A Deep Multi-Task Approach for Residual Value Forecasting

Ahmed Rashed(✉)[1], Shayan Jawed[1], Jens Rehberg[2], Josif Grabocka[1], Lars Schmidt-Thieme[1], and Andre Hintsches[2]

[1] Information Systems and Machine Learning Lab, University of Hildesheim, Germany
{ahmedrashed,shayan,josif,schmidt-thieme}@ismll.uni-hildesheim.de
[2] Volkswagen Financial Services AG, Braunschweig, Germany
{Jens.Rehberg,Andre.Hintsches}@vwfs.com

Abstract. Residual value forecasting plays an important role in many areas, e.g., for vehicles to price leasing contracts. High forecasting accuracy is crucial as any overestimation will lead to lost sales due to customer dissatisfaction, while underestimation will lead to a direct loss in revenue when reselling the car at the end of the leasing contract. Current forecasting models mainly rely on the trend analysis of historical sales records. However, these models require extensive manual steps to filter and preprocess those records which in term limits the frequency at which these models can be updated with new data. To automate, improve and speed up residual value forecasting we propose a multi-task model that utilizes besides the residual value itself as the main target, the actual mileage that has been driven as a co-target. While combining off-the-shelf regression models with careful feature engineering yields already useful models, we show that for residual values further quantities such as the actual driven mileage contains further useful information. As this information is not available when contracts are made and the forecast is due, we model the problem as a multi-task model that uses actual mileage as a training co-target. Experiments on three Volkswagen car models show that the proposed model significantly outperforms the straight-forward modeling as a single-target regression problem.

Keywords: Multi-Task Learning · Residual Value Forecasting · Pricing · Automotive Industry.

1 Introduction

Forecasting enables key-decision making in many business applications, including but not limited to fields of lease, loans and insurance. A leasing system in place, reduces the initial up-front costs of goods for the clients significantly, for example, automobile leasing. Globally, there has been a surge in the demand of leased vehicles, and Germany is leading the market with a very high penetration of operating leases within Europe [13]. The leasing contracts are designed with respect to the residual value of the vehicle at a point in the future. This stands to reason as an overshoot on the estimate would suggest a lower leasing rate leading to diminishing profits if the vehicle is sold at a lower price after the expiration of the leasing contract. Vice versa in the case of undershooting the residual value. Consequently, organizations require a dependable method to forecast residual values as accurately as possible to manage the risk inherent to their business.

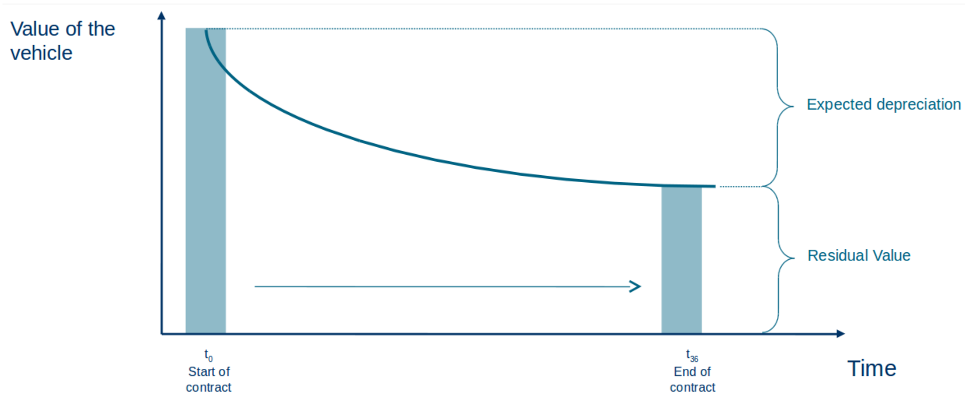


Fig. 1: A sample timeline of depreciation and final residual value of a vehicle.

A residual value forecasting method takes into account various factors such as the initial mileage set for the vehicle, engine configuration, model launch date, list price, etc. On the other hand, there could be factors that are more dynamic in nature as opposed to former which stay static over the leasing period. Actual mileage a vehicle is driven, damages incurred throughout the leasing period are factors that are both unknown at the start of a contract and dynamic. However, it is reasonable to assume that both play an important role in the future residual value of the vehicle. This paper formulates the vehicle residual value forecasting problem as a multi-task learning model by defining other than the primary task of residual value, additional auxiliary tasks of predicting various quantities that are prone to change in the duration of the lease.

