Syrian Commodities Price Prediction using machine learning algorithm

By:

Ula Yousef

Artificial Intelligence Engineer

Data Science Enthusiast

Abstract: this project includes an analytic study of commodities price in Syria in the last 12 years and aims to predict their prices in the future using machine learning techniques and algorithms.

Keywords: Syrian commodity, price, prediction, machine learning, regression model.

Introduction: during the recent years in Syria, there was a significant increasing in prices especially after the economic sanctions, therefor it was necessary to develop prediction model which plays an important role in agriculture field to maximize profits and minimizing risks, it can also help industry section which leads to optimization of resource allocation.

Project steps:

1. Exploratory data Analysis (data cleaning, data visualization).
2. Data Preprocessing.
3. Model Development.
4. Model Evaluation.

Exploratory data Analysis

A. Data Collection: The data set that used in this project is a real data stored in Global Food Prices Database and has provided by WFP (World Food Program), it consists of 14 field and 182779 samples:

Date	Timestamp of commodity price.
admin1	The name of the city that produces or sells the commodity.
admin2	The name of district in the admin1.
market	The name of the market.
latitude	Horizontal lines that measure distance north or south of the equator.
longitude	Vertical lines that measure east or west of the meridian in Greenwich.
category	Group to which food items assigned based on similarity.
commodity	The type of product that belongs to specific category.
unit	measurement unit like kg, L, g, packet,…..etc.
priceflag	If the price is aggregate or actual.
pricetype	The type of a price specified for an offered product.
currency	Name of money that serves as a means of exchanging commodities and services and in our data is SYP(Syrian Pounds).
price	Price in Syrian pounds
usdprice	Price in dollars

B. Data Cleaning: Clean the dataset from null, duplicated values and unnecessary features, after checking the dataset using pandas function I didn't find any null or duplicated values but it was necessary to delete some unwanted columns which are: priceflag, currency, usdprice.

C. Data Visualization: Visualize the data is an important step that should be done before and after cleaning the data set because it can help us in understanding the features and the correlation between them , also we can identify the noise and outliers for each value when we visualize it, there are three types of visualization I used in my project:

i. Univariate: the simplest form of analyzing data, it requires to analyze each variable separately and it doesn't answer project question about relationships between variables, but used to describe the characteristic of each feature, we can do univariate analysis with or without target like the follow graph:

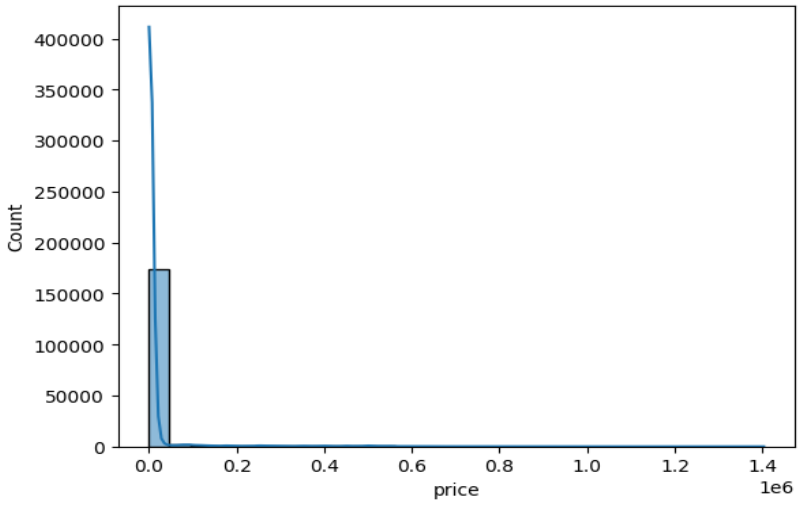


Figure 1 histplot for price column

1e6 means that the price is too large to be write in decimal form, it seems that something wrong with the values, we will see how to resolve this in data preprocessing step.

