Assignment 3: Data Analytics, Group 60

188.429 Business Intelligence (VU 4,0) – WS 2022/23

William Amminger
e12229199@student.tuwien.ac.at
12229199
Person A

Zsombor Iszak
e11709501@student.tuwien.ac.at
11709501
Person B

1 BUSINESS UNDERSTANDING

The dataset used in our assignment is from the open source machine learning platform, Kaggle and can be found on this link.

(a) **Scenario:** This dataset contains the latest information on car prices in Australia for the year 2023. It covers various brands, models, types, and features of cars sold in the Australian market. It provides useful insights into the trends and factors influencing the car prices in Australia. The dataset has over 16k records of car listings from various online platforms in Australia.
The business scenario is that a car dealership located in Germany wants to expand and get into the Australian car market. However, before penetrating the market some research on the market needs to be done in order to set up a business model and a pricing strategy.

(b) **Business Objectives:** The 1st business objective of the company is to get an idea about the current pricing in the Australian car market through classical analysis of the dataset. This will help with entering the market to have a better understanding of how to price their cars based on existing ones. The 2nd aim is to provide a model that predicts the probable price of a car based on its characteristics and features. This is helpful after the company has already entered Australia & wants to get an idea of how to price a new vehicle before launching. This in turn would also help with the development of the car based on the expected sale price. The 3rd goal is to make the company more resistant against the possible price fluctuations of changing employees, as usually the pricing is determined by employees with experience in the field. This model can ultimately do similar yet more stable predictions for the company without the need of an experienced employee.

(c) **Business Success Criteria:** This overall picture about the pricing strategies is crucial in order to be able to determine, for what price the cars can be sold. The prediction accuracy and reliability directly effects the selling price, hence impacting the profitability of the company. The aim is to help the dealer to better approximate what is the realistic selling price for given car is.

(d) **Data mining goal:** Extract patterns and dependencies that influence the price of the cars. Finally a regression model needs to be built on top of those patterns, that predicts the future selling price of the cars.

(e) **Data Mining Success criteria:** Come up with a model, that delivers reasonable results based on the input features of a car.

(f) **AI Risk Aspects:** A concern of ours is that both selling price and local parameters appear in the data. Therefore one can run into conclusions and categorizations about the financial state of inhabitants of specific towns. To note, that for example in 450 cases out of 16734 the location is not given. This might be just due to poor data maintenance, however it should be considered, that for some cases it was a sensitive information and was not published intentionally.

2 DATA UNDERSTANDING

(a) **Attribute Types:** For the first view on the data, one can immediately recognise, that most of the columns contain strings, however in those string in many cases numeric information is stored with inclusive units, such as '5 Doors' or '6.7 L / 100 km'. Regarding the missing value it turns out, that beside the missing cells, some cells contain the string '-'. The number of missing values can be seen in the figure 1. Additionally the columns FuelType and BodyType contain a limited number of "Other" string value. In those cases it is also not quite clear what was meant by that value and as the factors covered in our opinion all the possibilities, we assumed, that "Other" means some unique features and is not a homogeneous class by its own, therefore those values were also labeled as NaN.

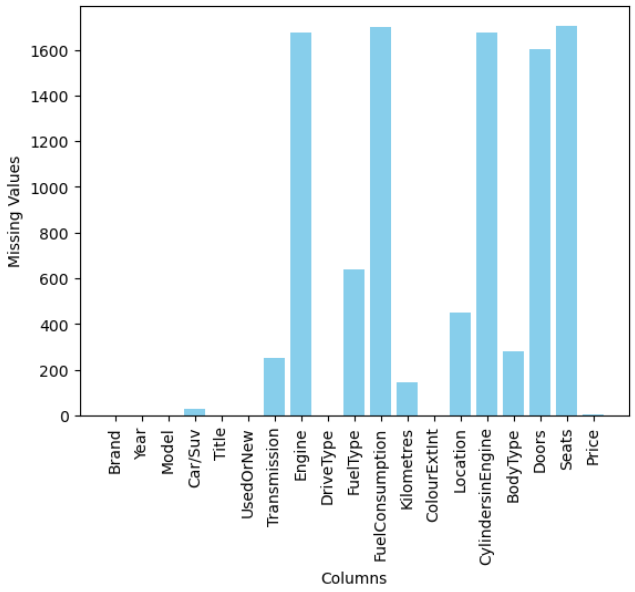

Figure 1: The missing values for each column in the dataset

On a per column basis:

