Steven Vuong
Machine Learning Engineer Nanodegree Capstone Proposal
17/04/2020

Project Domain Background:

Kaggle Competition, “Predict Future Sales” (https://www.kaggle.com/c/competitive-data-science-predict-future-sales/overview). Snippet from the competition description: This competition also serves as the final project for the “How to win a data science competition”.

Historical Information: The context for a firm to predict sales can be very important, it can help them to make higher level business decisions. And in the context of a traditional store, direct executive decisions as to where to invest or cut back leading to big savings and optimisations that could benefit the organisation as a whole. With new techniques from research into Machine Learning, we can perhaps use this powerful tool to enable forecasting for this particular problem. As for research in, Machine Learning, much has been published in recent history.

Motivation: Time series forecasting has always been an interesting domain (personally at least) and this dataset provides a good opportunity to showcase the data science skills I have accumulated over this course and from past experience to try and “predict the future”.

Below are some examples of academic papers referencing the usage of machine learning techniques for time series forecasting:

- 25 years of research into time series foorrecasting - https://www.sciencedirect.com/science/article/abs/pii/S0169207006000021
- Time series forecasting using Hybrid ARIMA model - https://www.sciencedirect.com/science/article/abs/pii/S0925231201007020
- Neural Network forecasting for seasonal and trend forecasting - https://www.sciencedirect.com/science/article/abs/pii/S0377221703005484
- Sales Forecasting using neural networks - https://ieeexplore.ieee.org/abstract/document/614234
- Linear, Machine learning and probabilistic approaches to time series forecasting https://ieeexplore.ieee.org/abstract/document/7583582

Dataset can be found here: https://www.kaggle.com/c/competitive-data-science-predict-future-sales/data and is referrenced again below.

Problem Statement:

To predict total sales for every product and store in the next month, given historical sales data.

This is measured against a loss function (Root-mean squared error) for the test data against predicted data and is therefore quantifiable and measurable.

Our solution will be with a freely accessible dataset and will be run in a Jupyter Notebook with Python3 kernel, making it replicable.

There are also other possible solutions to this problem, found in other Kaggle competitor's submitted notebooks. One example is: https://www.kaggle.com/dlarionov/feature-engineering-xgboost

Dataset:

In this competition you will work with a challenging time-series dataset consisting of daily sales data, kindly provided by one of the largest Russian software firms - 1C Company.

There are multiple CSV files with sales data of stores and items sold with multiple data fields. It is intended that all of the files will at least be investigated and some exploratory data analysis will occur to determine what fields from the data will be useful or used.

It is required that for each item, a prediction is required for the total number of sales.

Data Files:

- sales_train.csv - the training set. Daily historical data from January 2013 to October 2015.
- test.csv - the test set. It is required to forecast the sales for these shops and products for November 2015.
- sample_submission.csv - a sample submission file in the correct format.
- items.csv - supplemental information about the items/products.
- item_categories.csv - supplemental information about the items categories.
- shops.csv- supplemental information about the shops

Data Fields:

- ID - an Id that represents a (Shop, Item) tuple within the test set
- shop_id - unique identifier of a shop
- item_id - unique identifier of a product
- item_category_id - unique identifier of item category
- item_cnt_day - number of products sold. You are predicting a monthly amount of this measure
- item_price - current price of an item
- date - date in format dd/mm/yyyy
- date_block_num - a consecutive month number, used for convenience. January 2013 is 0, February 2013 is 1,..., October 2015 is 33

- item_name - name of item
- shop_name - name of shop
- item_category_name - name of item category

I plan to use the entire training data (about 32 months of data, so approx 32*30days=940 rows of data) to train our model. There is no need to split into a test set as a test dataset is already provided. (Following evaluation, we could also incorporate the test set into our training set to make our model even more powerful). I plan to take about 10% of the training set for validation during training.

As I have not performed exploratory analysis of the dataset yet, I am unable to say which features I will use. However, I can imagine the following features will be used:

- Shop_id
- Item_id
- Item_category_id
- Item_cnt_day (target variable)
- Item_price
- Date
- Date_block_num
- Item_category_name

Further information can be found here:
https://www.kaggle.com/c/competitive-data-science-predict-future-sales/data

Solution Statement / Project Design:

Explore data: Preprocess, visualise, feature engineer and draw inferences

- Identify anomalies within the dataset with visual plots (1d line graphs, histograms, violin plots etc..) and eliminate or categorise
- Deal with missing entries, either remove or fill in with a selected average (mean/median/mode)
- Visualise spread of dataset, perhaps categorising by any given column within the dataset
- Wrangle, wrangle, wrangle until the data provides us with some direction as to how we might want to model it

Model Data: Model selection, hyperparameter selection/tuning, training model

- From examining other notebooks, considering using XGBoost model for a supervised learning approach as it is quite fast, and there is nice support for it in AWS SageMaker with high level APIs for hyper parameter tuning also (allowing us also to practice what we have learnt from this nanodegree!)
- I am also considering ensembling with another model, such as LightBGM. https://www.kaggle.com/arthurtok/introduction-to-ensembling-stacking-in-python (Thanks to the Udacity reviewer who referenced this)

Evaluate: Evaluate model with test data and assess loss function (RMSE). Use findings to determine whether we need to go back to exploration or modelling data

Benchmark Model:

The notebook in: https://www.kaggle.com/minhtriet/a-beginner-guide-for-sale-data-prediction has a RMSE loss of 1.2, I will attempt to get a loss function at least matching this and hopefully lower.