GROCERY SALES FORESTING FOR SUPERMARKET

INFO7390 : Advances Data Science/Architecture, Spring 2018

Rohan Naik (naik.ro@husky.neu.edu)
Chaitanya Joshi (joshi.chai@husky.neu.edu)
Mayank Gangrade (gangrade.m@husky.neu.edu)

ABSTRACT

Product sales forecasting is a major aspect of purchasing management. Forecasts are crucial in determining inventory stock levels, and accurately estimating future demand for goods has been an ongoing challenge, especially in the Supermarkets and Grocery Stores industry.

If goods are not readily available or goods availability is more than demand overall profit can be compromised. As a result, sales forecasting for goods can be significant to ensure loss is minimized.

Additionally, the problem becomes more complex as retailers add new locations with unique needs, new products, ever transitioning seasonal tastes, and unpredictable product marketing.

In this analysis, a forecasting model is developed using machine learning algorithms to improve the accurately forecasts product sales. The proposed model is especially targeted to support the future purchase and more accurate forecasts product sales and is not intended to change current subjective forecasting methods.

A model based on a real grocery store's data is developed in order to validate the use of the various machine learning algorithms. In the case study, multiple regression methods are compared. The methods impact on forecast product availability in store to ensure they have just enough products at right time.

INTRODUCTION

Taken together, forecasting is a process where information about past experiences is transformed into the estimates for the future. In this way the sales forecasting can support the decision making of the managers regarding what to do next. Moreover, forecasting is used to make plans for the future. Evidently, it should reduce uncertainty in management regarding strategic decisions and allocating resources. Forecasting is continuous process of learning and adaptation. (Lakhani and Kleiner, 2014; Saffo, 2007)

In this project, we are trying to forecasts product sales based on the items, stores, transaction and other dependent variables like holidays and oil prices.

This is a Kaggle Competition[4] called "Corporación Favorita Grocery Sales Forecasting" where the task is to predict stocking of products to better ensure grocery stores please customers by having just enough of the right products at the right time.

For this particular problem, we have analyzed the data as a supervised learning problem. In order to forecasts the sales we have compared different regression models like Linear Regression, Decision Tree, ExtraTreeRegressor, Gradient Boosting, Random Forest and XgBoost. Further to optimize the results we have used multilayer perception (MLP: a class of feed forward artificial neural network) and LightGBM (gradient boosting framework that uses tree based learning algorithms).

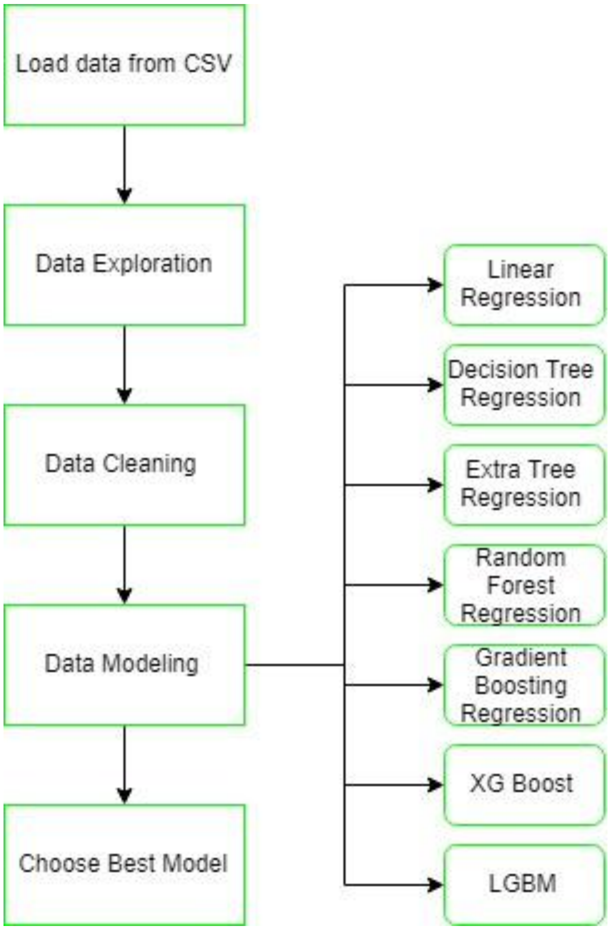
The data comes in the shape of multiple files. First, the training data (train.csv) essentially contains the sales by date, store, and item. The test data (test.csv) contains the same features without the sales information, which we are tasked to predict. The train vs test split is based on the date. In addition, some test items are not included in the train data.