- Brand: `string` - the manufacturer (make) of the vehicle
- Year: `integer` - the model year
- Model: `String` - the model name itself
- Car/Suv: `String` - this column is not quite so clear, as sometimes it contains a custom string about the car type, while usually it mirrors the "BodyType" column
- Title: `String` - full title of the car, including the year, make, model, & sometimes the trim type
- UsedOrNew: `String` - indicates USED, NEW, or DEMO
- Transmission: `String` - Automatic or Manual
- Engine: `String` - specifications in the format # cyl, # L
- DriveType: `String` - indicating which axles have power (AWD, Front, Rear)
- FuelType: `String` - fuel type required by car, such as Diesel, Premium, Unleaded, ...
- FuelConsumption: `String` - efficiency in the format # L / 100 km
- Kilometres: `integer` - current odometer reading
- ColourExtInt: `String` - color in the format (color) / (color), though this data is very heterogenous & sometimes is missing the interior color or contains the trim color
- Location: `String` - location in the format (City), (Territory)
- CylindersinEngine: `String` - number of cylinders in the format # cyl
- BodyType: `String` - body style of the car, such as SUV or Coupe. Most of the information this implies is encoded in other columns
- Doors: `String` - in the format # Doors
- Seats: `String` - in the format # Seats
- Price: `integer` - the target column, being the selling price of the vehicle

(b) **Statistical Properties:** The first step was to determine the number of unique values for each column:

- `Brand`.....76
- `Year`.....45
- `Model`.....781
- `Car/Suv`.....618
- `Title`.....8804
- `UsedOrNew`.....3
- `Transmission`.....3
- `Engine`.....106
- `DriveType`.....5
- `FuelType`.....9
- `FuelConsumption`.....157
- `Kilometres`.....14262
- `ColourExtInt`.....834
- `Location`.....618
- `CylindersinEngine`.....11
- `BodyType`.....10
- `Doors`.....13
- `Seats`.....13
- `Price`.....3794

Another interesting aspect is the correlation matrix, 2, which does not show any immediately promising correlations besides the year & Kilometres. As for the the target variable of "price", no variable correlates above |.35|

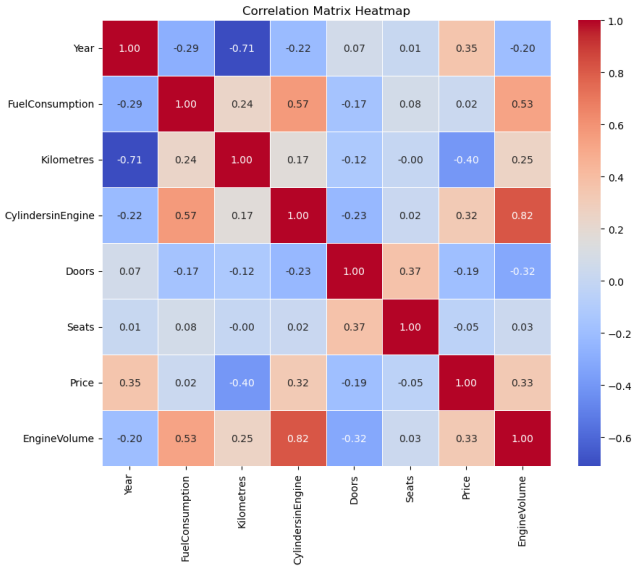

Figure 2: Correlation matrix between numerical columns

(c) **Data Quality:** To explain we took a closer look on examples with missing values but we couldn't identify any specific patterns. The Price column contains in 50 cases the string "POA" and we assume it stands for "Price on Asking", thus the prices is negotiable. However, we converted to column to numbers and "POA" was changed to NaN. After extracting the numeric values, the distribution of the values of the numeric columns is shown in figure 3.
As one can see the distribution plot for the target column Price is hard to read. Our assumption was, that the skewness can be explained by an outlier and by plotting the boxplot (figure 4) one can also show, that truly there is a significant outlier that is responsible for the skewness of the distribution plot. Another helpful visualization is a logarithmic histogram of the prices (figure 5)

(d) **Visual Exploration:** Here it is useful to plot the distributions of some numeric variables. (Brand names 6) (Fuel Types 7) (Transmissions 8) (Used/New 9)

(e) **Ethically Sensitive:** In terms of data representation, there are many classes that are unbalanced, as shown by the previous graphs. There are many more cheap cars than expensive cars for example, which may cause the model to assume that everything is cheaper in order to maintain the distribution at the expense of true accuracy.

(f) **Risks & Bias:** A possible risk is that since a single entry has many values, it is possible to locate the owner of the vehicle. This is why the location was removed as mentioned previously. Since the title also does not contain information usable by machine learning, it will also be dropped.