Multi-task learning [2] is a mainstay of numerous machine learning applications. The intuition behind learning multiple tasks jointly is to learn a richer shared representation from multiple supervisory tasks than possible from learning a single task. A rich representation as such can then effectively generalize to out of sample test data better. Natural Language Processing [12], Computer Vision [6] and Time Series forecasting [2] domains have benefited from using multi-task learning strategies.

It is worth noting that although forecasting the residual value is of primary importance, we can explore additional auxiliary tasks to model the problem as a multi-task learning problem. These auxiliary tasks solely exist to provide additional supervision. Specifically, we explore four such tasks, the actual mileage the vehicle gets driven, as the initial mileage limit at the time of contract is not set in stone, the cost of damages it can incur, expected days it takes for the vehicle leasing company to resale the vehicle and lastly, the expected date of returning the vehicle as opposed to the initial set date at the time of contract.

This paper foremost tackles the research problem of designing an optimal machine learning model that caters for multiple tasks. Secondly, an exhaustive search procedure is applied to determine the feasibility of incorporating one auxiliary task over another, or possibly multiple together. Third, we validate the results of multi-task learning by comparing to single task learning objective of residual values. Our focus in this paper is on deep neural network architectures to model the task at hand, given that the shared representation between the tasks could be represented by a hidden layer as opposed to more classic machine learning models such as Gradient Boosted Decision Trees [3] and Support Vector Machines [5] where catering for multiple tasks requires considerable alterations.

The rest of the paper is organized as follows. In Section 2, we summarize the related work. We discuss the problem formulation of the multi-task residual value forecasting in Section 3. In Section 4, we present and discuss the technical details of the proposed model. We present the experimental results in Section 5. Finally, we conclude with discussing possible future work in Section 6.

2 Related Work

This section sketches an overview of prominent methods in automobile residual value forecasting and secondly works that exploit auxiliary tasks in the light of multi-task learning. We note the work from [9], where the authors propose an SVM regression method for automobile residual value forecasting. Model selection for the method was based on an evolution strategy in favor of a grid search procedure. The dataset consisted of more than 100,000 samples for a single vehicle model from an anonymous car manufacturer. Interestingly, the dataset had 176 features without any dimensionality reduction. More closely related to our work is a Neural Network based regression method proposed by [11], which was tested on 5 different however undisclosed vehicle models. Model selection was done via an evolutionary algorithm similar to previously noted work. Lessmann et al. [10] provided a comparative analysis of 19 different regression algorithms applied to a dataset of 450,000 samples of 6 different vehicle models and report that Random Forest [1] based regression achieved the optimal results. Furthermore, the authors noted that some car models were difficult to model than others.

Various approaches have been proposed that fall under the umbrella of multi-task learning. A survey on multi-task learning could be found in [14]. Pioneering work in multi-task learning [2] proposed a neural network regression model that forecast mortality risk. Besides the main target of risk, the authors predicted for various other correlated attributes that lead to a significant gain in accuracy over the main task when compared to single task estimation of the same risk task. Auxiliary tasks have also been explored in but not limited to, Natural language processing [12] and Computer Vision [6]. An unsupervised auxiliary task was defined in [12] that predicted surrounding words together with the main objective of sequence labeling leading to a multi-task learning framework. The authors noted that utilizing the auxiliary task encouraged the framework overall to learn richer features for semantic composition without requiring additional training data and ultimately leading to consistent performance improvements on various sequence modeling benchmarks. Work done by Girshick [6] involved an extensive ablation study that compared the multi-task learning loss of object classification and bounding box regression to single-task object classification. The results established the supremacy of the multi-task learning approach.

In light of the above, we propose a multi-task approach to model the residual value forecasting problem. To the best of our knowledge, our paper is the first to do so. We show that this not only leads to superior performance when compared to a variety of standard baselines but also an equally important result that the proposed method beats the single task learning objective of predicting for the main task.

3 Problem Definition

Residual value forecasting can be formulated as a multi-task regression problem where given some car configuration details $X_c \in \mathbb{R}^m$ and the leasing contract details $X_l \in \mathbb{R}^q$, we need to define a function $\hat{y} : X_c \times X_l \to \mathbb{R}^{|\mathcal{Y}|}$ to predict a set of target values $\mathcal{Y} := \{y_1, y_2, ..., y_{|\mathcal{Y}|}\}$ after the end date of this leasing contract, such as the expected mileage to be driven or the expected damage value. In residual value forecasting, our primary target will be the car's market value (Residual value) y_{mv} after the contract end date while the rest of the targets will act as auxiliary targets $\mathcal{A} := \mathcal{Y} \setminus \{y_{\text{mv}}\}$ that will help in improving the model accuracy and generalization.

