**CAPSTONE PROJECT**

HERMANN, SOH TOM

March 20, 2021

MACHINE LEARNING ENGINEER NANODEGREE

DEFINITION

**PROJECT OVERVIEW**

The importance of prediction in various sectors of modern industry has grown drastically in the last couple of decades. As example the major companies of energy industry use models to predict the demand in terms of energy consumption in a particular area to satisfy the demand and avoid lack of stock, in the retail industry forecasting model are widely integrated with the stock management in store in order to be always aligned with the demand. In this project I will focus on retail environment.

**PROBLEM STATEMENT**

The objective is to create a forecast model that can be use by a store manager on a weekly basis to update the stock in the warehouse. The following steps will be done to complete the project:

* Download the data and preprocess the data analyzing outliers, missing values, use to train the model
* Make the forecast with the benchmark model a naïve forecasting model
* Train a lightGBM model and a Prophet models
* Compare the solutions and check the advantages on both

**METRICS**

Measuring forecast accuracy (or error) is not an easy task as**there is no one-size-fits-all indicator**. Only experimentation will show you what Key Performance Indicator (KPI) is best for you. As you will see, each indicator will avoid some pitfalls but will be prone to others.

As you can see in the formula, MAPE divides each error individually by the demand, so it is skewed: high errors during low-demand periods will significantly impact MAPE. Due to this, optimizing MAPE will result in a strange forecast that will most likely undershoot the demand.

One of the first issues of this KPI is that it is not scaled to the average demand. If one tells you that MAE is 10 for a particular item, you cannot know if this is good or bad. If your average demand is 1000, it is, of course, astonishing. Still, if the average demand is 1, this is a very poor accuracy. To solve this, it is common to divide MAE by the average demand to get a %

ANALYSIS

**DATA EXPLORATION**

The train dataset has 2935849 rows × 6 columns sales starting from June 2013 to December 2015 collecting data from 45 stores daily, in this project since the objective is to provide a weekly forecast I will aggregate the data by summarizing the sales of each item on a week basis. The dataset is also provided with category data describing the store and the category of items (84 rows × 2 columns) from electronics games such as PSP to album music in this project I will focus on forecasting the amount of CD the store manager will sell.

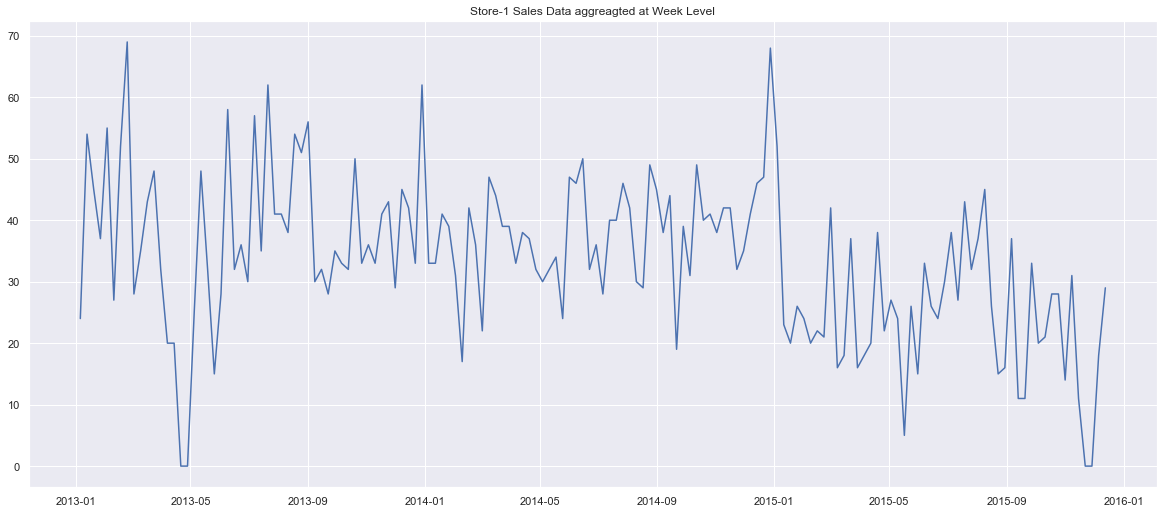
The collected data consist in:

