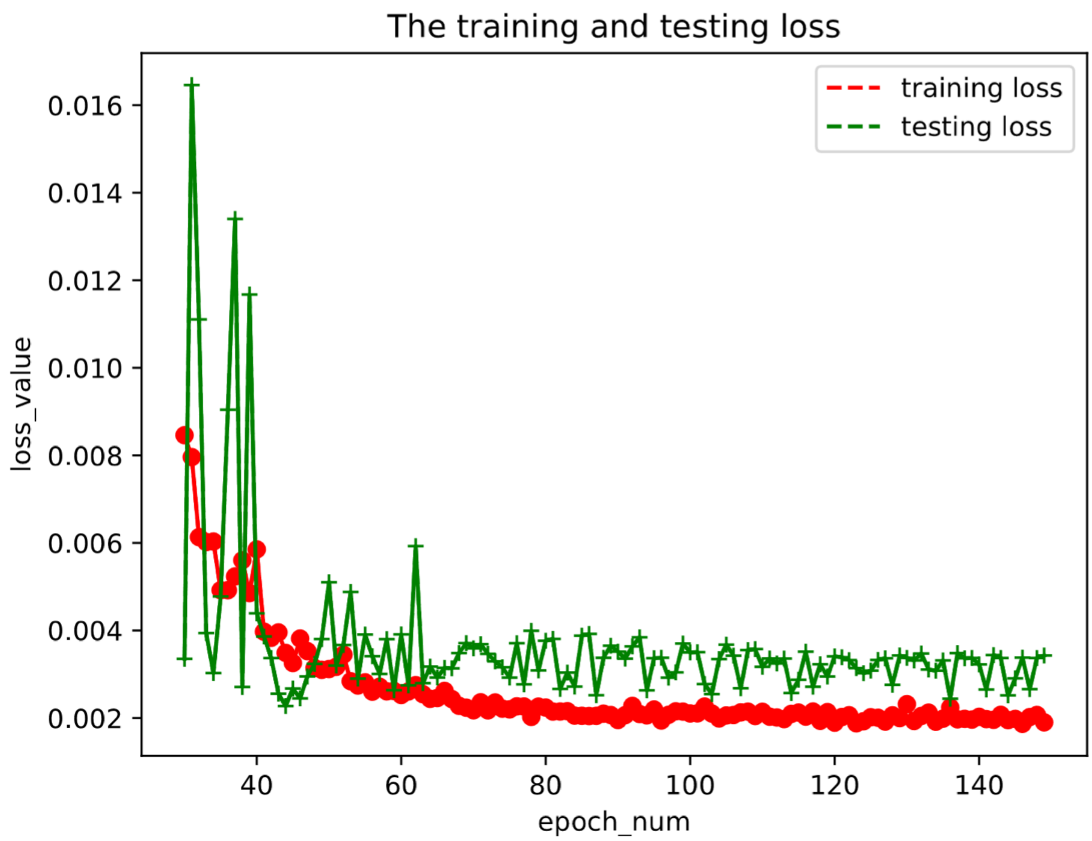


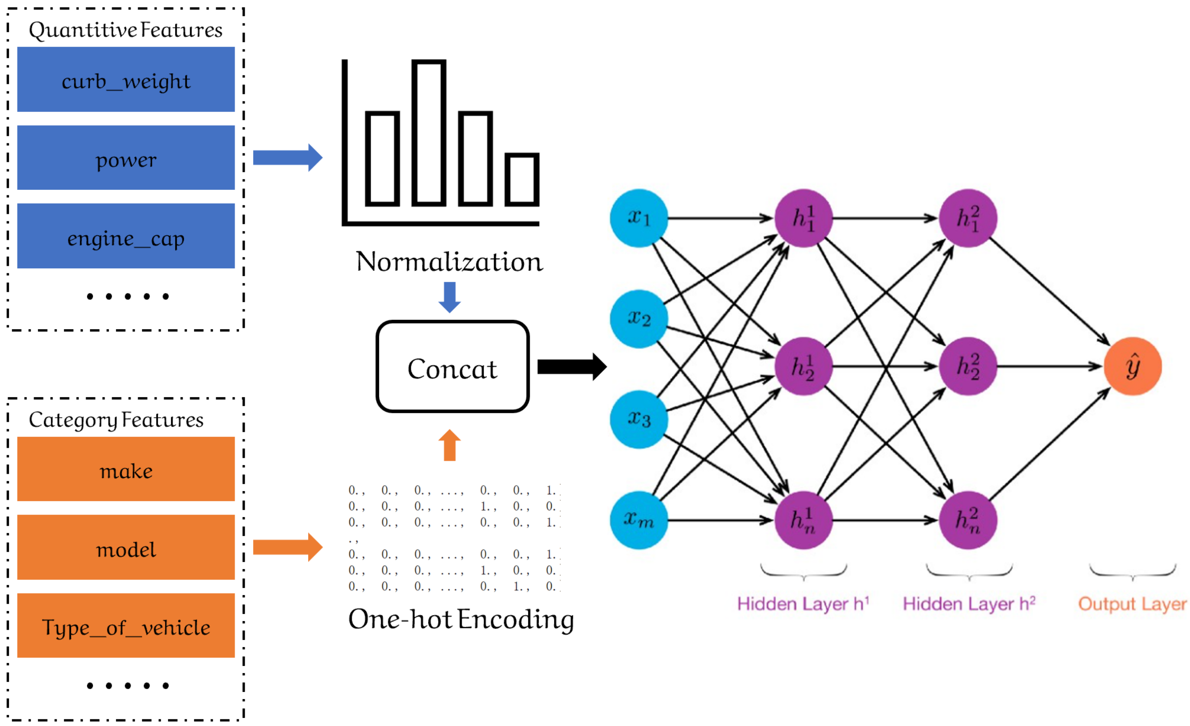


Figure 1 : The pipeline of proposed method.

Task1 Report

The background of the task1:

Recently, the size of the used car market has been continuously increasing. In this used car trading market, various used car trading damage is caused by the asymmetry of information between the seller and buyer. In particular, there are many cases of damage caused by information asymmetry between the seller and buyer, and the reliability of the used car trading market is deteriorating. In this project, we look into the market for used cars in Singapore. Car ownership in Singapore is rather expensive which includes the very high prices for new and used cars (compared to many other countries). There are many stakeholders in this market. Buyers and sellers want to find good prices, so they need to understand what affects the value of a car. Online platforms facilitating the sale of used cars, on the other hand, want to maximize the number of sales/transactions.

The goal of this task is to predict the resale price of a car based on its properties (e.g., make, model, mileage, age, power, etc). It is therefore first and foremost a regression task. These different types of information allow us to