BACKGROUND AND APPROACH

Sales forecasting is a process of predicting what the company's future sales are likely to be. Sales process is introduced, because it provides the essential data to be used to create in sales forecasts. The process of sales forecasting includes several steps. Although the aim of sale forecasting is to provide reliable information, it is not free from biases, which can occur in forecasting in many ways. The concept of predictive analytics is introduced as it is the key concept to predict future events.

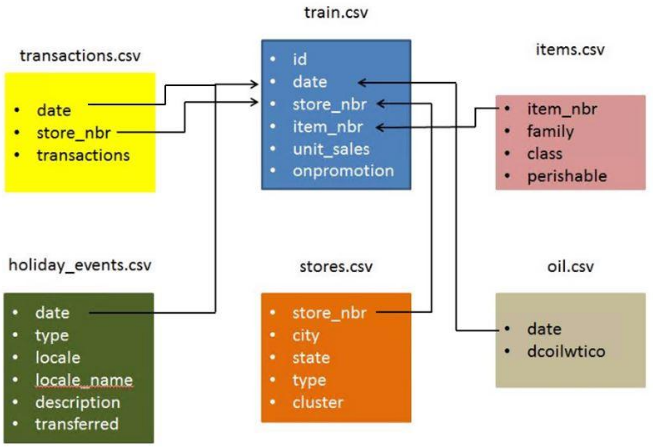
Regression analysis is a primary statistical tool that is used for predictive analytics in the organizations. In this way of modeling: an analyst makes first a hypothesis. He selects a set of independent variables. Variables could be such as store, item and number of unit sales at a particular store. It is assumed that variables are statistically correlated with the purchase of a product. A sample of the data is chosen. An analyst performs regression analysis and during the process he finds the best variables for the model to explain product purchase. The variation of the sales is explained by the all variables together. Regression coefficients are used for creating a score that predicts the likelihood of the purchase.

OVERALL FLOW



DATASET AND FEATURES

Data frames and their linkages: The arrows connecting the variable names show which of them can be used to merge data frames.



train.csv : consist of sales data on the basis of dates, store and item information, whether that item was being promoted, as well as the unit sales. The data span a period of 5 years from 2013 to 2017.

stores.csv : Consist of Store metadata, including city, state, type, and cluster (cluster is a grouping of similar stores).

items.csv : Item metadata, such as class and whether they are perishable. Note, that perishable items have a higher scoring weight than others.

transactions.csv : Count of sales transactions for each date, store_nbr combination.

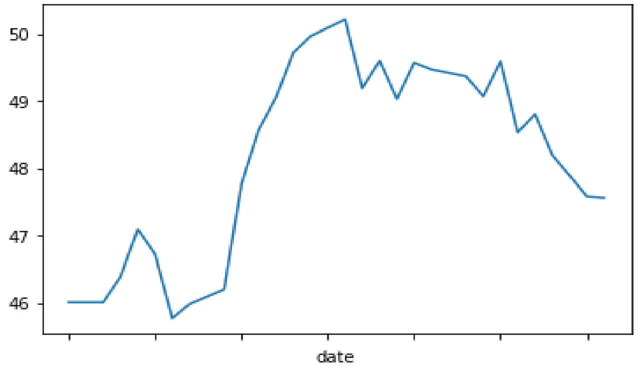
oil.csv : Daily oil price. This is relevant, because “Ecuador is an oil-dependent country and its economical health is highly vulnerable to shocks in oil prices.” (source).

holidays_events.csv : Holidays in Ecuador. Some holidays can be transferred to another day (possibly from weekend to weekday).