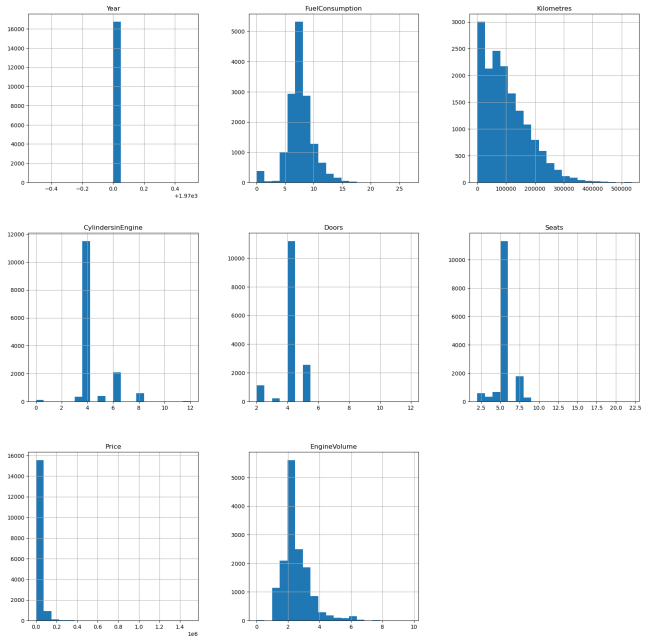

Figure 3: The distribution of the numeric columns

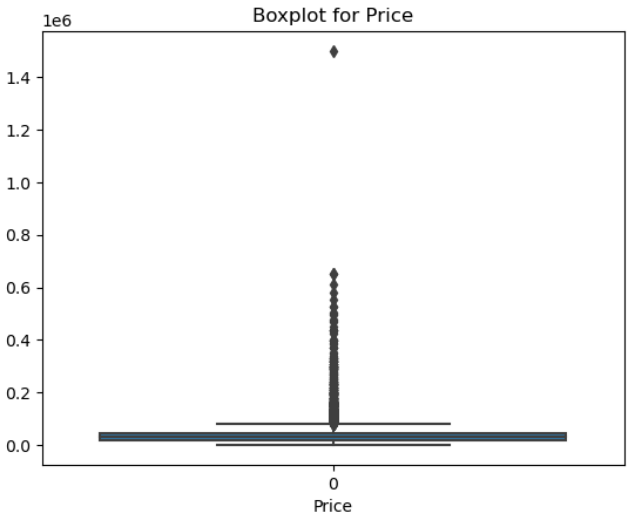

Figure 4: The statistical analysis of the target value

Furthermore, if this was used by people trying to sell their own car, the model could underprice it since the training data has a distribution skewed towards lower prices.

(g) **Required Actions:** One possible solution is normalizing the price information, so that the model receives & predicts a normal distribution. The result could then be reskewed back to the correct prices.

3 DATA PREPARATION

(a) **Necessary Actions:**

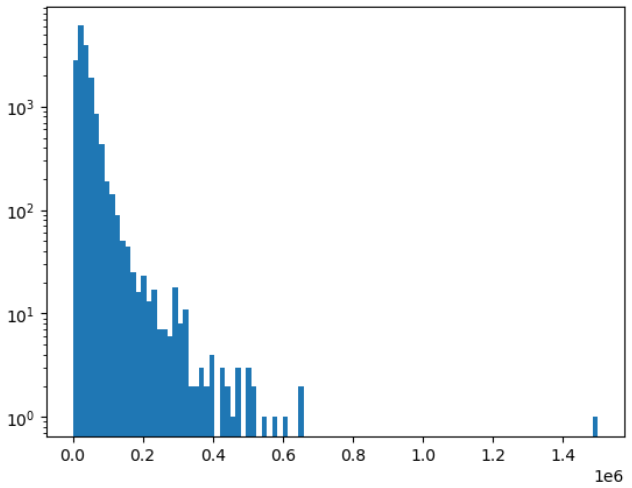

Figure 5: Histogram of the target value

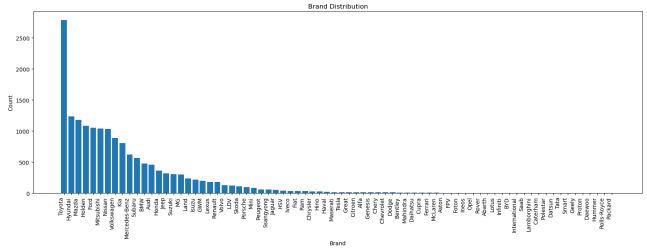

Figure 6: Distribution of brand names

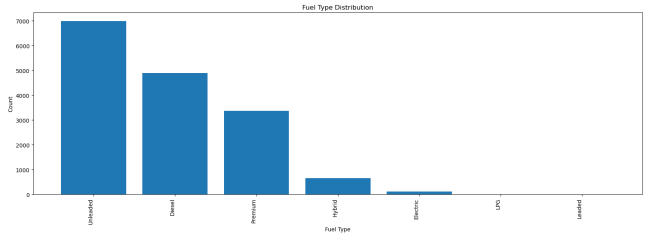

Figure 7: Fuel Type distributions

- A number of operations involved converting string columns into integers with little complexity, such as FuelConsumption, Doors, & Seats, which involved using regex to extract the numerical values.
- The UsedOrNew column contained the values "DEMO", "USED", & "NEW". After some research, we determined that while demo cars are mostly new, they are slightly used by the dealership. Thus, some ordinality can be applied to these values. More specifically, converted Used, Demo, New to 1, 2, 3
- The transmission type was changed to the boolean IsAutomatic column. The "-" values were set to np.nan.