4 Proposed Method

Given the car configuration details X_c and the contract details X_l we need to define the multi-target prediction function $\hat{y}$. A good choice for such function will be a multi-layered neural network that has a feature extraction part with hard parameter sharing g and a set of independent prediction heads f_k for every available target k as follows:

$$z = g\left([X_c, X_l]; \theta_g\right) \quad (1)$$

$$\hat{y}_k = f_k\left(z; \theta_{f_k}\right) \quad (2)$$

where z are the extracted latent feature vectors from the contract and car details. g and f are a series of non-linear fully connected layers with network parameters θ_g and θ_f respectively. The full model architecture is shown in Figure 2.

To optimize the proposed model we have to define a separate loss function $\mathcal{L}_{y_k}$ for every prediction head and jointly minimize all of them simultaneously

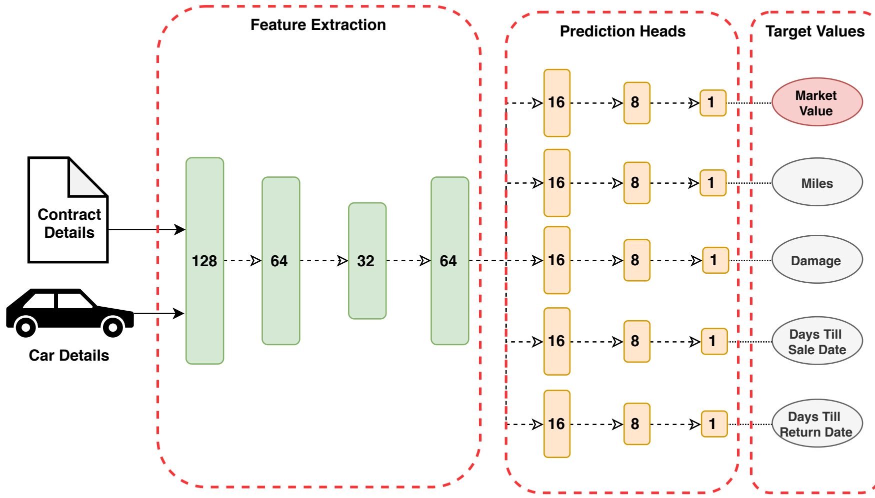


Fig. 2: The proposed multi-layered architecture for residual value forecasting. The first part is a multi-layered neural network for feature extraction with hard parameter sharing and the second part is a set of independent multi-layered prediction heads for every target value to be predicted.

with adding a specific weight for each target loss as follows:

$$\mathcal{L}(\Theta) = \sum_{k=1}^{|\mathcal{Y}|} \alpha_k |y_k - \hat{y}_k| \tag{3}$$

For simplicity, we converted all possible target values to continues values and we used the mean absolute error loss for all targets to avoid any sensitivity with outliers in the training data.

5 Experiments and Results

In this section, multiple experiments were conducted to evaluate and find the best architecture and target values for the proposed model. These experiments aim to answer the following research questions:

RQ1 How many layers of hidden units are needed for the proposed model to learn and predict the target values?

RQ2 What are the best set of co-targets that will improve the residual value forecasting accuracy?

RQ3 How well does the proposed model perform in comparison with the currently employed methodology and other well-known regression models for residual value forecasting?

5.1 Dataset

In the following experiments, we used our VWFS business to business sales propriety dataset which contains around 270k instances of leasing contracts. The dataset was split into three chronological parts with different train to test ratios. Each chronological part is further split into chronologically ordered train and test sets where half of the training set is used for parameter tuning. In the scope of this work, we only focused on evaluating the models on the primary trim line (Anchor model) of the top three Volkswagen popular car models which are Tiguan, Passat, and Golf. To do so, all test sets were filtered to contain only those models. We made sure that train and test parts are disjoint which means that any car instance had to be already sold before the end date of the data part in order to be considered. Detailed statistics of the utilized data parts are shown in Table 1.

