Used Car Price Prediction: Various Regression Techniques

Zhiyu Lei
University of Rochester
zlei6@u.rochester.edu

Yuting Bu
University of Rochester
ybu3@u.rochester.edu

Abstract

This report provides the detailed overview of used car price prediction using data mining algorithms and methods. We explored various regression models including simple multiple linear regression, LASSO linear regression, random forest regression and multilayer perceptron, among which a random forest model provides the highest predicting power with $R^2 = 0.8764$ on the test dataset. This paper is conducted as the final project report of CSC 240 at University of Rochester in Spring 2021. The notebook file with all the code to run the models and generate the figures and outputs is uploaded public on Kaggle via link to notebook.

1 Introduction

With the development of modern transportation infrastructure and the increasing needs for commuting and travelling, personal vehicles have become more and more popular among people. In the meantime, the used car market has also grown rapidly to fulfill the demands from the low- and middle-income groups. Yet, to set the prices of used cars is never an easy job, as a lot of conditions of a used car can significantly affect its price, including but not limited to its basic performance, how long it has been used, how far it has been driven, etc. Thus, we are interested in finding the relationship between the prices and those potential conditions, and we want to apply some data mining and machine learning techniques to develop a model to predict the prices, which can provide references for used car sellers and buyers.

We analyzed the "CARS DATASET (Audi, BMW, Ford, Hyundai, Skoda, VW)" and predicted the prices of used cars. The dataset is available on Kaggle via link to data. It contains the used car information of 7 brands including Audi, BMW, Skoda, Ford, Volkswagen, Toyota and Hyundai in UK. This data set has 72435 rows, each representing one used car, and 10 columns with the information of brands, models, years, prices, transmissions, mileages, fuel types, road taxes, miles per gallon (mpg), and engine sizes.

The approaches used were as follows:

1. Exploratory Data Analysis: got to know the distribution of each attribute as well as the trend between price and other attributes with the help of data visualization.
2. Data Preprocessing and Feature Engineering: transformed skewed numerical attributes and converted categorical data into dummies.
3. Modeling and Analysis: applied various regression techniques to further analyze the data and predict the prices based on the attributes.

Details and results of each steps are given in this report.

2 Exploratory Data Analysis

We first performed some preliminary exploratory data analysis. This was helpful for understanding the data more thoroughly and ready for more advanced analyses in the next steps. We used Pandas and Numpy [1] libraries in Python.

2.1 Data Type

Among the nine explanatory variables in the data set, five of them (`year`, `mileage`, `tax`, `mpg`, `enginSize`) are numerical and four of them (`Make`, `model`, `transmission`, `fuelType`) are categorical. We would convert categorical data into numerical data during the preprocessing step. Also, none of the attributes contain missing values.

2.2 Distribution of Numerical Features

Figures 1 and 2 show the distributions of numerical features as well as the relationships between price and other features in our data set.

From the figures, we can conclude that used car prices are higher for newer cars; used cars with higher mileages are cheaper; there is very little linear relationship between `price` and `tax`; cars with lower `mpg` tend to be more expensive; and there is quite a strong positive relationship between `price` and `engineSize`. To further look at the information of `mpg`, we found something weird: a few cars have `mpg` greater than 400 and seem to be outliers, but these cars are in fact electric or hybrid cars, and fuel economy makes no sense for these kinds of cars, so we would exclude this column from our analysis in the later steps.

The correlations between each pair of numerical attributes are shown on the heat map in Figure 3. The attribute affecting price the most is `engineSize`, which has a correlation of 0.63.