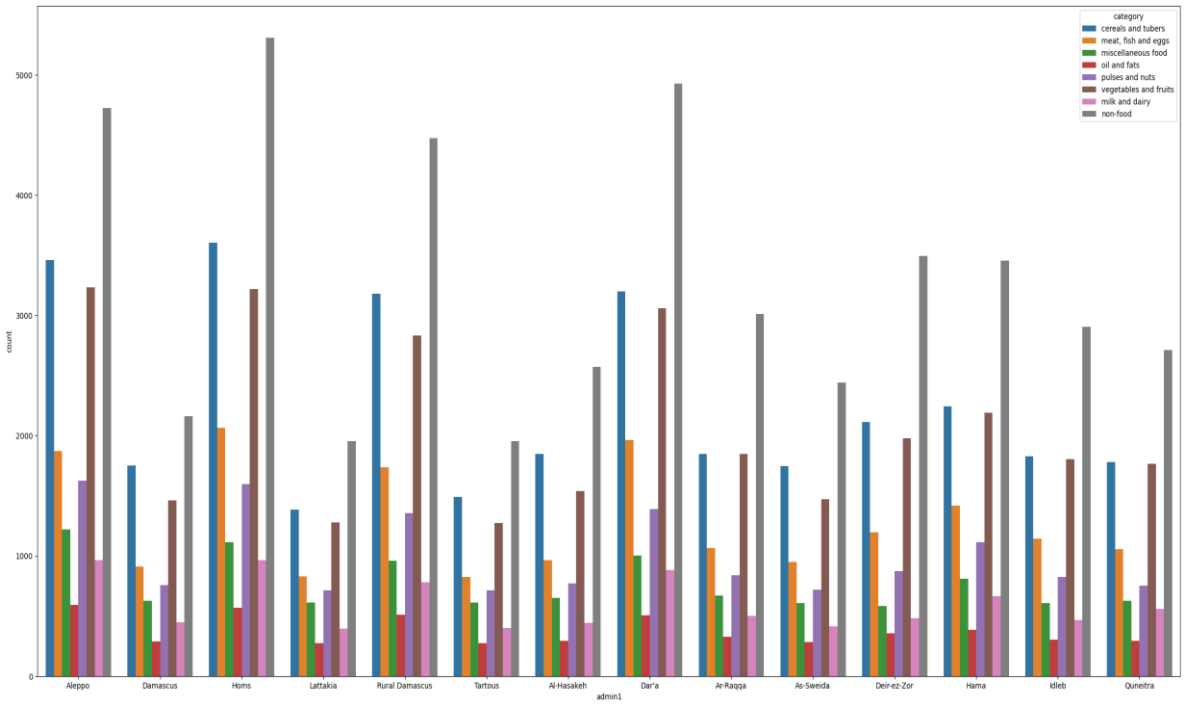


Figure 2: countplot shows univariate analysis for each category with target (price)

Here are a table with the results from figure2:

	Category	Admin1
Top count	non-food	Homs
	cereals and tubers	Homs
	miscellaneous food	Aleppo
	oil and fats	Aleppo
	pulses and nuts	Aleppo
	vegetables and fruits	Aleppo
	milk and dairy	Aleppo
	meat, fish and eggs	Homs

ii. Bi-Variate analysis: this type of analysis involved two variables and the purpose is to find the relation between them, Ex: boxplot, pivot table.