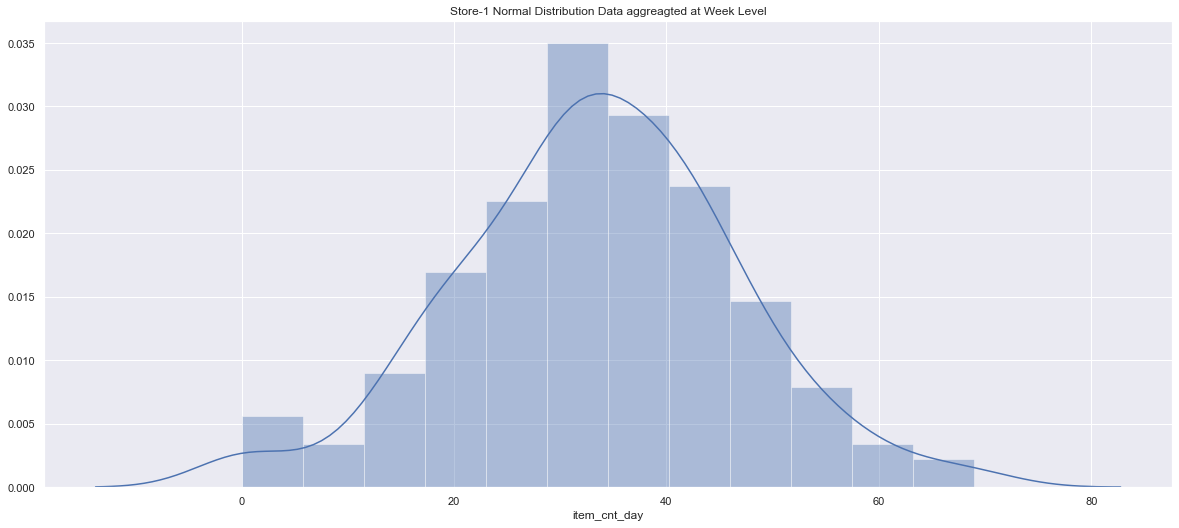
* Date the date where the item was sold
* Date\_block\_num an integer between 0 and 52 describing the week in the year
* Shop\_id the id of the shop the other inf can be retrieve in the shop\_id files provided
* Item\_id the numerical id of the item other info can be retrieve in the file provided
* Item\_price
* Item\_cnt\_day the number of item sales

a header of the dataset is shown below after merging the raw sales with all the categorical data associated.

**EXPLORATORY VISUALIZATION**

The plot below shows the sales for a specific all stores in the specific category I decided to explore, the sales are already aggregated based on the time horizon I choose and we have loose a level of granularity of our raw data I also plot the normal distribution of the sales data and a summary of missing data, in our case the missing value are completely absent of our dataset. .





**ALGORITHMS AND TECHNIQUES**

The LightGBM and the ARIMA, which are the state-of-the-art algorithm for most sales forecasting processing tasks. It needs a large amount of training data compared to other approaches; fortunately, the Training dataset are big enough.

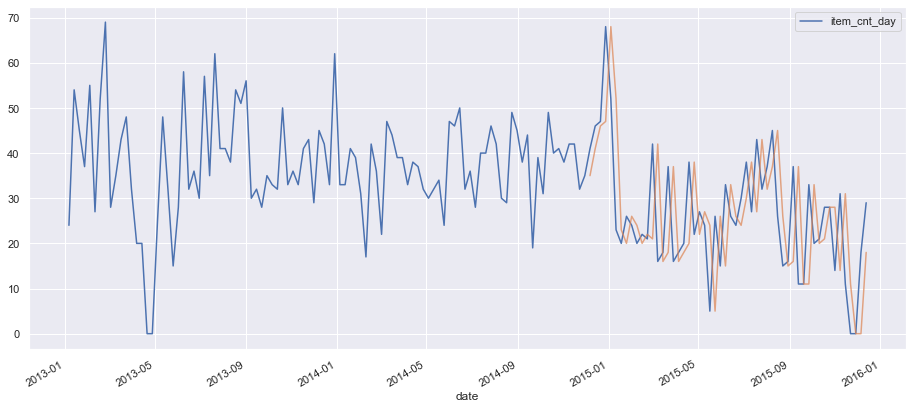
LightGBM supports both classification models and regression models. In our case, we set the objective function to mape which stands for mean absolute percentage error (MAPE) since we will build a regression model to predict product sales and evaluate the accuracy of the model using MAPE.

ARIMA model is a linear regression model that uses its own lags to predict the time series. The prerequisite for using ARIMA model forecasting is that the time series must be stationary. One common way to test whether a time series is stationary or not is Dickey Fuller test, if the p-value of the test is less than a significance level (normally 0.05), then we can infer that the time series is stationary.

**BENCHMARK**

To create an initial benchmark for the forecasting, A naïve forecasting will be use in this phase to make a comparison of the performance is simply the most recently observed value. In other words, at the time t, the k-step-ahead naive forecast equals the observed value at time t. For example, a store who has sells 10 items of a product has a high probability to sell the same numbers of items the next week.