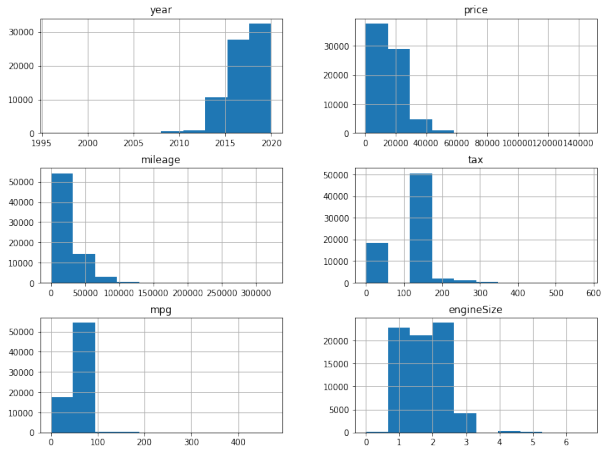

Figure 1: Histograms of numerical features

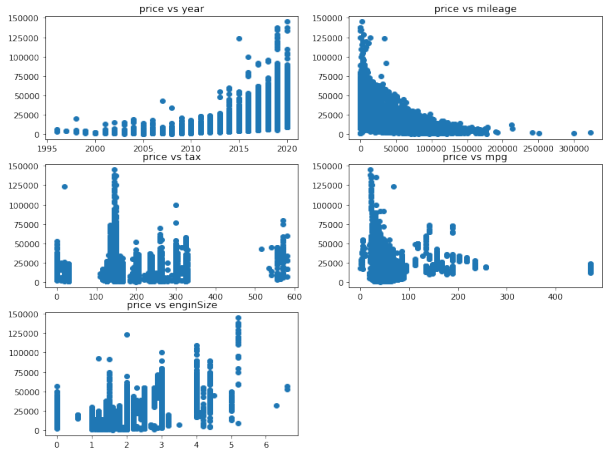

Figure 2: Scatter plots of price against other features

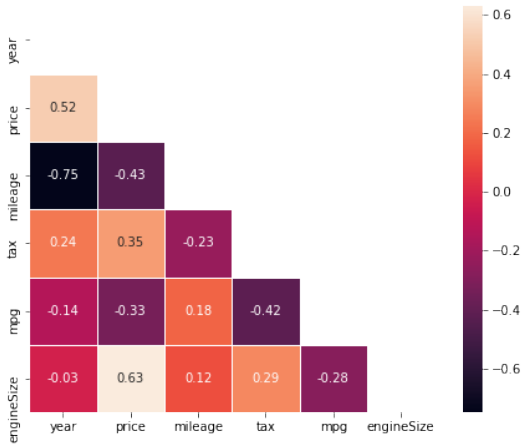

Figure 3: Heap map of correlations

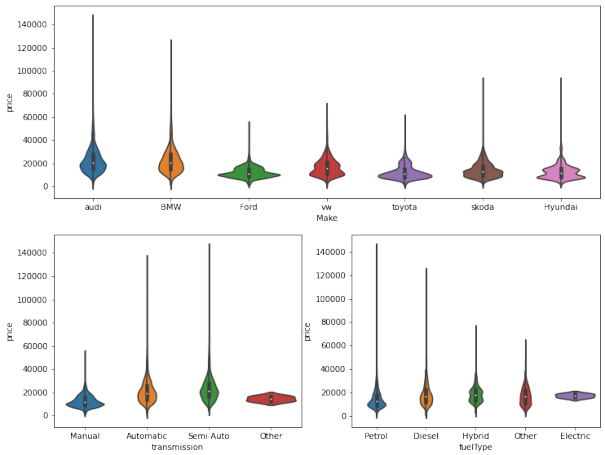

Figure 4: Violin plots of price against categorical features

2.3 Distribution of Categorical Features

The relationships between price and categorical features are visualized using the violin plots shown in Figure 4.

3 Data Preprocessing and Feature Engineering

During data collection, there are often some "out of range" values which will cause inaccuracy during data analysis. Data preprocessing enhances the quality of the data set by data cleaning and transformation. After preprocessing, the final data set can lead to more precise and reliable conclusion under data mining algorithms. In this report, based on our data set, we did the following steps:

1. `model` varies a lot and `mpg` makes no sense for electric and hybrid cars, so we dropped these two columns.