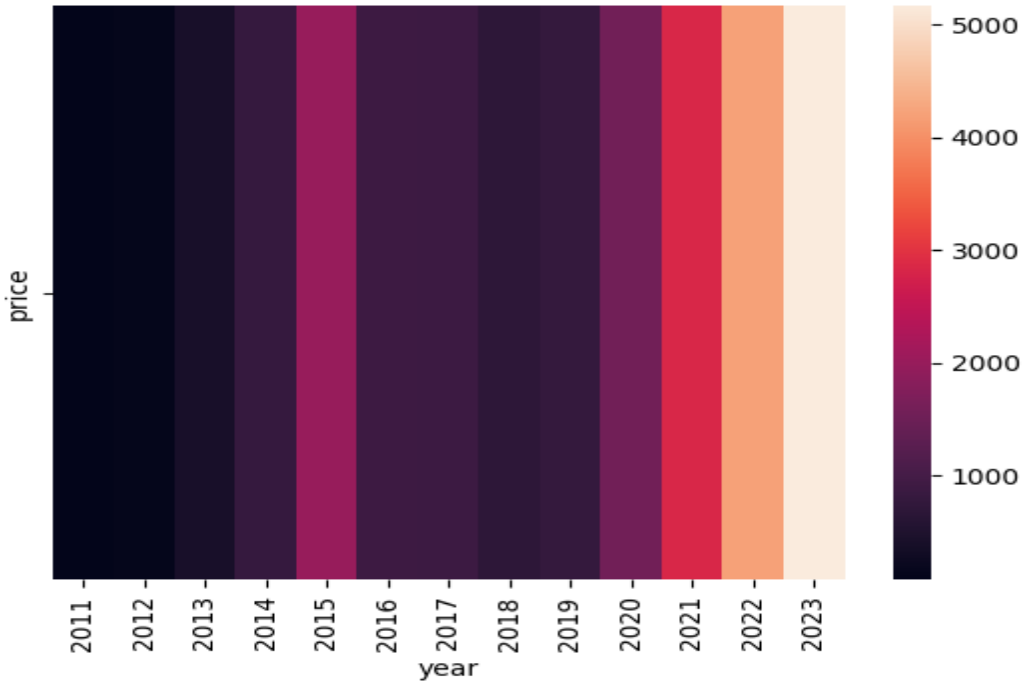


Figure 3 pivot table between price and year

iii. Multivariate Analysis: we have more than two variable, like pair plot and heat map.

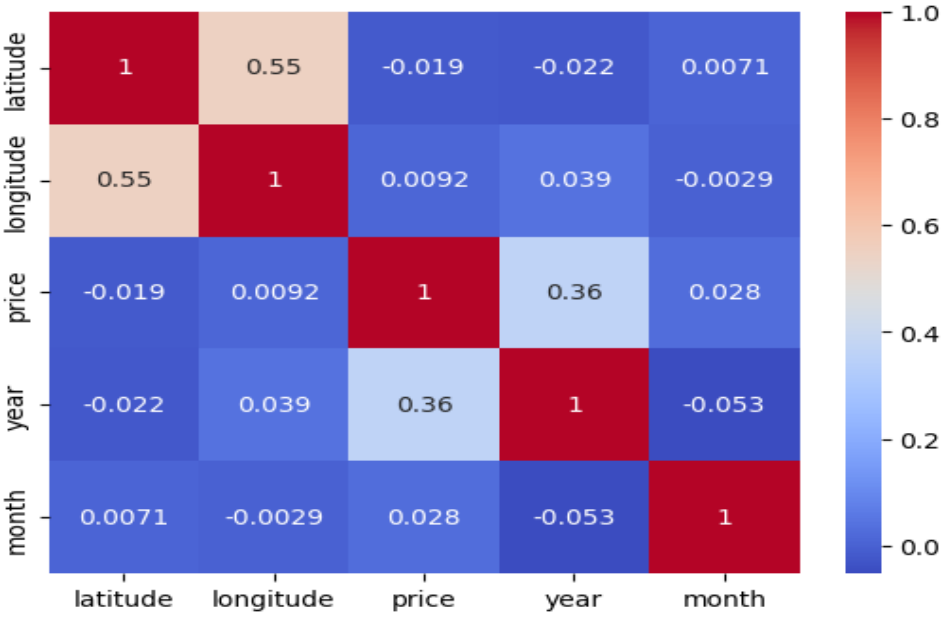


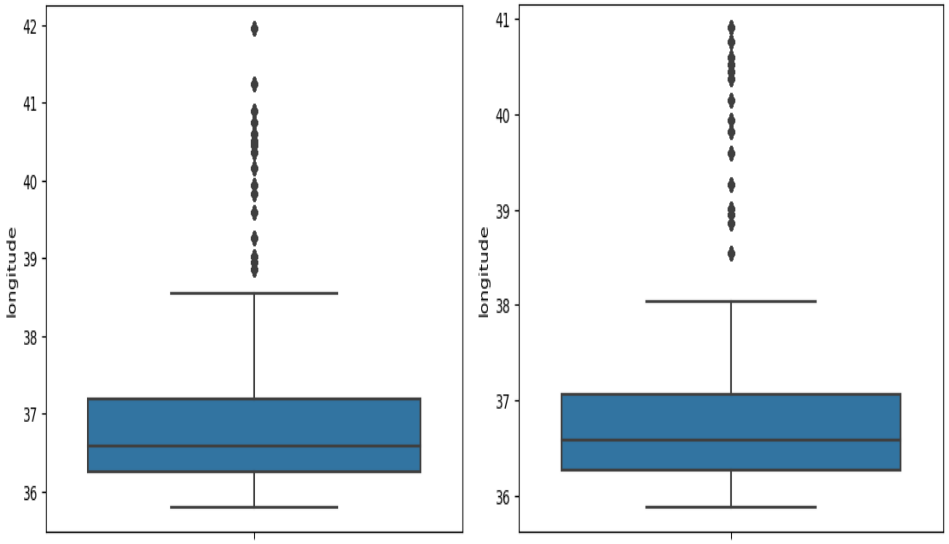
Figure 4 heatmap shows the correlation between the target and features