The metrics for this approach with the provided dataset is display as below:



METHODOLOGY

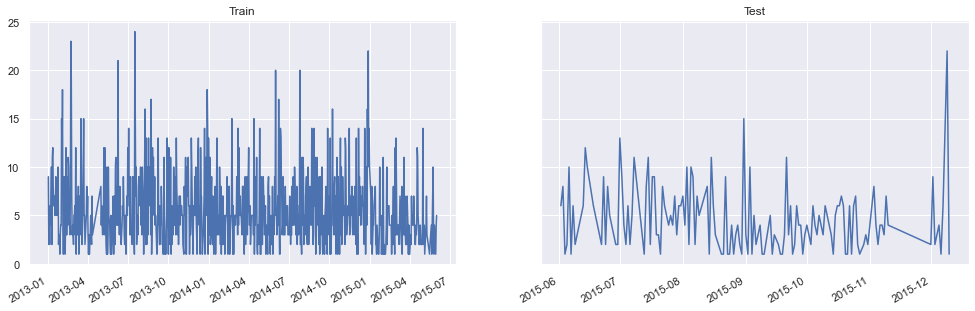
**DATA PREPROCESSING**

The preprocessing consists of the following steps:

1. The sales are imported and merge with the shop and the category
2. The sales are divided into a training set and a validation set

There are also some preprocessing steps which are done as the data get loaded before training:

1. The sales are resampling on a weekly basis
2. I define a split\_train\_test function that we split based on time basis to avoid the look-ahead bias, the train test is used to train the model this will also be subdivided in to dataframe train set and the validation set, sees the figures below.
3. A split\_data function is also implemented to split the train test into train and validation set based on the buit-in algorithm train\_test\_split of sklearn
4. The pixel values get divided by the standard deviation of the pixel values



**IMPLEMENTATION**

Two groups of models are implemented here: the statistical models, namely, ARIMA and a multivariate model, namely Light Gradient Boost. The data was split into two sets, namely, training and test sets, with splitting time set on the 1st June 2015. The analysis is conducted using the train set.

The hyperparameters for both the algorithm are then choose and a function is built to train the LGBM model, this function returns the model build and the predictions are made on the test set.

**LGBM**



This is mainly because, in practice the LGBM models perform quite competitively even without extensive hyperparameter tuning. To avoid the overhead of ensuring that the models do not overﬁt, we employ an early stopping mechanism to stop the training when an improvement in the validation error metric is not achieved over 5 consecutive epochs. The maximum number of epochs and the initial learning rate are speciﬁed as 1,200 and 0.075, respectively. MAPE is used as the objective function for training.

**ARIMA**

The ARIMA model has three parameters: the number of autoregressive terms (p), the number of times one must differentiate data to obtain a stationary data (d), and the number of moving average terms (q). As SARIMA is based on ARIMA, it has the same parameters plus parameters to indicate the seasonality in our case 12 month.



**REFINEMENT**

In both the model ARIMA and LGBM a hyperparameters tuning and validation phase can be improve the modularity of the code, by implementing a funtion to run the model and make the predictions has been think to be able to parametriza the parameters setting into the dictionary, in the arrima model the step of defining the hyperparameters con also be modularized in order to train the model and plot the learining curve and the validation curve. An hyperparameter tuning can be implemented for the LGBM model

We will use a “grid search” to iteratively explore different combinations of parameters. To validate the hyperparameters of the ARIMA model I opt to minize the AIC and choose the best value among the ones that returns the lowest value in a range of integer value. In estimating the amount of information lost by a model, AIC deals with the trade-off between the goodness of fit of the model and the simplicity of the model. In other words, AIC deals with both the risk of overfitting and the risk of underfitting. It takes around a half hour to find the right value for the model parameters.