come up with features for training a regressor. It is part of the project for you to justify, derive and evaluate different features. Besides predicting outcome in terms of a dollar value, other useful results include the importance of different attributes, the evaluation and comparison of different regression techniques, an error analysis and discussion about limitations and potential extensions, etc.

Data Analysis:
The given data includes “train.csv” and “test.csv”. The prices of used cars are provided in “train.csv” and not provided in “test.csv”. There are many attributes of a car given in these two csv files, such as curb_weights, power, type_of_vehicle, make, model and so on. We simply divide these attributes into two types, one is quantitative attributes and the other is category attributes. To be specific, we show quantitative and category data in the following Table 1 and Table 2.

Table 1. The Quantitative Attributes

Attributes	Quantitative Attributes					
Name	curb_weight	power	engine_cap	no_of_owners	depreciation	coe
Name	road_tax	omv	dereg_value	mileage	arf	

Table 1. The Quantitative Attributes

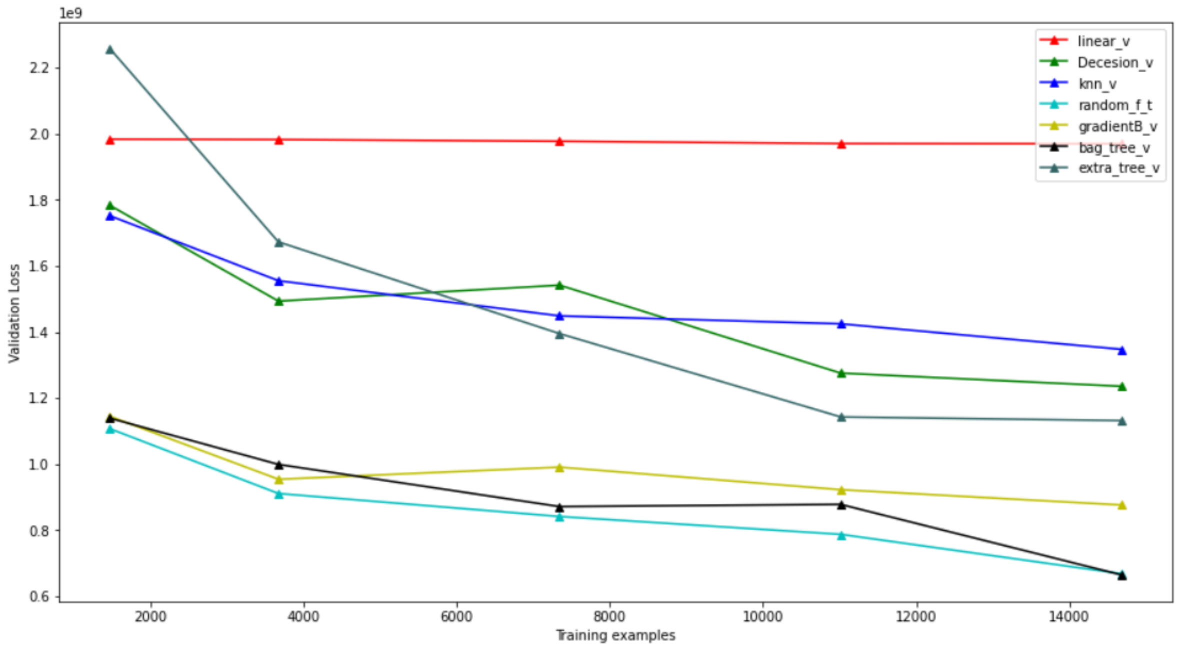
Attributes	Category Attributes					
Name	title	make	model	description	type_of_vehicle	transmission
Name	category					