Data Preprocessing

This step is the most crucial in any machine learning project, I performed it using the following steps:

a. Data cleaning: I did it at first and finish it now.
b. Data transformation.
c. Data Reduction.

a. Data cleaning: the reason for why I mention this step after data visualization is that I want to spot the light about the important of visualization in preprocessing the features and gain a sight from it. Data visualization step is more than nice colored figures but we can also use it as a detective tool that help at complement cleaning the data like find out the outliers and noisy values as the following:



The boxplot shows coordinate like 41 and 42, but we know that the longitude coordinate of Syria is from 35.79011 to 40.91854, so we will drop this extreme value.

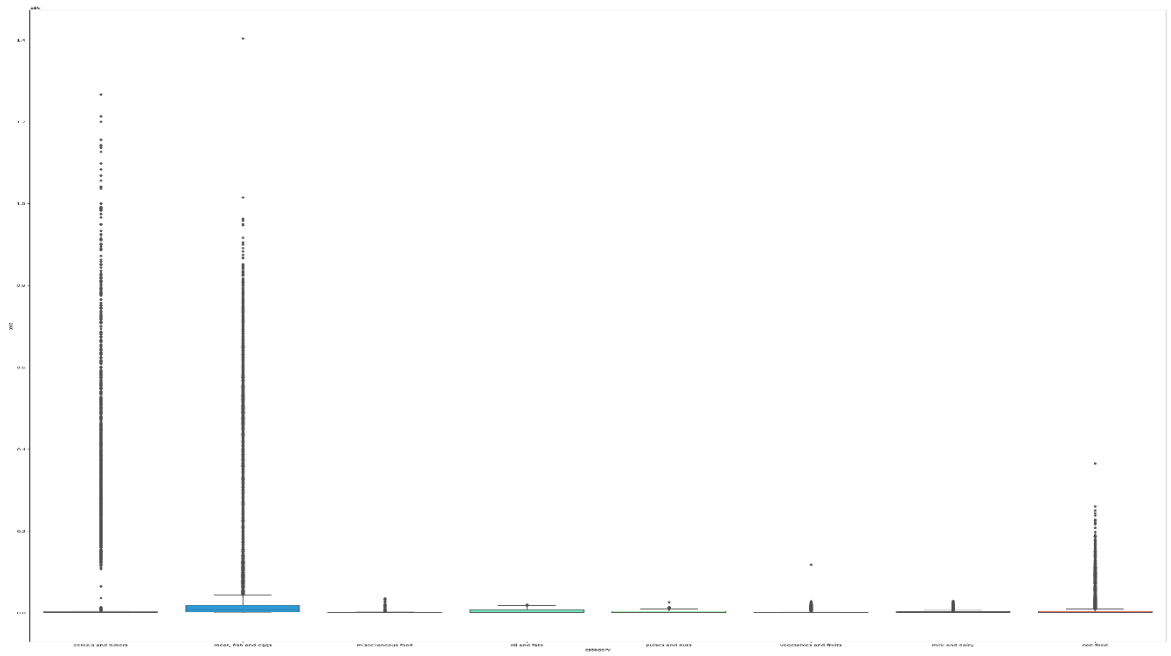


Figure 5: boxplot fro each category with target

Reviewing the figure above we notice that there are something wrong in the price, the values are not acceptable like 1.0, 0.0, 0.4, 0.6. also the median for each category isn't clear and the maximum is 1e6, the dots told us that there are too many outliers. To resolve the problem I used 1.5 IQR method that the boxplot identify the extreme values based on it, the IQR Formula is:

$$IQR = Q3 - Q1$$

Then we identify extreme values (outliers) using those two formula:

Anything above Q3 + 1.5 x IQR is an outlier.