2. Applied log-transformations to make the data not so skewed.

3. Converted categorical attributes into dummies.

3.1 Log Transformation

Since the histograms of `year`, `price` and `mileage` are not bell-shaped as were shown in Figure 1, we applied log-transformations to reduce the skewness of our original data and make them more normal. For `price` and `mileage`, we replaced x with $\log(x)$; for `year`, we replaced x with $\log(2022 - x)$, where we defined $2022 - x$ as time length.

The distributions of transformed features, the relationships between transformed price and other features, as well as correlation heap map are shown in Figure 5.

3.2 Convert Categorical Data

For categorical attributes of `Make`, `transmission` and `fuelType`, we converted them into dummies, taking only the value 0 or 1 to indicate the presence of the corresponding categories since they may be expected to shift the outcome.

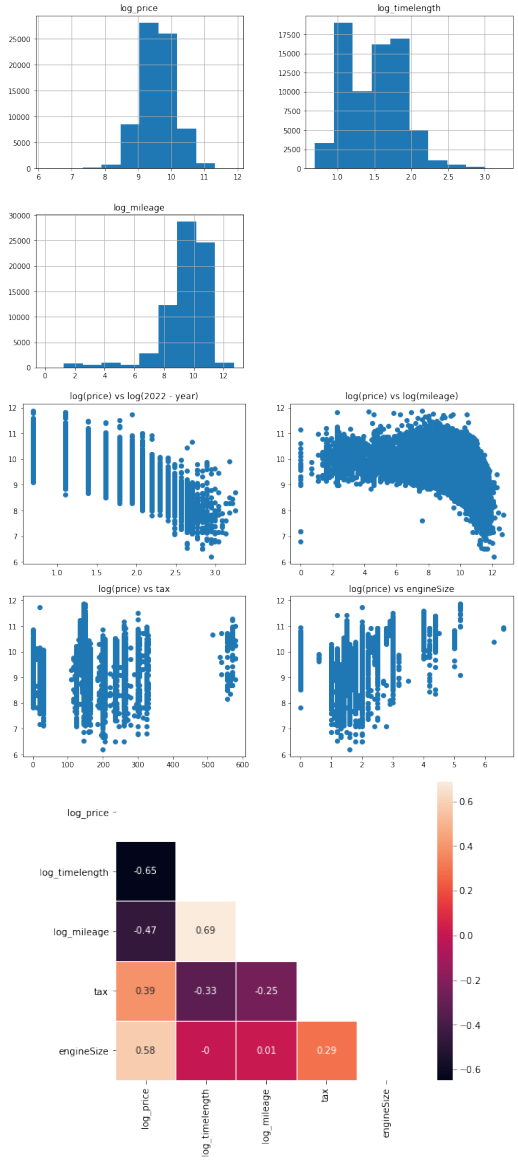

Figure 5: After log-transformation

Then, we ended up with 17 features to be used for predicting the prices of used cars.

4 Modeling and Analysis

After getting familiar with the data and doing some necessary transformations, we could start applying some regression models to predict the prices of used cars based on the attributes. To prevent the potential issue of overfitting, we used 80% of the data to train the models and the rest 20% to test the models. The model performances were measured and compared by the metric of R-square for the test set, which quantifies how much of the variation in the prices can be explained by the models. If y_i, $\hat{y}_i$ and μ_y represent the true value for the target variable, the predicted value for the target variable and the average of the true values respectively, then:

$$R^2 = 1 - \frac{SSE}{TSS} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \mu_y)^2}.$$

4.1 Simple Multiple Linear Regression

Multiple linear regression is one of the most simple regression models, so we used it as a baseline model. The predicted response value $\hat{y}_i$ for input $\mathbf{x}_i = (x_{i1}, x_{i2}, \cdots, x_{i17})^T$ is given as:

$$\hat{y}_i = b + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_{17} x_{i17} = b + \mathbf{w}^T \mathbf{x}_i$$

where $\mathbf{w} = (w_1, w_2, \cdots, w_{17})^T$ is the weight vector comprising the regression coefficients or weights w_j along each attribute X_j and b is the bias term [2]. The model is trained by finding the optimal weight vector and bias term that minimize the sum of squared errors for the training set, and the resulting coefficients sorted in decreasing order for all the 17 attributes are shown on Figure 6.