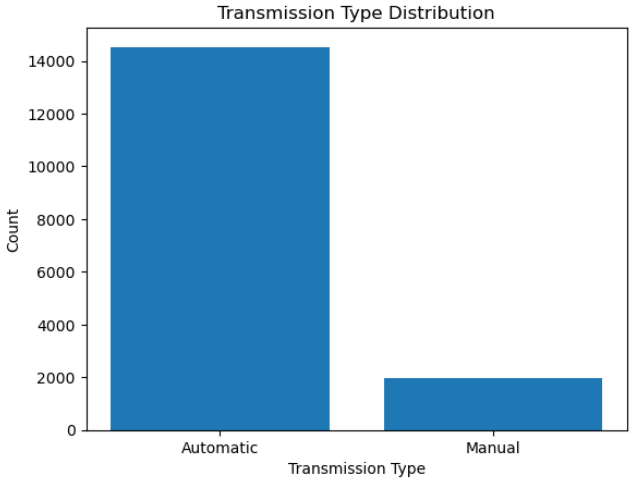

Figure 8: Transmission class balance

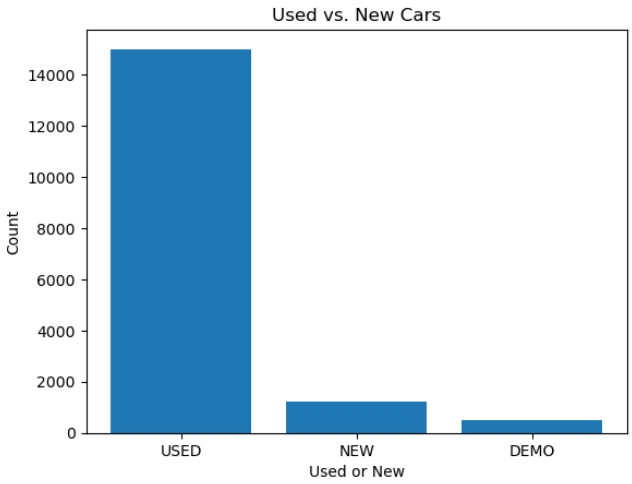

Figure 9: Used/New class balance

- The BodyType was also a subject of discussion. It was eventually concluded that to retain the most information, the categories would be ordered by their relative size to imply some ordinality. Although these body types contain other information (such as Ground Clearance, Number of Doors), this information was either found in other columns or of minimal importance. In the end the categories were sorted in increasing order as follows: Convertible, Coupe, Hatchback, Sedan, Wagon, SUV, Ute Tray, People Mover, Commercial and converted to numbers of the same order.

(b) **Derived Attributes:**

- The Engine column was split up into EngineCyl & EngineVolume. As the data about the number of cylinders was already included the column of CylindersinEngine, the EngineCyl was dropped.
- The ColourExtInt column was turned into simply the ExtCol column as most of the interior columns were missing.
- The Territory was extracted from the Location column to keep the data more anonymized while still retaining useful information
- The DriveType was split up into 2 columns: DriveFront & DriveRear, 2 boolean columns indicating if the vehicle was powered on that axle set.

(c) **External Data Sources:** There are numerous possibilities for expanding the business potential with more data.

- Supply chain information can show the trends of the new car market, which the used car market mimics to some extend. Disruptions in the supply chain of new vehicles can cause people to seek out used vehicles as alternatives, shifting demand. This information shows the relative abundance of vehicles with their pricing, allowing to make more informed decisions.
- Technological trends are playing an increasingly important role in the Auto industry. Datasets including more technical features of cars such as entertainment systems impact the price of modern cars. Having a better understanding of the technical offerings of cars can lead to more accurate pricing & exploitation of market trends in this respect.
- Economic indicators like unemployment rates, consumer confidence indices, & historical demand could allow the company to better forecast the potential selling price of a vehicle. For example, an increasing job market in an area would also potentially increase vehicle demand & thus their prices.
-

(d) **Pre-Processing Steps:** Another issue encountered were several categorical variables that could not be encoded with some ordinality. These columns (Brand, ExtCol, State, FuelType) were 1-hot encoded, which means that each category is split into a separate boolean columns indicating whether the observation had this feature or not. Regarding outliers, in most of the numeric cases, such as door, seats, cylinders or fuel consumption the occurrence of outlier is not possible or significant. On the other hand features, such as year, kilometers and price there might be some outliers. However based on the correlation matrix one can see, that the features kilometer and year are strongly correlation, so the decision was made, that we would like to ignore the eventual outliers with the assumption, that they follow the general patterns, thus are not heavily impacting the pattern.