Table 1: Datasets Statistics

	Train Period	Test Period	Train #	Test #	Train Ratio
Split 1	2002-2014	2015-2019	77980	8702	90.0%
Split 2	2002-2013	2014-2019	54151	22561	75.0%
Split 3	2002-2012	2013-2019	32136	37876	46.0%

5.2 Data Preprocessing

Our first step of data preprocessing was to define the best car configuration and contract features to be used as input. Regarding the car configuration, we used all available configurations that don't contain any personalized data which are shown in Table 2. These features also contain some expert features that might affect the residual value but not part of the car configuration details. For the contract details, we only used the start date, end date, mileage cap per year and contract term in months.

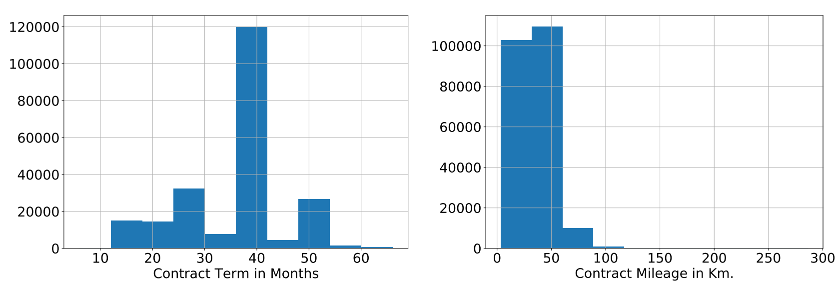
5.3 Exploratory Analysis

Before applying the experiments to answer our research questions, we did some exploratory analysis on the given dataset to have better insights. Firstly we plotted the distributions of all contract types based on their duration term and mileage cap which are shown in Figures 3(a) and 3(b). The distributions show that the majority of contracts have duration terms of 36 and 42 months with a mileage cap ranging from 5 km to 100 km per-year. Secondly, the distribution of the top popular car models and their list prices were analyzed. Figure 4 (a), shows that the majority of the instance are Passat and Golf while Tiguan has a much lower frequency. Figure 4 (b), shows that there is no significant deviation

Table 2: Car Configuration Features

Feature Name	Type	Expert Feature
Brand	Categorical	-
Model Description	Categorical	-
Fuel	Categorical	-
Engine	Categorical	-
Engine Description	Text	-
Registration Date	Date	-
List Price	Float	-
Equipements	List	-
Color	Categorical	-
Model's Market Launch	Date	Yes

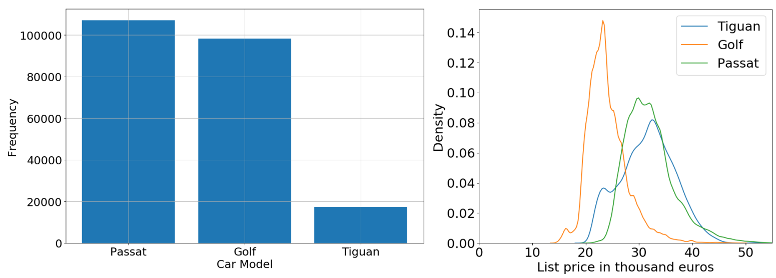
in the list price between the three models, however, Golf cars have the lowest price range because it falls in the compact cars segment.



(a) Contract Term (b) Contract Mileage

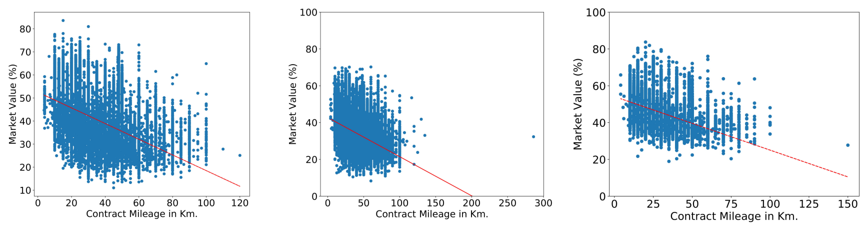
Fig. 3: Distribution of contracts' terms and mileage caps

Finally, we analyzed the correlation between the market value and the contracts' term and mileage caps for each car model to identify any anomalies or outliers. Correlation results in Figures 5 and 6, show that a strong correlation consistently exists between the contract's term, mileage cap, and the future market value. The longer the contract term is, the lower the market value will be. Also, the market value decreases with increasing the contract's mileage cap. The results also show that there is a considerable noise especially in the correlation between the market value and the contract's term, however, a trend is still clearly visible.



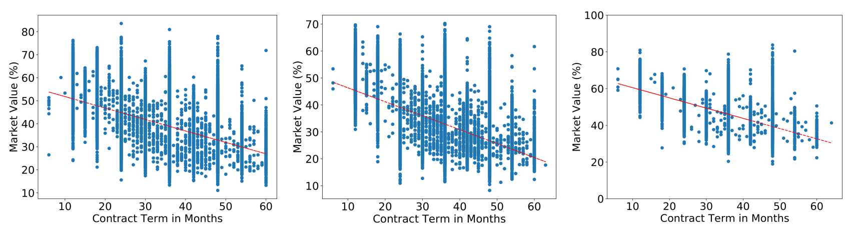
(a) Models (b) List Prices

Fig. 4: Distributions of car models and their list prices



(a) Golf (b) Passat (c) Tiguan

Fig. 5: Correlation between the market value percent and contract's mileage cap



(a) Golf (b) Passat (c) Tiguan

Fig. 6: Correlation between the market value percent and contract's term

5.4 Experimental Protocol

For our experimental protocol, we used the last three years of the training part for hyper-parameters tuning using a grid search. We then fully retrain the model again on the complete train part and evaluate on the test part using the mean absolute error metric (MAE). Regarding the grid search, we tested the learn rates of [0.01, 0.001, 0.0001], the batch sizes of [128, 256, 512, 1024, 2048, 2560], the loss weights of [0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00005, 0.00001] and number of epochs ranging from 50 to 650. The best found hyper-parameters were batch size = 2048, Learn rate = 0.001, optimizer =ADAM, $\alpha_{\text{mv}} = 1.0, \alpha_{\text{miles}} = 0.001, \alpha_{\text{damage}} = 0.005, \alpha_{\text{saledate}} = 0.01$ and $\alpha_{\text{returndate}} = 0.01$ for all data splits. Best number of epochs were 350 for the first two data splits and 450 for the third.

5.5 Model Complexity (RQ1)

To tune the model architecture, we applied several experiments on the first data split using different candidate architectures. We started first by increasing the number of shared layers and their width while fixing the prediction heads to one fully connected layer with 16 units. The best candidate shared architecture is then carried over to test a different number of layers for the prediction heads. We used Leaky ReLU activation functions in all of our experiments.

The comparison results in Table 3, shows that the error rate decreases by increasing the number of shared layers until it reaches four layers. Further fine tuning of the prediction heads has a lesser effect on the error rate which decreased from 3.73% to 3.70%.

Table 3: Performance Comparison of Different Candidate Architectures

Shared Layers	Prediction Layers	Mean Absolute Error(%)
128	16	5.22
64	16	5.32
32	16	5.79
128x64	16	6.29
128x32	16	6.21
128x16	16	7.95
128x64x32	16	4.0
128x64x64	16	3.78
128x64x32x16	16	3.74
128x64x32x32	16	3.74
128x64x32x64	16	**3.73**
128x64x32x64	8	3.71
128x64x32x64	32	3.77
128x64x32x64	32x16	3.83
128x64x32x64	16x8	**3.70**
128x64x32x64	8x4	4.01

5.6 Comparison Between Possible Co-Targets Values (RQ2)

In this section, an experiment was applied on the first and second data splits to measure the effect of every co-target value. The co-targets are features that are known for the past contracts, however, they are unknown for future contracts. To illustrate the point, the miles driven (or damage values, etc.) can be known only when a contract ends, while we want to estimate the residual value at the moment a contract being signed. For this reason, those features cannot be utilized as an input in estimating the residual values. As such, we innovate on treating those features which are available only for the training set (i.e. past contracts) as co-target variables.

After filtering all predictor features we managed to identify four possible co-targets to be used for forecasting the market value which are shown in Table 4.

Table 4: Co-Targets Values

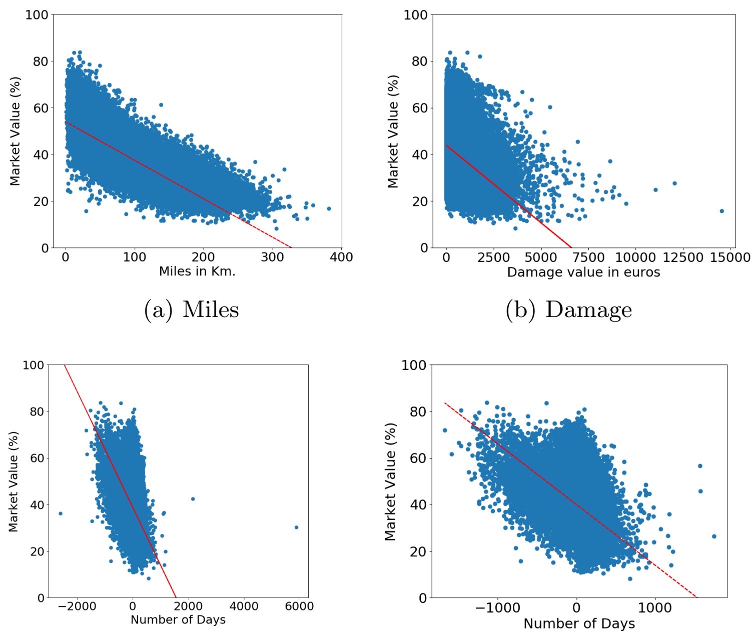
Target Name	Description	Type
Miles	The expected actual miles to be driven by the client	Float
Damage Value	Expected total damage value	Float
Sale Date	Expected sale date after returning the car	Date
Return Date	Expected car return date after the contract ends	Date

In order to change all targets into continuous values, we had to change the sale and return dates into a numerical number by using the difference in days between the contract end date and those target dates. A negative number of days for such targets will indicate that the car was returned and sold before the original contract end date.

To have first insights about the candidate co-targets, we plotted their correlation graphs with respect to the market value. Correlation results in Figure 7, show that there is a strong correlation between the total driven mileage and the market value which means it can be a very good candidate co-target. It also shows that the correlation of all other targets are very noisy however there exist a slight trend that indicates they might be useful as co-targets.

Assigning the correct loss weight α_k to each co-target is a crucial step in fine tuning any multi-task learning models [8, 7]. To do so, we conducted a sensitivity analysis using the first data split on the loss weights of each available co-target individually. The best-found weights are then used further in combining multiple co-targets at the same time for the sake of reducing the grid search space over all possible combinations. The sensitivity analysis results in Figure 8, show that the model is most sensitive to the Miles as a co-target while changing other co-targets weights show no significant effect on the model performance.

Table 5 shows the results for comparing the different co-targets and their effect on the residual value forecasting using the best-found loss weights. Results show that the highest improvement is achieved by adding the expected miles to be driven while the rest of the targets have a negligible effect. Adding the damage along with the miles also has a very small improvement over miles alone, if neglected, we can safely assume the expected miles alone has the most significant



(a) Miles (b) Damage

(c) Number of days till return date (d) Number of days till sale date

Fig. 7: Correlation between the market value percent and other co-targets

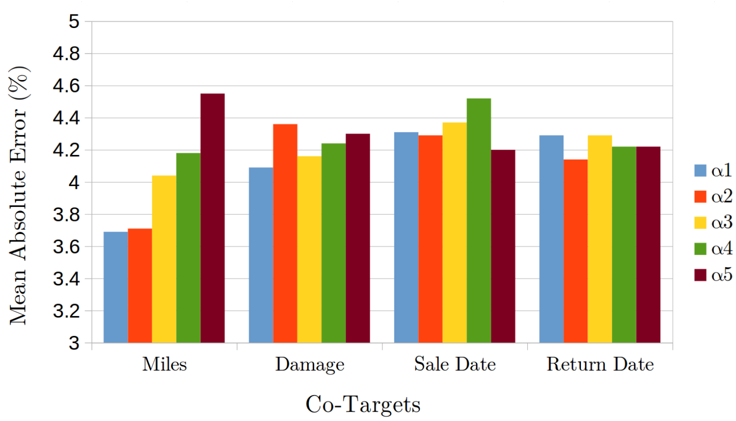


Fig. 8: Comparison between different loss weights for every available co-target. We used α=[0.001, 0.0005, 0.0001, 0.00005, 0.00001] for the miles; α=[0.01, 0.005, 0.001, 0.0005, 0.0001] for the damage; and α=[0.1, 0.05, 0.01, 0.005, 0.001] for the return and sale dates.

Table 5: Co-Targets Performance Comparison

Model	Mean Absolute Error In Market Value(%)	
	(2014-2019)	(2015-2019)
Single-Task Learning		
STL NN	5.57	4.27
Multi-Task Learning		
MTL NN (Miles)	**3.69**	**3.71**
MTL NN (Damage)	5.64	4.53
MTL NN (Sale Date)	5.41	4.37
MTL NN (Return Date)	5.49	4.29
MTL NN (Miles + Damage)	**3.68**	**3.70**
MTL NN (All)	3.70	3.74

effect which is also in line with the results shown in Figure 8. This can be contributed to the fact that the used damage value is just the total sum and there is no distinction between different damage types and their effect on the market value.