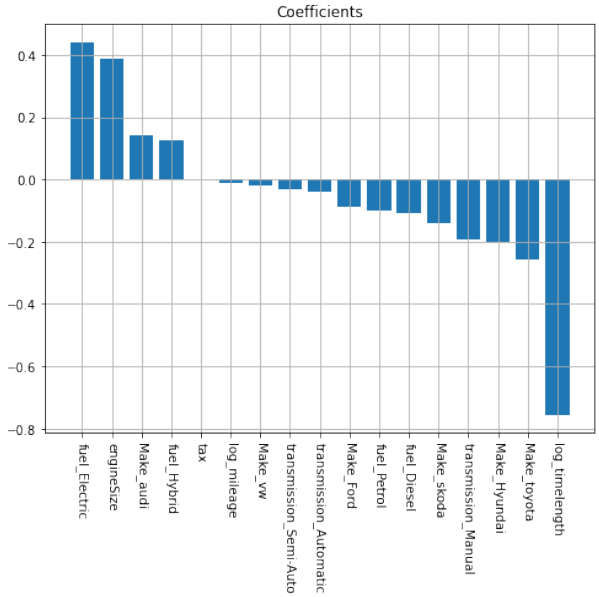

Figure 6: Simple multiple linear regression coefficients

Based on the figure, we can see that the fuel type of electric has the strongest positive influence to the price while the time length of the car has the most significant negative impact on the price. The R-square value for the test set of the baseline model is $R^2 = 0.8097$, which is pretty high but has the space of improvement.

4.2 LASSO Linear Regression

The result of the simple multiple linear regression shows some features have almost zero coefficients, which means those features might have little or even no contributions to the prices in reality, so we continued to run another linear regression model using the LASSO (least absolute

selection and shrinkage operator) regularization method. Instead of just minimizing the sum of squared errors for the training set, the weight vector in the LASSO model minimizes:

$$J(\mathbf{w}) = \frac{1}{2} \cdot \|\bar{Y} - \bar{\mathbf{X}}\mathbf{w}\|^2 + \alpha \cdot \|\mathbf{w}\|_1$$

where $\bar{\mathbf{X}} = \mathbf{X} - \mathbf{1} \cdot \boldsymbol{\mu}_{\mathbf{X}}^T$ is the matrix of the centered predictors for the training set, $\bar{Y} = Y - \mu_Y \cdot \mathbf{1}$ is the vector of centered targets for the training set, $\alpha \geq 0$ is the regularization constant and $\|\mathbf{w}\|_1 = \sum_{i=1}^{17} |w_i|$ is the L_1 norm of the weight vector.

The main advantage of using the L_1 norm is that it leads to sparsity in the solution vector, i.e. it can drive the coefficients to zero, resulting in a more interpretable model, especially when there are many predictor attributes [2], and allowing variable selection.

The most critical thing then was to select the regularization constant α, which represents the trade-off between the error term and the regularization term of the model, so we tried several different α ranging between 0 and 0.005, and plotted the resulting R-square for the test set against α on Figure 7.

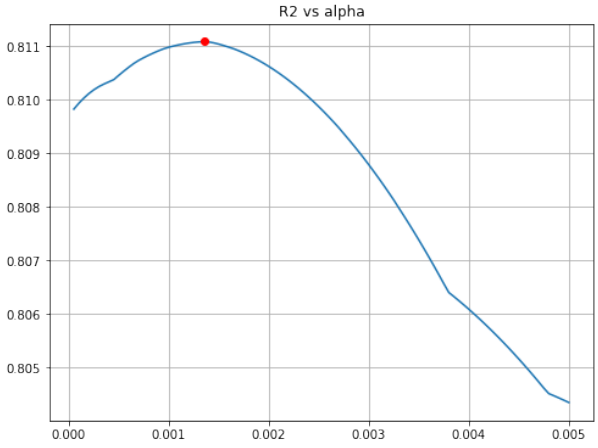

Figure 7: Relationship between R^2 and α

The optimal regularization constant is $\alpha = 0.00135$ yielding $R^2 = 0.8111$ for the test set, slightly higher than the one of the baseline model. Furthermore, this LASSO model drops the features of `Make vw`, `transmission Automatic`, `transmission Semi-Auto`, `fuel Diesel`, `fuel Electric` and `fuel Petrol`. The coefficients sorted in decreasing order for the rest attributes are shown on Figure 8.

