Model Results

Low price

performance	prediction(price)
R^2	**0.878**
MAPE(Mean Absolute Percentage Error)	**0.324**
RMSE(Root Mean Squared Error) (€)	**1732.82**

High price

performance	prediction(price)
R^2	**0.602**
MAPE(Mean Absolute Percentage Error)	**0.112**
RMSE(Root Mean Squared Error) (€)	**6190.55**

7. Conclusion

1. Based on the MLR result : Poor performance when the model was turned with label encoded data in MLR...

2. So we used the Random Forest Algorithm!

3. Even though R^2 was high and performed high accuracy, why is the RMSE too high?

4. So even though the MAPE size is similar, the RMSE gets bigger as the price gets higher. -> In Low/medium price, RMSE may be small

Low & medium prices : Predicts Well!
Luxury Cars : Seems to perform low..

5. Divided the low/medium price and the high price
-> it actually decreased from medium to low prices!

Q & A

Thank you for listening!