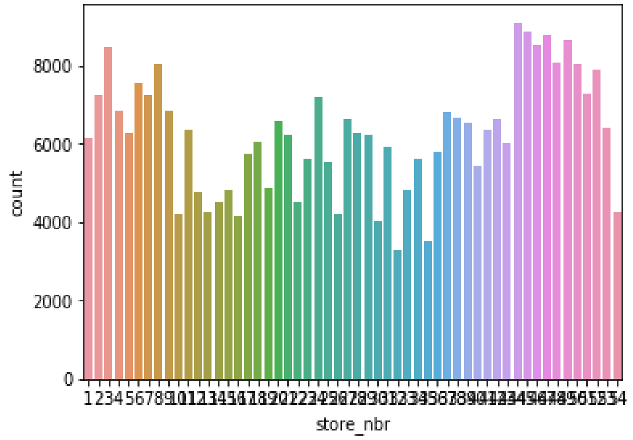
Data has been downloaded and collected from the following URL: https://www.kaggle.com/jeru666/all-csv-files-a-glance/data

EXPLORATORY DATA ANALYSIS

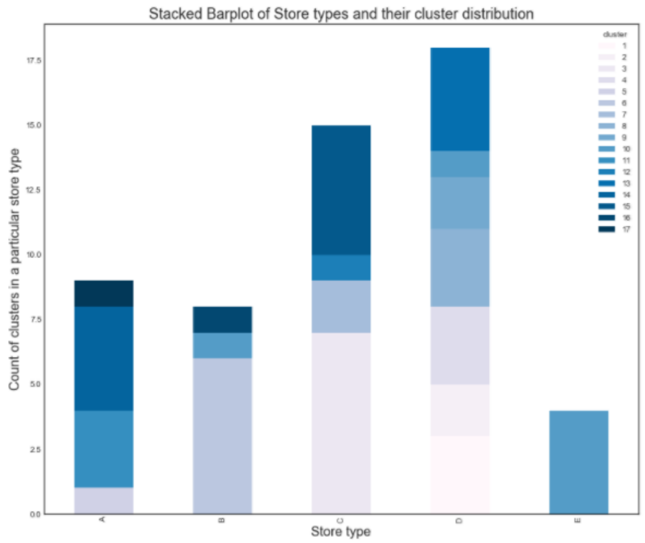
1. Oil values with respect to time:



2. Number of stores according to each store type:



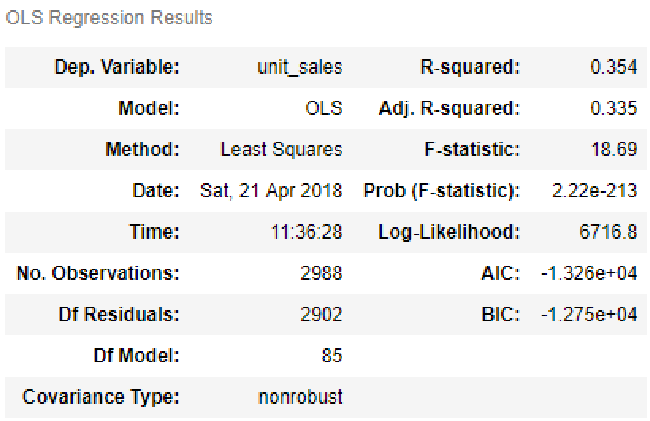
3. Cluster distribution across the stores type



MODELING

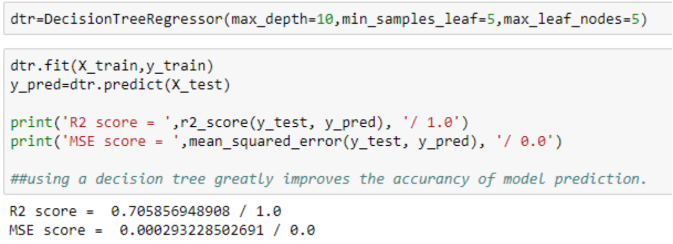
We have implemented different machine learning algorithms in Python to select the best performing model.

1. Linear Regression: Linear Regression is a linear approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. In our case we have applied simple linear regression.


OLS Regression Results

Dep. Variable:	unit_sales	R-squared:	0.354
Model:	OLS	Adj. R-squared:	0.335
Method:	Least Squares	F-statistic:	18.69
Date:	Sat, 21 Apr 2018	Prob (F-statistic):	2.22e-213
Time:	11:36:28	Log-Likelihood:	6716.8
No. Observations:	2988	AIC:	-1.326e+04
Df Residuals:	2902	BIC:	-1.275e+04
Df Model:	85		
Covariance Type:	nonrobust		

2. Decision Tree Regression: A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.


```
dtr=DecisionTreeRegressor(max_depth=10,min_samples_leaf=5,max_leaf_nodes=5)
```