5.7 Comparison with Current Methodology and Baselines Models (RQ3)

In this section we applied multiple experiments on all data splits to compare the performance of the proposed model against the current residual value forecasting manual method and other well-known regression models shown below.

Baselines

1. Random Forest Regressor [1]: A well-known ensemble model for regression. We used a grid search to find the best hyperparameters which are number of estimators = 100 and max tree depth = 3.
2. XGBoost [4]: A well-know gradient boosted tree model for classification and regression. A grid search was used to find the best hyperparameters which are number of estimators = 100, max tree depth = 3 and learn rate = 0.09.
3. STL NN: The single task version of our model
4. Current residual value forecasting method: This baseline is an in-house algorithm designed by the internal residual value management team. It relies mainly on the trend analysis of previous historical sales record after filtering any outliers, adding external factors and market indicators. Figure 9, shows the current workflow of the manual method.

We also compare the results against the Bayes error as we have identical instances that have different market values in the end. This means that the same vehicle type, with the same configuration (including same color, etc.) was sold for different prices at the very same day. Such a variance is dependent on car dealers' selling policies and can not be estimated based on the features we have. This error indicate the best possible accuracy that can be achieved on the given dataset and it was calculated using the average group value as prediction for all identical groups in the test set.

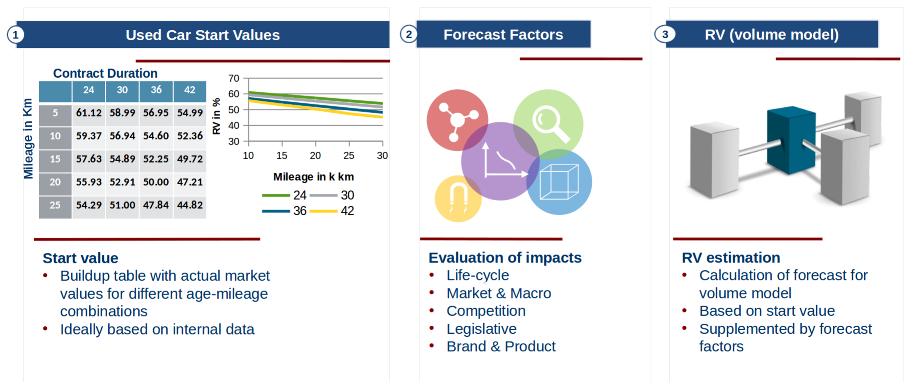
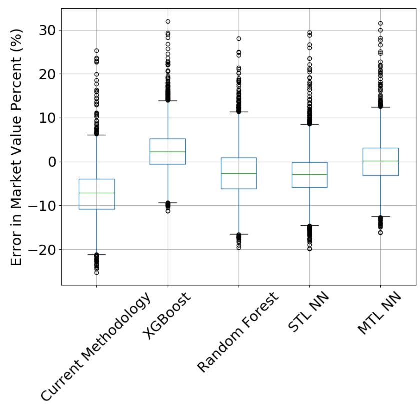


Fig. 9: Workflow of the current manual residual forecasting method. In the first step, regression analysis is done on historical sales record to draw the start value curves for every car model. Secondly, external factors are added to the value curves for adjustment. Finally the residual value is measured by using the adjusted start value curves.

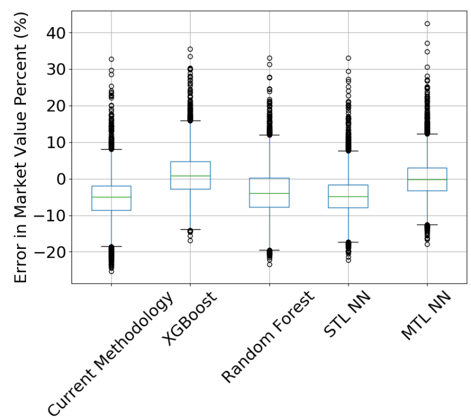
Table 6: Comparison between the multi-task model against other baseline method in terms of mean absolute error percent in market value

Model	Mean Absolute Error In Market Value(%)		
	(2013-2019)	(2014-2019)	(2015-2019)
Current Method	5.28	6.02	7.78
Random Forest	5.26	5.59	4.71
XGBoost	5.26	4.45	3.99
Single-Task Learning			
STL NN	4.24	5.57	4.27
Multi-Task Learning			
MTL NN (Miles)	**3.75**	**3.69**	**3.71**
Bayes Error			
-	2.12	2.03	2.01