Now, the engine size has the strongest positive influence to the price while the time length still has the most significant negative impact on the price.

4.3 Random Forest Regression

Since in reality, the process of determining the price is like making a decision, we thought decision trees might

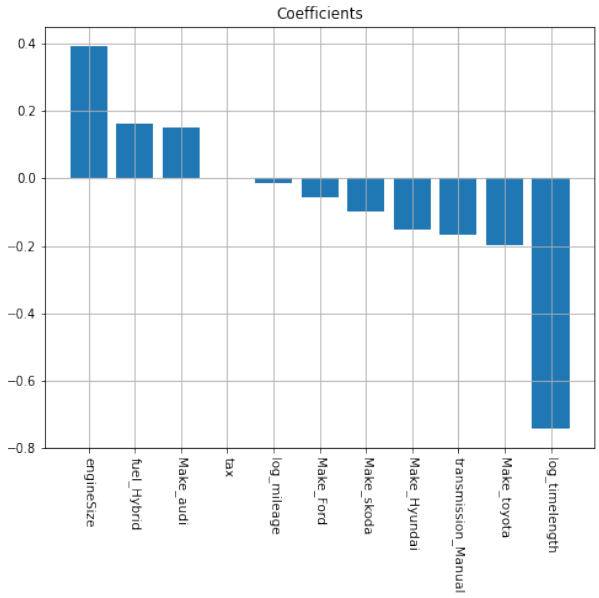

Figure 8: LASSO linear regression coefficients

be a good candidate model. Decision trees are generally used for classification, but the algorithm for training classification trees can be easily modified to train regression trees. In a regression tree, the node impurity is measured by the sum of squared deviations about the mean and the node prediction is the sample mean of corresponding targets. The regression tree algorithm yields piecewise constant models [3].

However, since piecewise constant models might not be very effective, and single tree is not stable and is likely to overfit the data, we chose random forest models which train a number of regression trees and make the prediction by averaging the outputs from all the trees. We used the default configuration from Scikit-Learn [4] to train the first version of model which contains 100 trees and uses all features and does not prune the trees during training. The resulting R-square for the test set is $R^2 = 0.8758$, higher than the previous two linear models. Figure 9 shows how important those features are in determining the prices of used cars in decreasing order.

The top three important features are time length, engine size and mileage. This matches exactly to the result of the LASSO model in terms of features with the three highest weights in absolute value.

Later we tried more configurations and found pruning the trees does not increase the model performance as a small value of 0.001 for the complexity parameter used for Minimal Cost-Complexity Pruning (`ccp_alpha`) can reduce the R-square for the test set greatly to 0.7473. Also, increasing the number of trees in the random forest can slightly increase the performance as 500 trees yield $R^2 = 0.8764$ while 1000 trees yield $R^2 = 0.8765$ for the test set.