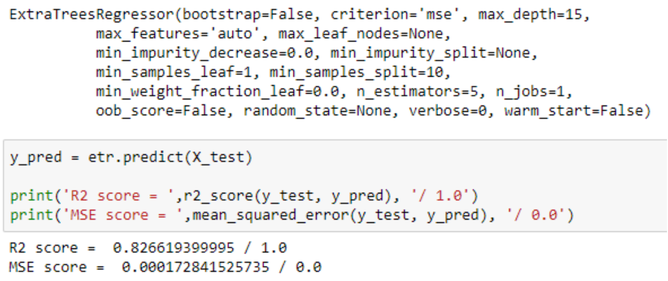
```
dtr.fit(X_train,y_train)
y_pred=dtr.predict(X_test)

print('R2 score = ',r2_score(y_test, y_pred), '/ 1.0')
print('MSE score = ',mean_squared_error(y_test, y_pred), '/ 0.0')

##using a decision tree greatly improves the accurancy of model prediction.
```

```
R2 score =  0.705856948908 / 1.0
MSE score =  0.000293228502691 / 0.0
```

3. Extra Tree Regression: Extra-trees differ from classic decision trees in the way they are built. When looking for the best split to separate the samples of a node into two groups, random splits are drawn for each of the max_features randomly selected features and the best split among those is chosen.

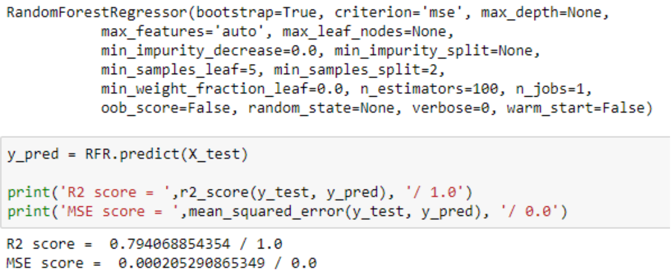

```
ExtraTreesRegressor(bootstrap=False, criterion='mse', max_depth=15,
          max_features='auto', max_leaf_nodes=None,
          min_impurity_decrease=0.0, min_impurity_split=None,
          min_samples_leaf=1, min_samples_split=10,
          min_weight_fraction_leaf=0.0, n_estimators=5, n_jobs=1,
          oob_score=False, random_state=None, verbose=0, warm_start=False)
```

```
y_pred = etr.predict(X_test)

print('R2 score = ',r2_score(y_test, y_pred), '/ 1.0')
print('MSE score = ',mean_squared_error(y_test, y_pred), '/ 0.0')
```

```
R2 score =  0.826619399995 / 1.0
MSE score =  0.000172841525735 / 0.0
```

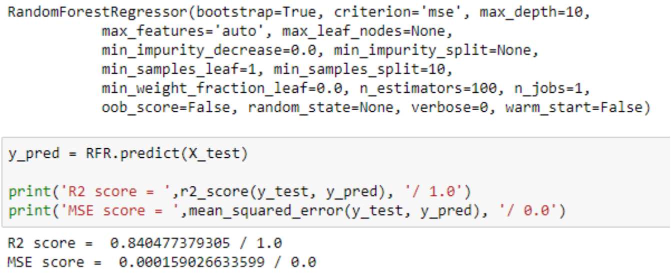
4. Random Forest Regression: Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set.


```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
           max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=5, min_samples_split=2,
           min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=1,
           oob_score=False, random_state=None, verbose=0, warm_start=False)
```

```
y_pred = RFR.predict(X_test)

print('R2 score = ',r2_score(y_test, y_pred), '/ 1.0')
print('MSE score = ',mean_squared_error(y_test, y_pred), '/ 0.0')
```

```
R2 score =  0.794068854354 / 1.0
MSE score =  0.000205290865349 / 0.0
```

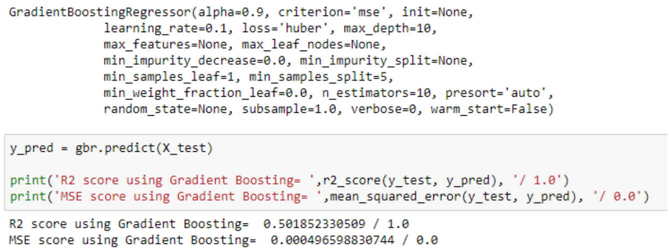

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=10,
           max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=1, min_samples_split=10,
           min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=1,
           oob_score=False, random_state=None, verbose=0, warm_start=False)
```