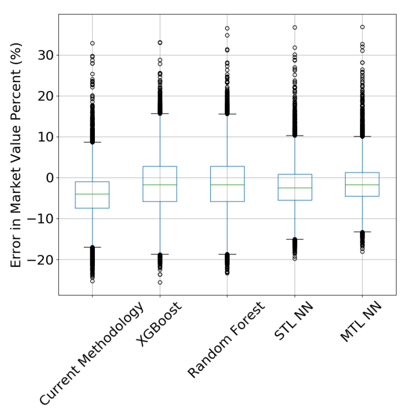
Results Table 6 shows the comparison results between the multi-task model versus all other models. The results show that the multi-task approach provides a significant decrease in error compared with the single task version and the improvement over the current forecasting method is ranging from 25% to 50% in terms of error reduction. The results also show that once an adequate number of training instances are available, all baseline machine learning models can provide a competitive accuracy compared with the current manual forecasting method.



(a) Data Split 1



(b) Data Split 2



(c) Data Split 3

Fig. 10: Distribution of the signed prediction errors

It is worth noting that the current manual forecasting method takes around 45 man-days to complete the full process by highly skilled domain experts, while the machine learning models need a couple of minutes to be trained, which is a significant reduction in execution time and effort.

Error plots in figure 10, show that the signed prediction errors of the multi-task approach are mostly centered around the zero value with a small standard deviation especially in the first two data splits. All other models have larger deviation and they are shifted further away from the zero value. In the third data split, errors of the multi-task approach are slightly shifted to the negative part, however, the deviation is still small compared to other models. This can be contributed to the fact that the third split has the smaller number of training samples, hence a lower accuracy for all machine learning models.

6 Conclusion and Future Work

In this paper, we proposed a multi-task approach for residual value forecasting that utilizes the expected mileage to be driven as a co-target. The proposed model was then compared against the current manual forecasting method and against well-known off-the-shelf regression models with carefully engineered features. Experimental results on the top three popular Volkswagen car models showed that the multi-task approach significantly outperformed the off-the-shelf models and the current methodology in terms of accuracy and with a significant reduction in execution time compared to the manual method. Results also showed that with the right set of features and enough training instances, the off-the-shelf regression model can provide a competitive accuracy to the current manual methodologies.

In future works, we plan to deploy the model in production to help the residual value management team in decision making and reducing the manual effort. We also plan to apply the model on car models with lesser volume and including extra features that might improve the prediction accuracy such as fuel price indicators and the Ifo business climate index.

Acknowledgments

This work is co-funded by the industry project "Data-driven Mobility Services" of ISMLL and Volkswagen Financial Services.(`https://www.ismll.uni-hildesheim.de/projekte/dna_en.html`)

References

1. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
2. Caruana, R.: Multitask learning. Machine learning **28**(1), 41–75 (1997)
3. Chapelle, O., Shivaswamy, P., Vadrevu, S., Weinberger, K., Zhang, Y., Tseng, B.: Boosted multi-task learning. Machine learning **85**(1-2), 149–173 (2011)

4. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794. ACM (2016)
5. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. Journal of Machine Learning Research **6**(Apr), 615–637 (2005)
6. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
7. Harutyunyan, H., Khachatrian, H., Kale, D.C., Galstyan, A.: Multitask learning and benchmarking with clinical time series data. arXiv preprint arXiv:1703.07771 (2017)
8. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7482–7491 (2018)
9. Lessmann, S., Listiani, M., Voß, S.: Decision support in car leasing: a forecasting model for residual value estimation. In: ICIS. p. 17 (2010)
10. Lessmann, S., Voß, S.: Car resale price forecasting: The impact of regression method, private information, and heterogeneity on forecast accuracy. International Journal of Forecasting **33**(4), 864–877 (2017)
11. Lian, C., Zhao, D., Cheng, J.: A fuzzy logic based evolutionary neural network for automotive residual value forecast. In: International Conference on Information Technology: Research and Education, 2003. Proceedings. ITRE2003. pp. 545–548. IEEE (2003)
12. Rei, M.: Semi-supervised multitask learning for sequence labeling. arXiv preprint arXiv:1704.07156 (2017)
13. Roland: Embracing the Caras-a-Service model – The European leasing and fleet management market (Jan 2018), `https://www.rolandberger.com/publications/publication_pdf/roland_berger_car_as_a_service_final.pdf`
14. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)