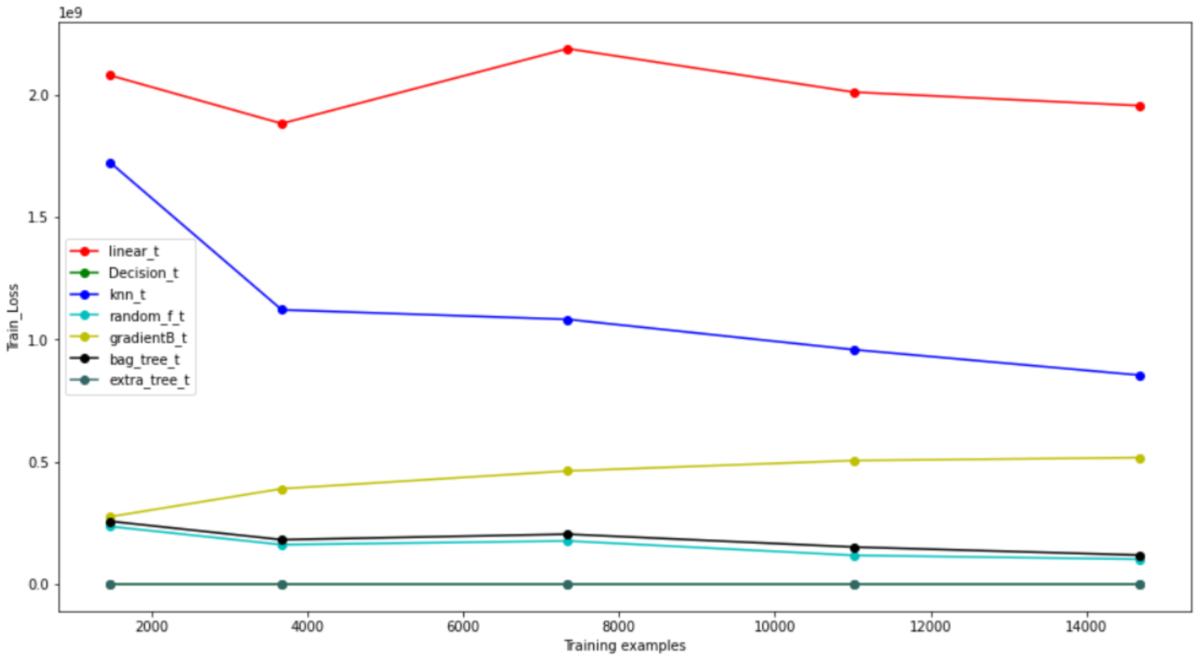
Data Processing:
We carefully check the data and have several observations as follows: 1). There are some NON-Value attributes of some lines. 2). There some attributes are not useful to the final results, such as the “listing_id”. 3). The divergence of some attributes is very large, so we need to normalize it. For quantitative attributes, we use mean and std to normalize the values to generate quantitative features. For category attributes, we utilize the one-hot

encoding to obtain the category features. Finally, we concatenate these features into fusion features.

Deep Learning method: After encode the features, we define a multi-layer perception (MLP) to regress the price of used cars. As shown in Figure 1, the features are fed into MLP which consists of 2 hidden layers. The input dimension is 817 and the hidden layer dimension is 1024.

Implementation Details: We train our network for 200 epochs using Adam, a starting learning rate of 0.001 and exponential decay, using mini-batches of size 64. Initially, the weights of our linear layers are set using Kaiming initialization. We implemented our code using Pytorch, which takes around 5ms for a forward+backward pass.





task3

To investigate which attributes impact the deprecation, we build two modules to lean this relationship. The first one is a non-deep learning method, such as random forest. The second one is a deep learning method, such as multiple perception layer (MLP).

The prediction of deprecation is very similar to the price regression, so we inherit the method of task1. We first process other attributes to generate one-dimension features. We