Machine Learning Engineer Nanodegree

Capstone Proposal

Dylan Chu
August 26th, 2018

(This proposal is written under the assumption that the reader is familiar with Kaggle competitions.)

Domain Background

Forecasting is a common activity in the business world. Being able to accurately forecast sales allows businesses to hold adequate levels of inventory to meet future demand, to smartly invest in capital equipment and to benefit in a myriad of other areas. I have not created sales forecasts in any professional capacity before but I imagine that looking at past data would be vital in order to make educated estimates of future sales. Of course, statistical methods such as moving averages could be employed for the task. However, it seems that machine learning can be an even more handy tool for generating sales forecasts.

The ability to predict customer behaviour is of personal interest to me as I currently work for a company where we need to make good predictions of customer behaviour on ecommerce websites. We observe and analyze the behaviour of customers as they shop and use that behaviour to determine the potentiality of their future actions. Using this data-based analysis, we may present offers to customers to incentivize them to perform certain activities on the website.

Problem Statement

The problem is a Kaggle competition called *Predict Future Sales* (1). The competition is a one-step forecasting problem. Competitors are given 34 months of product transactions from January 2013 to October 2015, and are asked to predict the sales of certain products at certain shops in November 2015. These pairings of a product and a shop are listed in the test file. Submissions are scored based on the metric described further in the *Evaluation Metrics* section.

Datasets and Inputs

The datasets for the Kaggle competition have been provided by the organizers. The table below provide more information about the data files that are provided.

File	Number of records	Description
shops.csv	60	reference file of mappings of shop id to shop name
item_categories.csv	84	reference file of mappings of item category id to item category name
items.csv	22170	reference file of mappings of item id to item name and item category id

sales_train_v2.csv	2935849	training file with sales transactions records
test.csv	214200	test file containing pairings of shop and item that require predictions

The two reference files for shops and item categories are not particular useful for the problem. The training file is a time series dataset with records of quantity changes (sales and returns) of an item at a shop on a particular date. For example, there is a record that on January 6th, 2015, a shop with id 25 sold 1 item with id 2554 at a price of 1709.05 units. The date range of the records span from January 2013 to October 2015.

Solution Statement

Since the training file consists of time series data, I can connect multiple records together such that I can use supervised learning algorithms on the problem (2). The supervised learning algorithm will require an input file where each record is in the form of X, y. X are the input variables and y is the target variable. Some of the current columns in the provided training file would be part of the input variables. The target variable is the sales total for a particular item at a particular shop in the next respective month. I will feed the input file to a supervised learning algorithm to train a model to predict the target variable. Predictions for the entries in the test file will be submitted to Kaggle and the score for the predictions will be compared to one for the benchmark model.

Benchmark Model

The benchmark model that I will use for comparison purposes is a submission for the competition by a user called *TrietChau* (3). It seems that the user created a kernel to teach beginners how to tackle this particular Kaggle challenge. *TrietChau* used a recurrent neural network (LSTM in particular) to perform the predictions. A LSTM network with its ability to retain memory of previous inputs seems appropriate for time series data. The score for the user's submission (based on the evaluation metric discussed in the next section) is 1.25011.

Evaluation Metrics

With the problem being a Kaggle competition, the organizers have already defined an evaluation metric to judge all submitted solutions. The metric they have chosen is the root mean squared error (RMSE). Therefore, for all pairings of shop and item in the test file, the difference between the actual sales amount and the predicted sales amount are squared and sum together before being divided by the number of pairings. The RMSE value is the square root of the result from the previous calculation. This seems like a suitable metric to judge the accuracy of the predictions.

Project Design

Since the Kaggle competition is a one-step forecasting problem, the solution is to train a model to do one-step forecasting. The steps involved in arriving at a solution are converting the training data, augmenting the data to create an input file, using the input file to train a model to forecast sales total for one month ahead and using the model to generate predictions to submit to the Kaggle competition to assess the performance of the model.

Converting the data
The given training file has 6 columns: date, date_block_num, shop_id, item_id, item_price, item_cnt_today. Each record in the training data file is for a day's transaction(s) of a particular product at a particular shop. These records should be aggregated so that each record in the training file represents a month's transactions for each unique pairing of shop and product. Then I need to append the target variable to each record. As mentioned earlier, this target variable is the total sales amount for that product and shop in the next respective month.

Augmenting the data
The training file does not include the category of the product. This seems like an important piece of information with which to train the model. However, feeding a category id (along with a shop id and an item id) does not help to improve a model predictive's power. Therefore, it is necessary to analyze the given data to extract some useful features of the shops, items or categories and augment the training data with those features. For example, a streetfront shop in a small neighbourhood is a different from a store in a popular shopping mall. It is much more beneficial to add an input variable of store "type" to the training data Similarly, I would feed the model the item category "type," not the item category id. Using clustering algorithms such as Gaussian Mixture Model could help to determine types for shops, items and categories.

Training the model
After creating an appropriate input file, I will need to select a suitable supervised learning algorithm for regression problems. Candidates include the different regressors provided by sklearn such as RandomForestRegressor. Using the input file and the selected algorithm, I will train a model to forecast the next month's sales of a product at a particular shop.

Assessing the performance of the model
I will need to take the data in the provided test file and use it to create another file similar in structure to the input file to train the model. Then I will feed this new test file to the model so it can predict the next month's sale forecasts for each shop/item combination in it. Use the predictions, I will create a submission file for the Kaggle competition. After submitting it, I hope the RMSE value will be less than the one for the benchmark model.

References

1. Kaggle. "Predict Future Sales." https://www.kaggle.com/c/competitive-data-science-predict-future-sales

2. Brownlee, Jason. Machine Learning Mastery. "Time Series Forecasting as Supervised Learning." https://machinelearningmastery.com/time-series-forecasting-supervised-learning/

3. TrietChau. Kaggle. "A beginner guide for sale data prediction." https://www.kaggle.com/minhtriet/a-beginner-guide-for-sale-data-prediction