```
y_pred = RFR.predict(X_test)

print('R2 score = ',r2_score(y_test, y_pred), '/ 1.0')
print('MSE score = ',mean_squared_error(y_test, y_pred), '/ 0.0')
```

```
R2 score =  0.840477379305 / 1.0
MSE score =  0.000159026633599 / 0.0
```

5. Gradient Boosting Regression: The idea of boosting came out of the idea of whether a weak learner can be modified to become better. A weak hypothesis or weak learner is defined as one whose performance is at least slightly better than random chance. Hypothesis boosting was the idea of filtering observations, leaving those observations that the weak learner can handle and focusing on developing new weak learns to handle the remaining difficult observations.

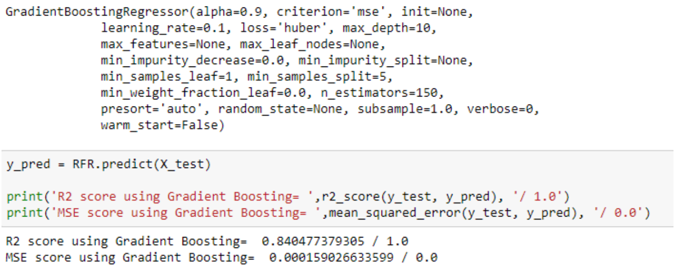


```
GradientBoostingRegressor(alpha=0.9, criterion='mse', init=None,
             learning_rate=0.1, loss='huber', max_depth=10,
             max_features=None, max_leaf_nodes=None,
             min_impurity_decrease=0.0, min_impurity_split=None,
             min_samples_leaf=1, min_samples_split=5,
             min_weight_fraction_leaf=0.0, n_estimators=10, presort='auto',
             random_state=None, subsample=1.0, verbose=0, warm_start=False)
```

```
y_pred = gbr.predict(X_test)

print('R2 score using Gradient Boosting= ',r2_score(y_test, y_pred), '/ 1.0')
print('MSE score using Gradient Boosting= ',mean_squared_error(y_test, y_pred), '/ 0.0')
```

```
R2 score using Gradient Boosting=  0.501852330509 / 1.0
MSE score using Gradient Boosting=  0.000496598830744 / 0.0
```



```
GradientBoostingRegressor(alpha=0.9, criterion='mse', init=None,
             learning_rate=0.1, loss='huber', max_depth=10,
             max_features=None, max_leaf_nodes=None,
             min_impurity_decrease=0.0, min_impurity_split=None,
             min_samples_leaf=1, min_samples_split=5,
             min_weight_fraction_leaf=0.0, n_estimators=150,
             presort='auto', random_state=None, subsample=1.0, verbose=0,
             warm_start=False)
```

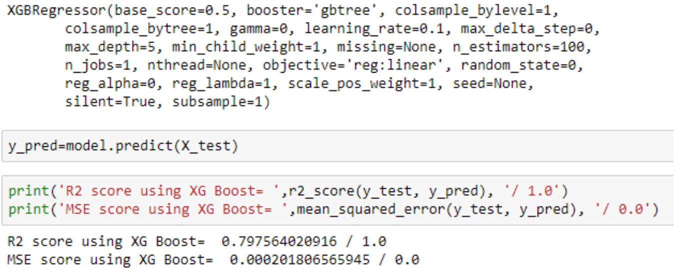
```
y_pred = RFR.predict(X_test)

print('R2 score using Gradient Boosting= ',r2_score(y_test, y_pred), '/ 1.0')
print('MSE score using Gradient Boosting= ',mean_squared_error(y_test, y_pred), '/ 0.0')
```

```
R2 score using Gradient Boosting=  0.840477379305 / 1.0
MSE score using Gradient Boosting=  0.000159026633599 / 0.0
```

6. XGBoost: XGBoost (eXtreme Gradient Boosting) is a direct application of Gradient Boosting for decision trees Main advantages are as follows:

- Easy to use
- Computational efficiency
- Model Accuracy
- Feasibility—easy to tune parameters and modify objectives



```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bytree=1, gamma=0, learning_rate=0.1, max_delta_step=0,
       max_depth=5, min_child_weight=1, missing=None, n_estimators=100,
       n_jobs=1, nthread=None, objective='reg:linear', random_state=0,
       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
       silent=True, subsample=1)
```