4 MODELING

(a) **Data Mining Algorithms:** Since our problem involves regression (predicting the numerical price of a vehicle), several variants have been identified.

- **Linear Regression** is the simplest case, when there is a linear relationship between each predictor & the response. This is the easiest model to implement &

explain, however due to this it can fail to capture more complex relationships. It is also susceptible to highly correlated variables.

- **Ridge & Lasso Regression** perform some level of shrinkage for less important variables, which is useful againts overfitting. However they are still sensitive to outliers, & may not perform proper feature selection.
- **Principal Component Regression** involves reducing the dimensionality of the feature space, being especially useful for when there are many predictors. This algorithm loses interpretability.
- **Partial Least Square Regression** uses a weighted version of the predictors based on the response, using the highest variance & correlation. The reduction yields higher stability than simpler methods. Like principal component regression, the creation of an intermediary covariance matrix renders it uninterpretable.

Considering the question of robustness, one can tell, that based on the figure 3, there might be some outliers, however the dataset in general is not heavily impacted by outliers. First of all we considered the ElasticNet algorithm. In case of ElasticNet we used once the RandomizedSearch and the GridSearch. Both the final r2 scores of the two tuning and the tuned hyperparameters were quite similar in the end. However as the RandomizedSearch significanty outperforms the GridSearch in computational time, the RandomizedSearch is the preferred method to use. The l1_ratio parameter of ElasticNet tunes the ratio of the L1 and L2 ratio. The tuned parameter was 0.9 for both cases. However, we decided to test also the Lasso and Ridge regression separately. The results of the comparison of the different settings are shown in the figure 10. Our experiments showed, that the optimal linear model is the Ridge regression. Hence, the only parameter to tuned is the α value for the Ridge regression.

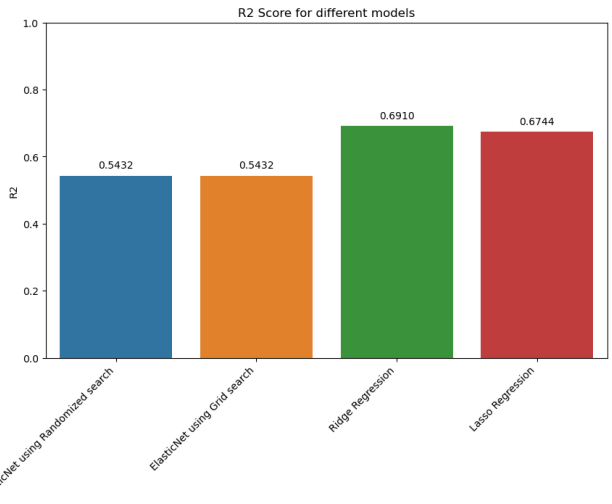

Figure 10: Model comparison

Model	R2 score	α	l_1
Ridge regression	0.6910	0.9	-
Lasso regression	0.674	17	-
ElasticNet (Grid)	0.5432	0.1	0.9
ElasticNet (Randomized)	0.5432	0.1	0.9

Table 1: Fine tuned parameters

The Ridge regression uses a square penalty shrinkage method. This is because this algorithm is useful when dealing with multicolinearity (correlation between predictors), as vehicles of a certain type share many similar features. Additionally, Ridge regression reduces the likelihood of a predictor shrinking to zero, which is potentially helpful in saving all of the information in the dataset.

(b) **Hyperparameters:** Ridge regression only contains 1 hyperparameter; α, controlling the strength of regularization. A higher α pushes the coefficients closer to zero, introducing bias to prevent overfitting. An alpha of zero results in standard linear regression. The selection of this parameter is most often done via Grid search cross-validation, running through a range of values, & choosing the value with the lowest validation error. However, based on the results for

(c) **Performance metric:** As describing metric, the R^2 regression score is chosen. R^2 score is a measure of how well the independent variables explain the variability of the dependent variable. A disadvantage of R^2 is, that it does not indicate whether the coefficients are statistically significant or not. However we have a mentionable high number of input variables, due to the encoding of categorical columns. Hence, it is expected and also accepted, that a some variables are not significant for the prediction and it is not intended to penalize the usage of less significant variables in the score metric.

(d) **Training of the model**: The first step of the training process is the splitting of the data. To make sure reputability a random seed was set, right at the beginning of the process. Then as first step the data was split into train (75%), validation (12,5%) and test (12,5%) set. This was done already before of the comparison of different models (see figure 10). Based on the results, the decision was made, that the chosen model is Ridge regression and the parameter to tune is α. To get our best model, we used Grid search combined with cross validation on the wide range of values for alphas: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 7, 12, 17, 22, 27, 32, 37, 42, 47]. As the grid show, we set a higher granularity for smaller values. After fine tuning the model we ended up with the tuned parameters shown in table 1:
As one can see the resulted best parameter for the chosen Ridge regression is $\alpha = 0.9$.

5 EVALUATION

Based on the result table (table 1, in the evaluation part we pick the Ridge regression with $\alpha = 0.9$. The seed is set again for reproducibility and the train test split is repeated. This time as the assignment

states, the model is trained on the train and validation set and the results are computed based on the test set. The parameter α is set to the previously founed tune value of 0.9. To get a better idea of the accuracy of our tuned model, it was compared to two different models.

Our dataset was chosen from the Kaggle platform and multiple users were modeling the same dataset. Therefore our model can be compared to those found on the Kaggle page. Looking at the previous work of the users, we found the work of Sung Keum (https://www.kaggle.com/code/sungkeum/eda-and-lr-random-forest-accuracy-86) to valide our results to. Thus, two models were picked beside the chosen Ridge regression. One of them, a similarly performing model was the linear regression model using default parameters, this model is taken as the baseline. The better performing model, the Random Forest regression model was chosen as a benchmark model based on the Kaggle reference code. In the reference code, the result for the linear model was 0.7153 and for the random forest 0.8612 R2 score.

Our results of our evaluation of the models are shown in the figure 11. In the reference code, the used underlying data is slightly different. Features such as Brand, Year, Model, City and Location features are removed. However the State feature is kept, just like in our approach. The further difference between the two approaches can be seen in the outlier handling. In our case, we argued, that the number of outliers is not crucial and fits the general pattern, but in the case of the reference code, roughly 1100 entities were filtered out based on the outlier criteria implemented both on the kilometer and price features. Based on the results it can be said

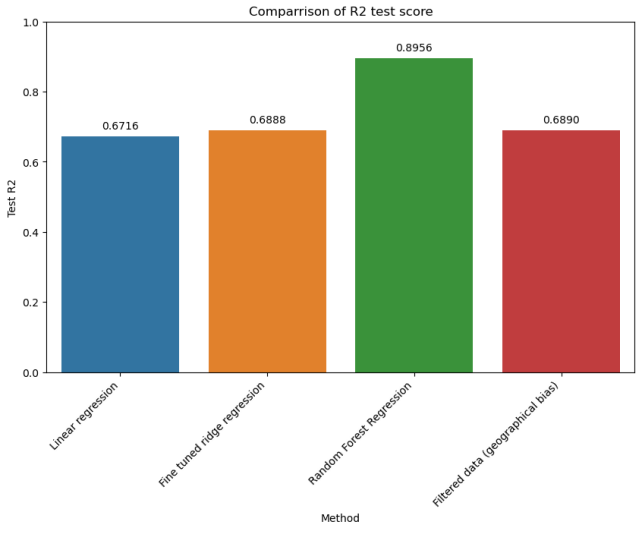

Figure 11: Model comparison

that our model achieved the base line expectations, as it performed slightly better than the linear regression model of the reference code. However, the benchmark model Random Forest significantly outperformed the Ridge regression regarding the R2 score nearly by 20 percent.

Our assumption is that it can be due to the question of robustness and outliers. We believe that not implementing outlier filtering and choosing Ridge regression at the same time was a controversial decision. It might have been resulted in higher scores if we had either filtered out the outliers, as it was done in the reference code, or alternatively, had picked a more robust method for the regression task.

Additionally it should be emphasized that the results for the Random Forest are slightly higher and this might be due to the fact that for the features such as NewOrUsed and Body Type, we didn't use OneHot encoding as we wanted to keep the hierarchy of the labels, thus we used a mapping list, where as in the reference code these features were also OneHot encoded.

Moreover, I would like to note of the computational costs of the models. First of all in the given use case, this might not be a crucial aspect, given the size of the datasets, but in total there are significant differences in training time. The Ridge regression outperforms both alternatives, but the difference is specially significant compared to the Random Forest regression. Thus, for a larger amount of data, the question of training time needs to be considered for the Random Forest Regression. Only in our small scale use-case this is still a negligible issue.

Regarding the data mining success criteria, one can say, say based on the R2 scores, that the Random Forest regression is favored above the Ridge regression. To better understand the model and to answer the question of success criteria, one can take a look at the figures 12, 13 and 14. As one can see, on one hand the Random Forest regression delivers good result, the predicted prices match the test values. On the other hand the linear regression's and the Ridge regression's match is poorer, but the main problem with both models is that the predicted prices are in some cases negative. Hence, it can't be said that those models fulfil the **business requirements**.