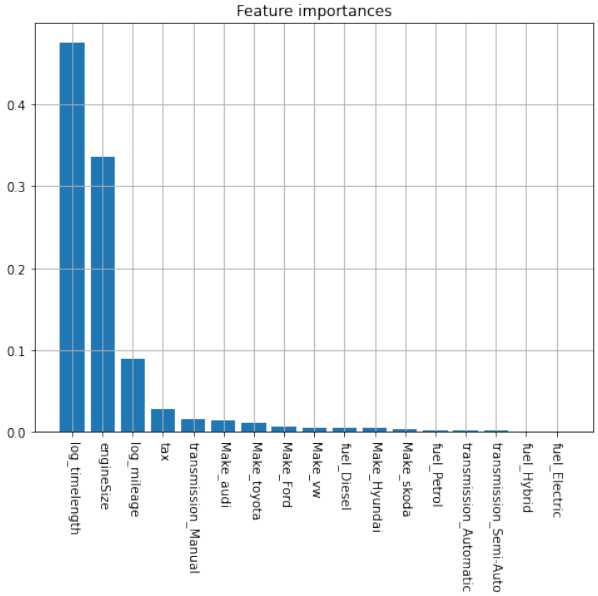

Figure 9: Feature importances in random forest

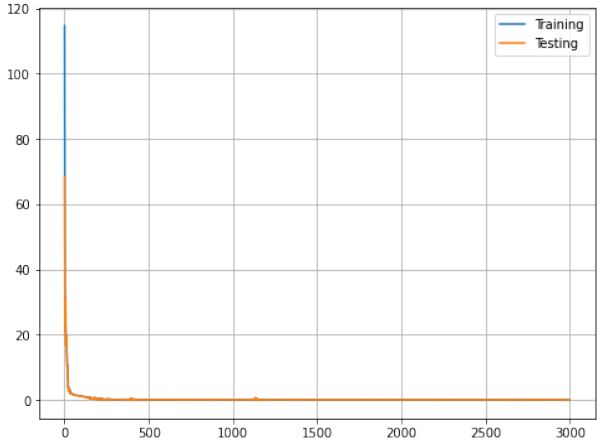

Figure 10: Training plot of MLP

4.4 Neural Network

Finally, we explored a more advanced model of neural networks. Since the number of features in the data is fixed and the data does not contain sequences or image structures, we just explored the feed-forward multilayer perceptron (MLP) model. The model we tried contains three hidden layers with 20, 50 and 20 neurons respectively, and neighboring layers are fully connected, i.e."each neuron in the input layer is connected to all the neurons in the first hidden layer, and each neuron in the first hidden layer is connected to all neurons in the second hidden layer, and so on, and finally, each neuron in the last hidden layer is connected to all neurons in the output layer"[2]. The activation function for each hidden layer is the leaky-relu function with negative slope of 0.01:

$$f(net_k) = \begin{cases} 0.01 net_k & \text{if } net_k \leq 0 \\ net_k & \text{if } net_k > 0 \end{cases} .$$

We trained the MLP model by back-propagation and gradient descent through PyTorch framework [5]. Figure 10 shows how the mean squared error (MSE) loss decreases during the training iterations: the blue curve is for the training set while the orange curve is for the test set.

We expected the more advanced neural network should have much better performance than the previous traditional machine learning techniques we tried, but unfortunately, the resulting R-square for the test set is only $R^2 = 0.8336$, higher than the linear models but lower than random forest models. Yet, since the two curves on the training plot shown in Figure 10 are very close to each other, we do not need to worry about overfitting at that moment, so we could further increase the complexity of the model by adding more hidden layers and increasing the number of hidden layer neurons. This could probably increase the model performance to some extent, but it requires huge computing power that is hard to achieve on laptops.

5 Conclusion

The regression models including simple multiple linear regression, LASSO linear regression, random forest regression and multilayer perceptron provide a quantitative analysis of how various attributes can affect the price of a used car. Basically, features like time length and mileage can negatively influence the price, while features such as engine size can positively impact the price. Based on the performances of the models measured by the R-square metric for the test set as well as the time and memory required to obtain these models, we found a random forest containing 500 regression trees and using all features and not pruning the trees is the optimal model, and we believed this model can provide good guidelines for used car sellers and buyers.

For future directions, there are several aspects that we want to implement and try to optimize the model. First is to try other transformation methods or use kernel tricks to mine the non-linear patterns instead of log transformation in our current preprocessing step. Secondly, we can explore other ensemble models such as boosting, stacking, etc. beyond random forests. Moreover, we will continue with more complex neural network structure if there is available computing power.

References

[1] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Shep-

pard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.

[2] M. J. Zaki and J. Wagner Meira, *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*. Cambridge University Press, second ed., 2020.

[3] W.-Y. Loh, "Classification and regression trees," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, pp. 14–23, 01 2011.

[4] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

[5] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.