Anything below Q1 - 1.5 x IQR is an outlier.

Where: Q1,Q2, Q3,Quartiles divide a data set into four groups, each containing about 25% (or a quarter) of the data points.

Q1 (the first quartile or lower quartile) is the 25th percentile of the data.

Q2 (the second quartile) is the 50th percentile or median of the data.

Q3 (the third or upper quartile) is the 75th percentile of data. The Interquartile Range (IQR) is the distance between the first and third quartile. Subtract the first quartile from the third quartile to find the interquartile range.

To remove the outliers at price column I have two choices:

a. Apply IQR method on the price column totally.
b. Apply IQR method on the price for each category alone then contact the whole prices in one column.

I implement the second choice for two reasons, the first one is that each category has its own price and the second there is no equal samples of all category, for example the non-food category has 51631 samples and this is the high one in our data set but oil and fats has 5279 samples, so when we call the IQR method for price totally the IQR will affect by values of non-food and maybe it will define outliers based on it, another example is that 5000 maybe is an extreme value for cereals and tubers category but in the other hand it could be a normal value for oil and fats one.

Here are a result after removing outliers:

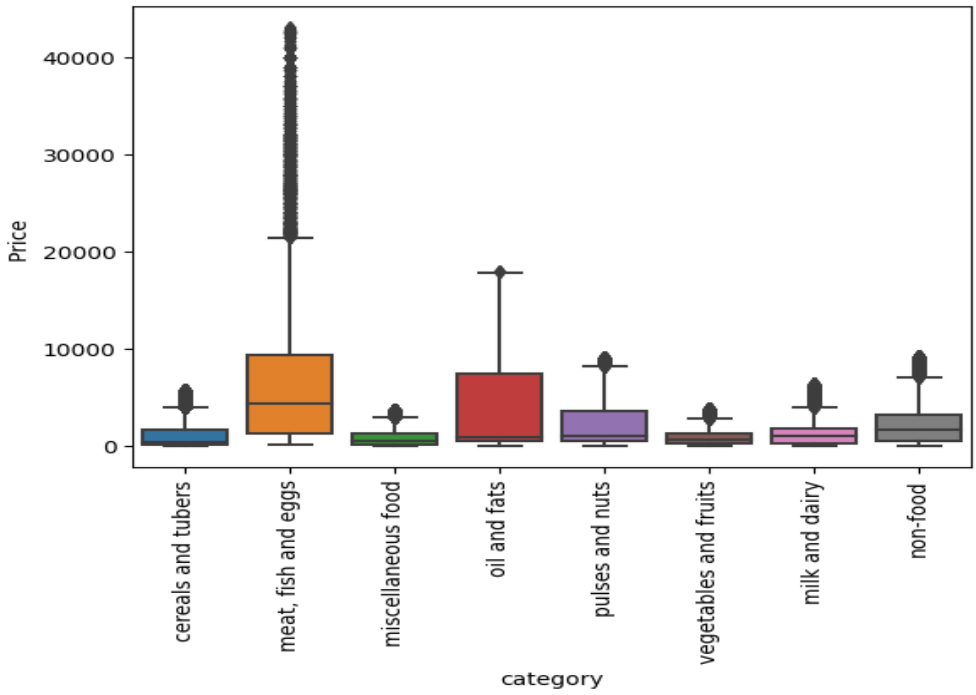


Figure 6 boxplot for category after apply IQR rule

For those remaining outliers I will replace their values used Quantile_baesd Flooring and Capping technique (the outlier is capped at a certain value above the 90th percentile & floored at a factor below the 10th percentile value).