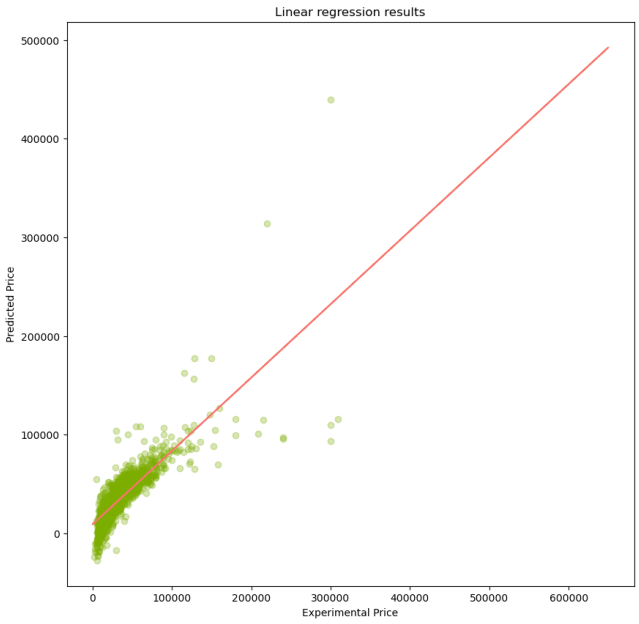

Figure 12: Liner regression results

Finally, I would like to also address the **question of protected variables**. In the dataset the sensitive data might be the detailed

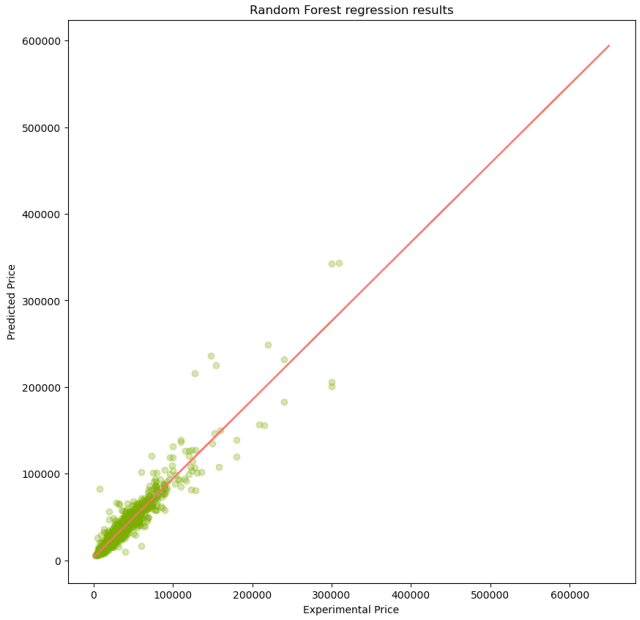

Figure 13: Random Forest regression results

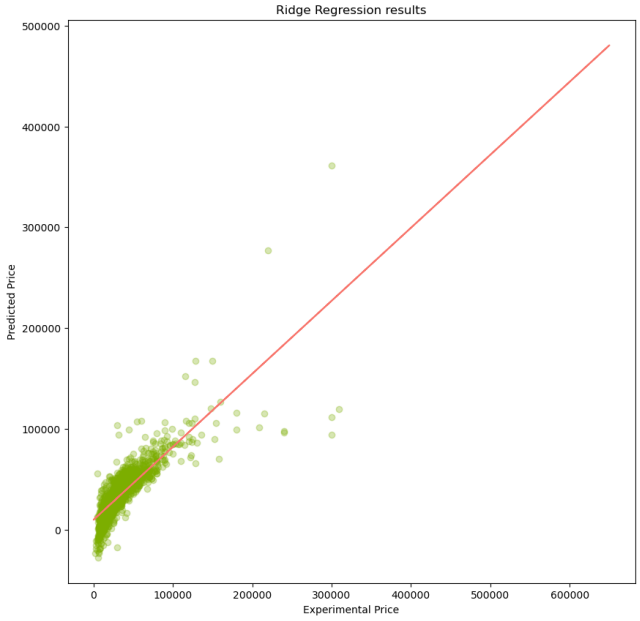

Figure 14: Ridge regression results

geographical information, mentioning not just the state but also the name of the town. This information was partly modified and the granularity of the geographical information was change from town level to state level. The reason for the elimination of city names was, that this information might negatively impact poorer regions, towns compared to rich towns. This is a form of spatial "bucketing". Our assumption was that leaving poorer, less developed town names in the geographical information might have ensured a more accurate regression but on the cost of social and geographical bias. However, as we wanted to eliminate (or at least dampen) any kind of geographical and social bias, we decided to ignore the town names and just to include the more broader geographical information, the state names. As states are more homogeneous and we don't expect to include significant bias that is negatively impacting the people. To evaluate our assumption we tested our model for the case, when we also exclude the state names and completely ignore any geographical information. This approach is labeled as 'Filtered data (geographical bias)' in figure 11. As one can see the overall accuracy is not impacted significantly by the presence or absence of the geographically sensitive columns. Thus, we can conclude as expected the state information does not introduce any bias into the results. However, based on this result it could also be argued that the state information could be completely eliminated from the data, as the presence or absence of it does not influence the results.