```
y_pred=model.predict(X_test)
```

```
print('R2 score using XG Boost= ',r2_score(y_test, y_pred), '/ 1.0')
print('MSE score using XG Boost= ',mean_squared_error(y_test, y_pred), '/ 0.0')
```

```
R2 score using XG Boost=  0.797564020916 / 1.0
MSE score using XG Boost=  0.000201806565945 / 0.0
```

COMPARISON OF MODEL

Model	R2 Score
Linear Regression	0.354
Decision Tree Regression	0.825
Extra Tree Regression	0.825
Random Forest Regression	0.836
Gradient Boosting	0.836
XG Boost	0.797
LGBM	0.759

CONCLUSION

This project on machine learning was a lot of fun, and also serves as an example of how different machine learning solutions can be from real-world implementations. Decision tree solutions are really suitable for building a recommendation system. It can train this system based on all features, producing more and more accurate points of each branch. Based this iteration process, we can train our train dataset and give a prediction for target data.

Also boosting technique like Gradient boosting and XG Boost plays import role in regression and classification problem during dealing with huge data and produces fast and accurate prediction model.

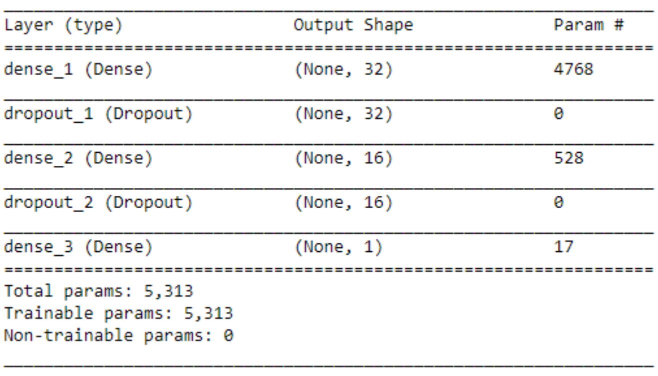
Finally, we figured out that Gradient Boosting was more accurate and fast as compare to Extra tree and decision tree with an accuracy of 83.6% in our case.

FUTURE SCOPE

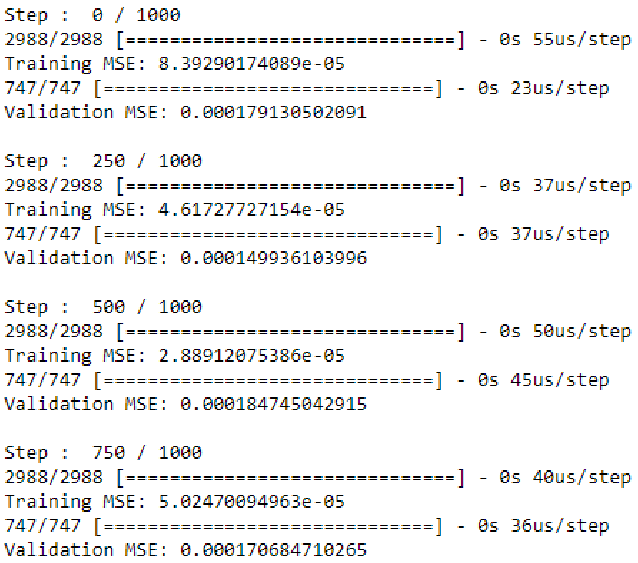
Mathematical models are typically used to model a system when the system is not so complicated, but when the complexity of a system is increased other methods should be used for modeling. Fuzzy theory is typically used when the behavior of system is the most complicated or the linguistic rules are needed to define the behavior of a system. But in a condition between above-mentioned Artificial neural networks (ANN) are good modeling method which can produce good results.

[12]Artificial neural networks are algorithms that can be used to perform nonlinear statistical modeling and provide a new alternative to regression, the most commonly used method for developing predictive models. Neural networks offer a number of advantages, including requiring less formal statistical training, ability to implicitly detect complex nonlinear relationships between dependent and independent variables, ability to detect all possible interactions between predictor variables, and the availability of multiple training algorithms.

Multi-layer Perceptron[13] (MLP) is a supervised learning algorithm that learns a function by training on a dataset, where is the number of dimensions for input and is the number of dimensions for output.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_1 (Dense)              (None, 32)                4768      
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0         
_________________________________________________________________
dense_2 (Dense)              (None, 16)                528       
_________________________________________________________________
dropout_2 (Dropout)          (None, 16)                0         
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 17        
=================================================================
Total params: 5,313
Trainable params: 5,313
Non-trainable params: 0
_________________________________________________________________
```