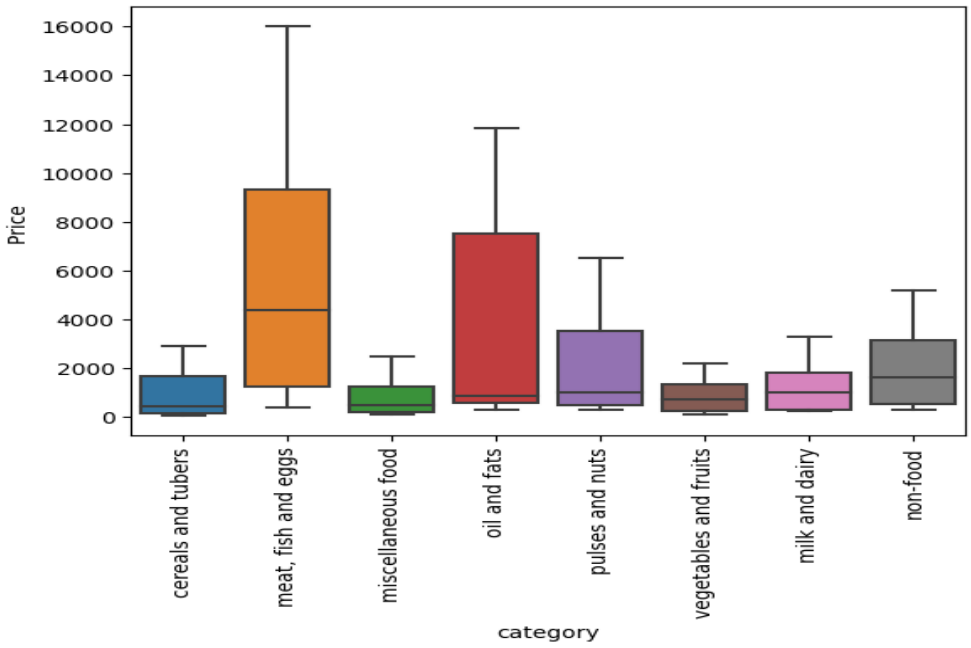


Figure 7 boxplot after using flooring and capping technique

To improve the quality of the data I split the date column into month and year and I do one hot encoding for categorical features so they can be provided to ML algorithms to do a better performance at prediction.

b. Data Transformation:

The data set have different columns with different unit, for example: year column ranges from 2011 to 2023, on the other hand latitude column ranges from 32.61889 to 37.07279, so to make sure that each model will treat this variables equally we need feature scaling, in my studying I used Robust Scalar which seeks to mitigate the effects of any remaining outliers by ignore them from the calculation of the mean and standard deviation, this scalar is removes the median ans scales the data according to the quantile range (IQR), its formula is as follow:

$$\frac{\boldsymbol{x - Q1(x)}}{\boldsymbol{Q3(x) - Q1(x)}}$$

to avoid data leakage I fit and transform the robust scalar to the training set only, for test set just transforming.

c. Data Reduction:

To reduce the complexity of the model and improve the performance of a learning algorithm I used dimension reduction techniques which help in reducing the number of features (or dimensions) while retaining as much information as possible, the two main approaches I used:

i. Feature Selection:

selecting a subset of the original features that are most relevant to the problem, in my project I used

SelectKBest method which return the features accodring to the k highest score.

ii. Feature Extraction:

I used Principal Component Analysis (PCA) to reduce the data in a high dimensional space to a lower dimension one, at first and for help in preserving as much as possible of varaiability in the data

I passed 0.95 as n_component to pca which describes the cut off point (the desired variance that is supposed to be explained by the principal components) then I drew the plotted chart of cumulative variance to select the right number of principle component which shows the total percentage of variance vs. the number of principle components.