The 1st business objective of the company is to get an idea about the current pricing practice in the Australian car market. This will help companies entering the market to have a better understanding of how to price their cars based on existing ones.

The 2nd aim is to provide a model that predicts the probable price of a car based on its characteristics and features. This is helpful for a company already in Australia that wants to get an idea of how to price a new vehicle before launching.

The 3rd goal is to make the company more resistant against the possible price fluctuations of different employees, as usually the pricing is determined by employees with experience in the field. This model can ultimately do similar predictions for the company without the need of an experienced employee.

6 DEPLOYMENT

(a) The **business objectives** of the company were threefold. 1st, to get an idea about the pricing practices in a new market. This is done by analyzing the data and all three models deliver acceptable results. The 2nd and 3rd objective regarding a prediction model both require a good quality prediction, that results both an advantage against the competitor and a robustness against the fluctuation of workforce. The originally chosen model, the Ridge regression, unfortunately does not seem to work as intended. The main reason is, that some predictions turn out to be negative prices. Therefore, the Ridge and Linear regression models are not sufficient enough to fulfil the 2nd and 3rd business objectives. However, the benchmark model, the Random Forest regression model, seems to work well for this purposes. However, one needs to be aware of the fact, that for the higher the price of a car gets, the bigger the variance is, so it is important to have an expert to validate the price predictions for expensive cars. Therefore a recommendation would be a hybrid deployment. More precisely, the model can deliver good results, but mainly for average priced cars. An expert's opinion is still required for more expensive cars, which have been the outliers in the original dataset price-wise. Therefore we recommend to introduce an art

of threshold or a **trigger** in the documentation, that for special, luxurious and/or sport cars, an expert is asked to double-check the prediction results of the model.

(b) Finally, commenting again on the **ethical aspects of the prediction**. To not differentiate poor and rich neighborhoods and hence, not to include social bias in the prediction, the town names were removed from the base data and just the more homogeneous and broad information, the state names are included. However, as one can see, the present or the absence of the state names are not significantly impacting the final R2 scores. Thus, one might also remove this information. On the contrary, even the town names can be included and that is expected to improve the prediction. As this would result in a much bigger base data table, that might effect the computational time and also introduces some geographical and social bias into the model.

(c) An additional remark goes to the **reproducibility**: The used packages during of the project are freely accessible, standard packages, such as sklearn, numpy, pandas and matplotlib and seaborn for visualizaton. In the original case, the setting of the seed is crucial only for the train-test split. However based on results we recommend to use the Random Forest model, which is based on a random initialization when fitting the mode. Therefore, for reproducibility and comparison a seed should be also set for that model. In our case we defined a seed at the top of the code and used the same seed for both the splitting and the model initialization.

7 SUMMARY

Choosing a regression task for an interesting dataset for predicting car prices provided us valuable insight into possible applications of machine learning in real world business classes. However the first issue was to convert class information into numeric data. In some cases, it was decided to keep a hierarchical structure and used mapping to encode the classes of a feature. This might be also the reason why we ended up with higher values for the Random Forest model, although the difference might be just due to the random initialization of the model.

The mistake or issue we made during our project was the addressing of the issue of outliers. We didn't think that the outliers are significant and thought, that outliers in features such as kilometer or year should not be crucial as they correlate strongly and we believed these outliers fall in the same pattern as the others. However, this might be the reason, why we ended up having some negative predicted prices, as high kilometer numbers should should logically reduce the price, and extremely high kilometers might cause eventually negative pricing. Additionally, the target feature 'price' also includes some outliers, that may impact the model and these outliers should be truly directly considered, as they are special cars, thus not fitting the general schema. To name as example, the most expensive car in the dataset is a Ferrari from year 1959 with 9902 kilometers and a price of 1.5 Million dollars. This car is clearly not fitting the general pattern and the chosen model, the Ridge regression also not robust enough to handle these entities. Therefore either the outliers regarding the target variable must have been filtered out or a more robust model should have been chosen from the get go. This is also proved with an alternative model, with the Random Forest approach, which is based on the same dataset containing outliers, but can deliver high quality results.