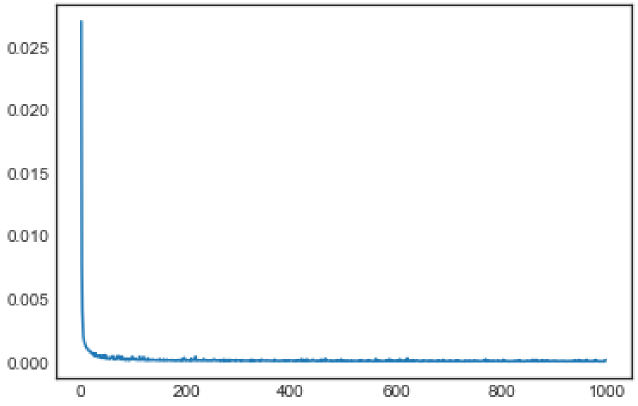


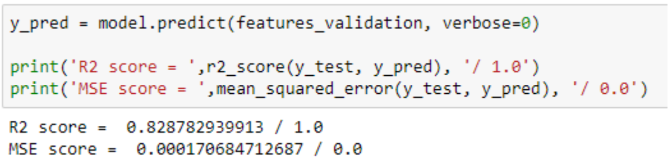
```
Step :  0 / 1000
2988/2988 [==============================] - 0s 55us/step
Training MSE: 8.39290174089e-05
747/747 [==============================] - 0s 23us/step
Validation MSE: 0.000179130502091

Step :  250 / 1000
2988/2988 [==============================] - 0s 37us/step
Training MSE: 4.61727727154e-05
747/747 [==============================] - 0s 37us/step
Validation MSE: 0.000149936103996

Step :  500 / 1000
2988/2988 [==============================] - 0s 50us/step
Training MSE: 2.88912075386e-05
747/747 [==============================] - 0s 45us/step
Validation MSE: 0.000184745042915

Step :  750 / 1000
2988/2988 [==============================] - 0s 40us/step
Training MSE: 5.02470094963e-05
747/747 [==============================] - 0s 36us/step
Validation MSE: 0.000170684710265
```





```
y_pred = model.predict(features_validation, verbose=0)

print('R2 score = ',r2_score(y_test, y_pred), '/ 1.0')
print('MSE score = ',mean_squared_error(y_test, y_pred), '/ 0.0')
```

```
R2 score =  0.828782939913 / 1.0
MSE score =  0.000170684712687 / 0.0
```

ACKNOWLEDGEMENT

We would like to show our gratitude to professor Nicholas Brown for guiding us and encouraging us during the course of this project.

GITHUB LINK

https://github.com/mgangrade7/INFO-7390-ADS-Project.git

REFERENCES

[1] Cui, G., Wong, M. L., & Lui, H. K. (2006). Machine learning for direct marketing response models: Bayesian networks with evolutionary programming. Management Science, 52(4), 597-612

[2] Taylor, E. L. (2014). Predicting Consumer Behavior. Research World, 2014(46), 67-68

[3] Morwitz, V. G., Steckel, J. H., & Gupta, A. (2007). When do purchase intentions predict sales?. International Journal of Forecasting, 23(3), 347-364

[4] https://www.kaggle.com/c/favorita-grocery-sales-forecasting/data

[5] https://en.wikipedia.org/wiki/Xgboost

[6] https://en.wikipedia.org/wiki/Random_forest

[7] https://en.wikipedia.org/wiki/Decision_tree

[8]https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vsxgboost/

[9] https://www.tutorialspoint.com/sales_forecasting/sales_forecasting_discussion.html

[10] https://www.datawatch.com/what-is-data-blending/

[11]https://www.theseus.fi/bitstream/handle/10024/106191/Haataja_Timo.pdf?sequence=1&isAllowed=y

[12] https://www.ncbi.nlm.nih.gov/pubmed/8892489

[13]http://scikitlearn.org/stable/modules/neural_networks_supervised.html