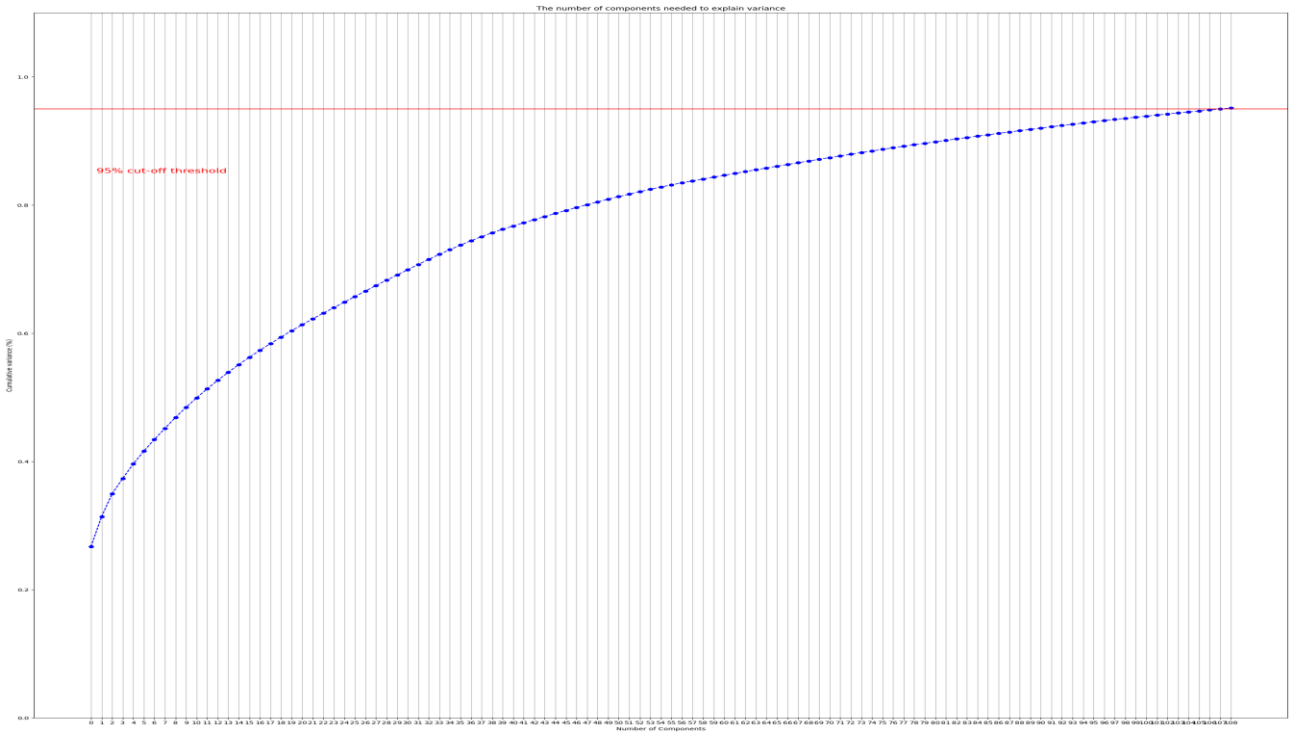


Figure 8: pca component of 95% varaiance

The figure above shows that I need 108 principal components to get 95% of variance explained.

Model Development

After splitting data into train & test set, I applied ML models

1. Linear Regression.
2. Decision Tree Regressor: with hyperparameter tuning using GridSearchCV.
3. Random Forest Regressor: with hyperparameter tuning using GridSearchCV.
4. GradientBoostingRegressor.

Model Evalutaion

Evalute the models with the best parameter using r2_score and accuracy measure.

Model	Hyper parameter Tunning	Range	Best value	r2-score	Training accruracy	Testing acuuracy
Linear Rregression	Tunning is not available for this model	none	none	0.62	0.622	0.622
Decision Tree regressor	Max_depth min_samples _leaf min_samples _split	2 to 15 1 to 6 2 to 6	14 2 2	0.92	0.946	0.923
Random forest regressor	Max_depth With n_estimators equal to 10	1 to 15	14	0.95	0.961	0.945
Gradient Boosting Regressor	n_estimators	1 to 100	99	0.85	0.855	0.854

Reference:

- https://www.pluralsight.com/guides/cleaning-up-data-from-outliers.
- https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiz9OiO5ImBAxWuZ_EDHcl4BnIQFnoECA8QAw&url=https%3A%2F%2Ftowardsdatascience.com%2Fpractical-implementation-of-outlier-detection-in-python-90680453b3ce&usg=AOvVaw3lRd5Y5htEGco0QfjqT663&opi=89978449.
- https://machinelearningmastery.com/data-preparation-without-data-leakage/.
- https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/.
- https://mikulskibartosz.name/pca-how-to-choose-the-number-of-components/.
- https://machinelearningmastery.com/robust-scaler-transforms-for-machine-learning/.
- https://machinelearningmastery.com/calculate-feature-importance-with-python/.