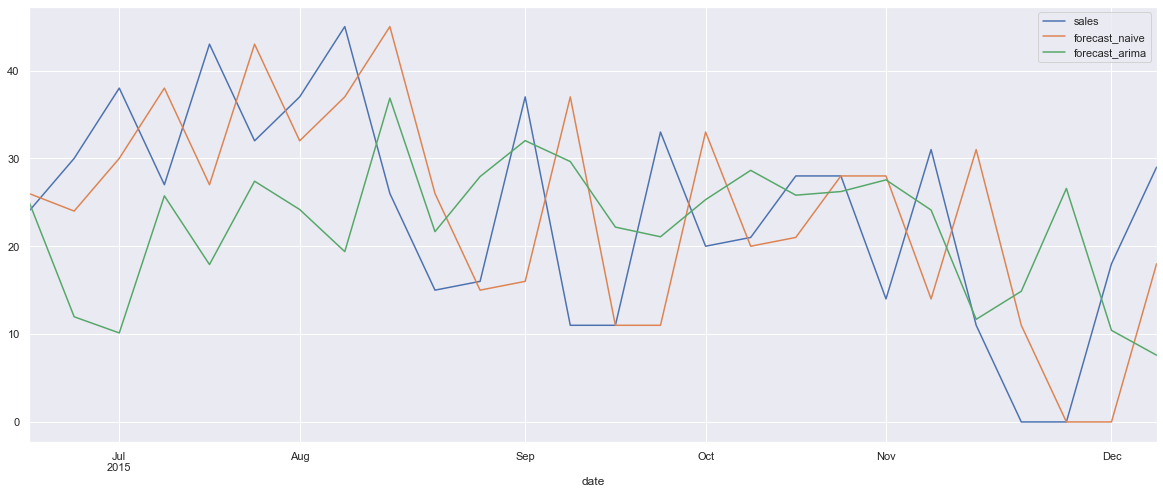
RESULTS

**MODEL EVALUATION AND VALIDATION**

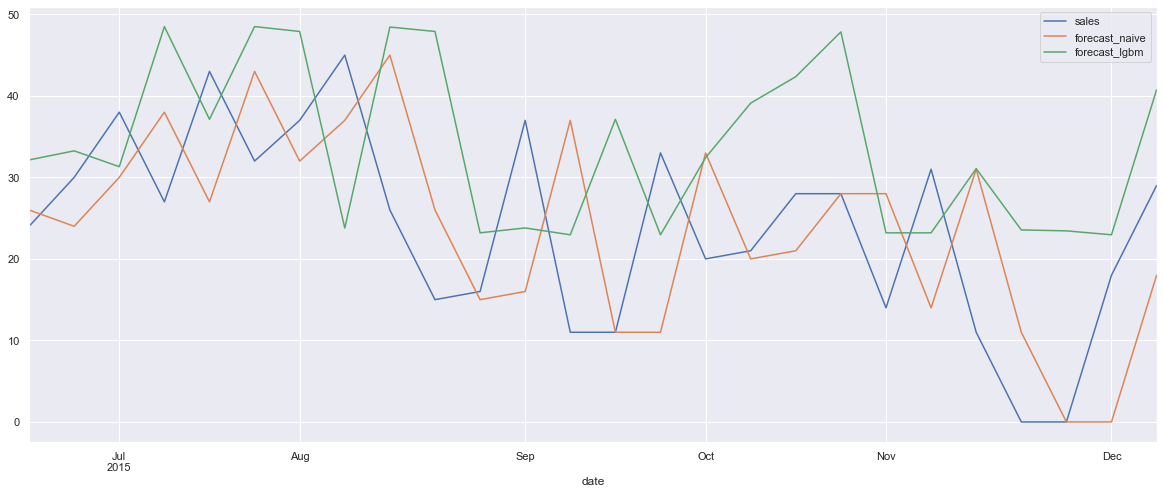
During development, a validation set was used to evaluate the model. The final architecture and hyperparameters were chosen because they performed the best among the tried combinations.

**ARIMA**

**MAE =** 11.57442872706511, MAE NAÏVE =

****

**LightGBM**

****

**JUSTIFICATION**

The advantages of LightGBM algorithm can be summarized as follows:

1. Shorter training time: LightGBM converts continuous feature values into discrete values which results in faster training process.
2. Lower memory usage: discrete values require less memory than continuous values.
3. High accuracy: decision tree built by leaf-wise growth approach is more complex than that of lever-wise growth approach. However, it may result in overfitting sometimes, this can be solved by tuning the hyperparameters in the model.
4. Compatibility with large data sets: even with large data set, LightGBM can perform equally good with shorter training time.

Among the main advantages of the ARIMA

The ARI

CONCLUSION

The scope of this project was to implements a relaiable techniques to forecast the demand in a given category present in the stock of a warehouse in thi case we focus on the CD industry and we try two state-of-art techiniques to forecast on a weekly basis.

The techniques tries were 2:

* ARIMA we implementeed a moving average
* LightGBM with a lag function we this techniques shows to perform the best among the ones tried in this report

**IMPROVEMENTS**

To be fully industrialize and autonomous a preprocessing phase is a critical part of a forecasting modelling techniques so a first step that can be implemented is the handling of the outliers and the missing values even if in this case we have a good enough data to ignore the preprocessing steps, a further steps should be consists to try another innovative techniques such as Prophet techniques the forecasting method implemented by facebook capable to handle feast days to predict the demand.

To achieve the optimal user experience, using more capable hardware and moving the text extraction 9 process from the cloud to the device would be essential. This would reduce the processing time and give access to the outputs of all the modules of the text extraction pipeline, which would, in turn, enable the following features: