Walmart Sales Forecasting

Team: Tech Divas

- Maria Mathew
- Rekha Yadav

Under the guidance of
Prof. Ramkumar Hariharan

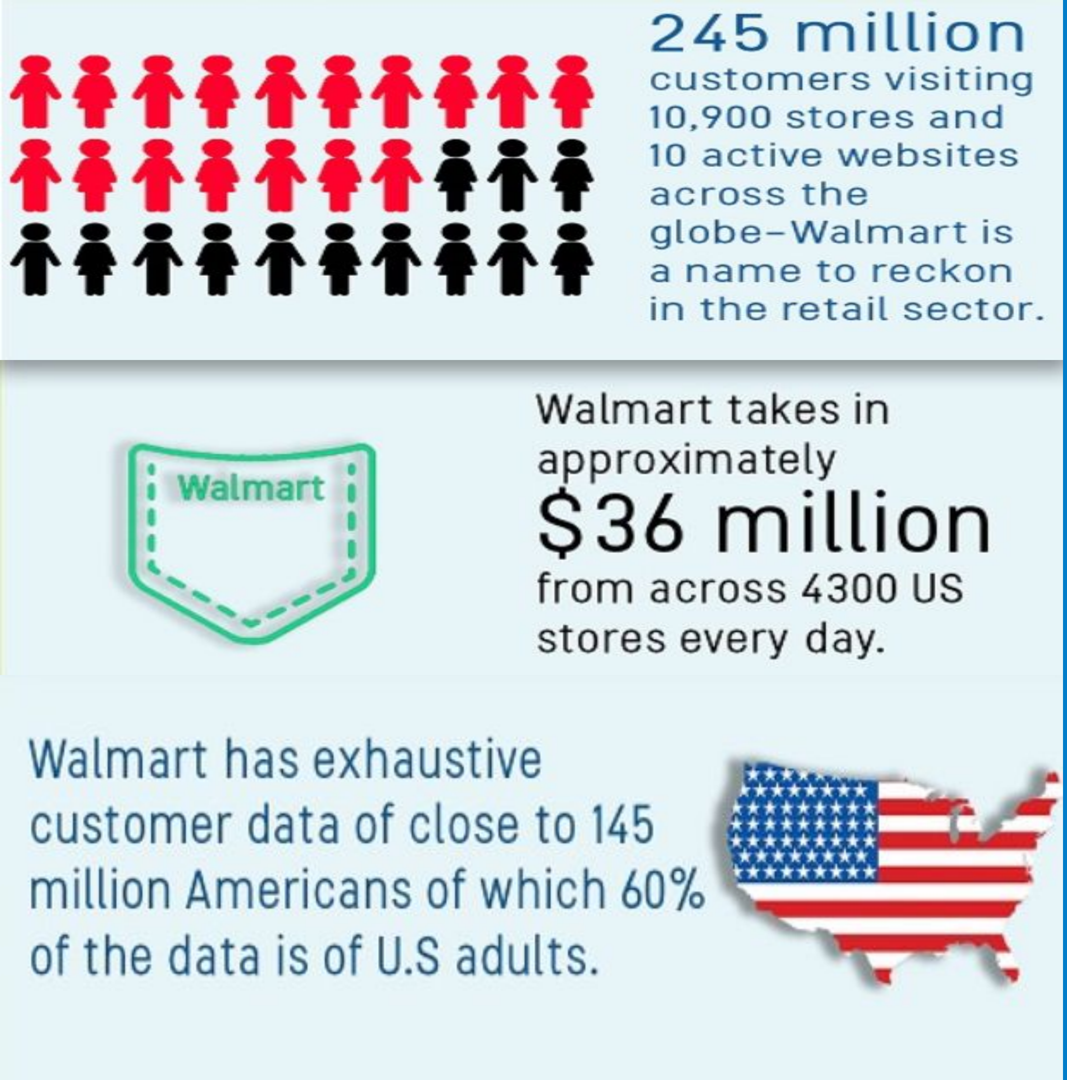
Table of contents

- Problem Statement
- Background and Motivation
- Business Case
- Data Set
- Data Flow
- Libraries
- Data Preprocessing
- Feature Importance
- Evaluation Metrics
- Machine Learning Algorithms
- Deep Learning
- Model Performance

PROBLEM STATEMENT

Predict Walmart Weekly Sales for the given department in the given store

BACKGROUND & MOTIVATION



245 million customers visiting 10,900 stores and 10 active websites across the globe–Walmart is a name to reckon in the retail sector.

Walmart takes in approximately $36 million from across 4300 US stores every day.

Walmart has exhaustive customer data of close to 145 million Americans of which 60% of the data is of U.S adults.

BUSINESS CASE

Accurately forecast sales of Walmart as it is key for its ability to function

DATA SET

- Kaggle competition : Walmart Recruiting - Store Sales Forecasting

  https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting

- Contains real-world historical sales data of 45 Walmart stores located in different regions dated from 2010-02-05 to 2012-11-01.

- Each row represents a record that comes from a specific walmart store, department and date combination.

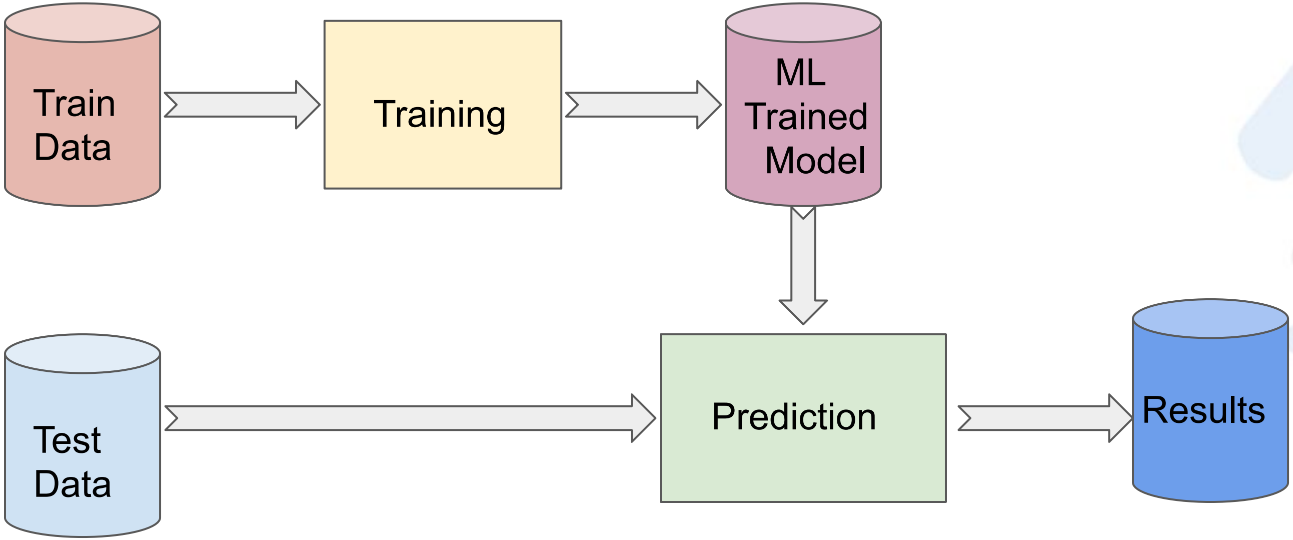
INPUT FEATURES

- **Store** - The store number
- **Dept** - The department number
- **Date** - The week
- **IsHoliday** - Whether the week is a special holiday week
- **Type** - Store type
- **Size** - Store size
- **Temperature** - Average temperature in the region
- **Fuel_Price** - Cost of fuel in the region
- **MarkDown1-5** - Anonymized data related to promotional markdowns that Walmart is running. MarkDown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA.
- **CPI** - The consumer price index
- **Unemployment** - The unemployment rate

TARGET

- **Weekly_Sales** - sales for the given department in the given store

DATA FLOW


Train Data
Training
ML Trained Model
Test Data
Prediction
Results

LIBRARIES

matplotlib

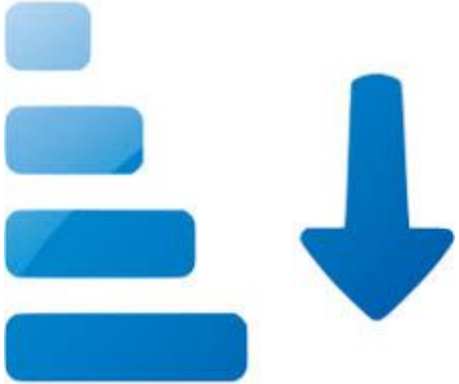
NumPy

scikit
learn

pandas
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

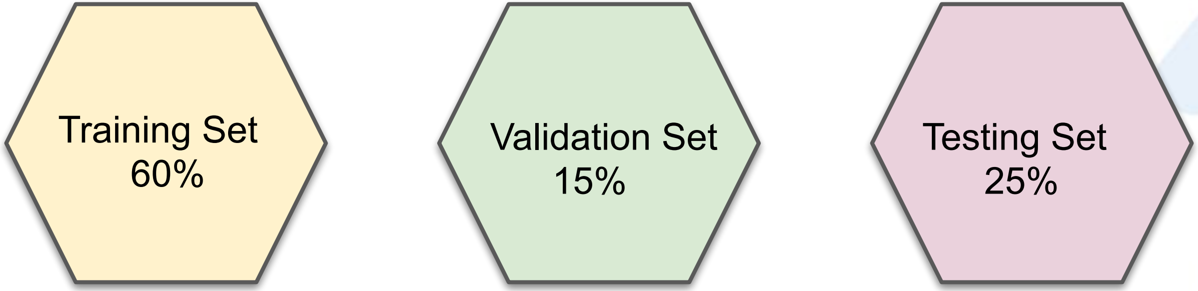
DATA PROCESSING

DATA PREPROCESSING



- Sorting data based on dates in ascending order

DATA PREPROCESSING



Training Set
60%

Validation Set
15%

Testing Set
25%

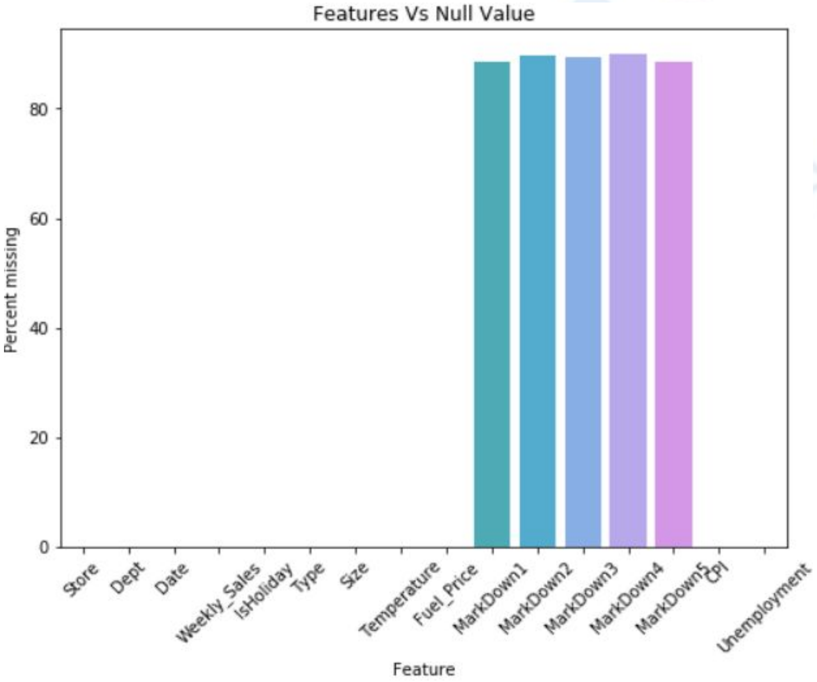
To train the model

To make sure the models are not overfitting

To determine the accuracy of the model

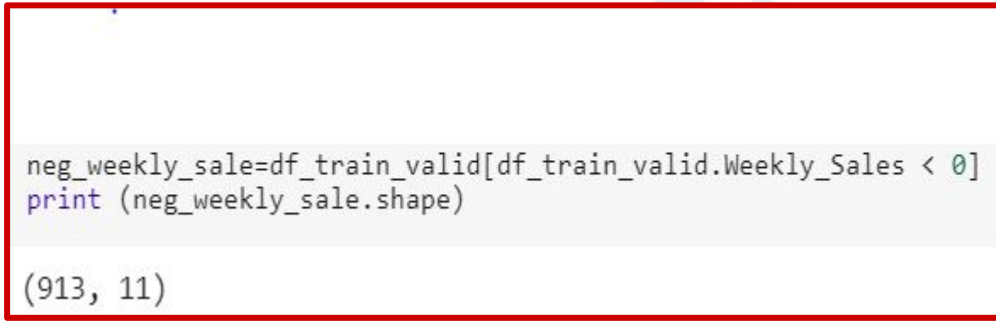
DATA PREPROCESSING

- Dropping column Markdown1, Markdown2, Markdown3, Markdown4 and Markdown5 as more than 80% of data is Null.



DATA PREPROCESSING

- Dropping records with negative weekly sale

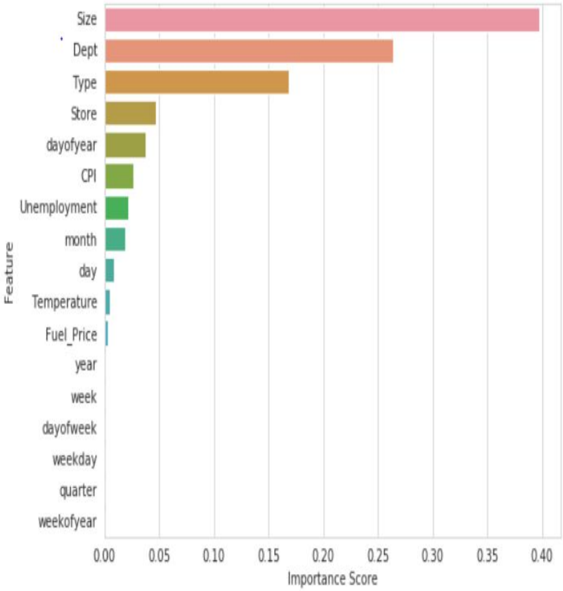


```
neg_weekly_sale=df_train_valid[df_train_valid.Weekly_Sales < 0]
print (neg_weekly_sale.shape)
```

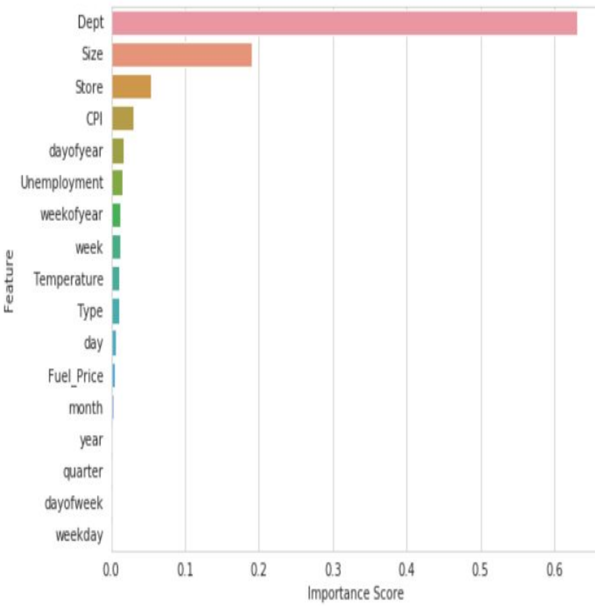
```
(913, 11)
```

Feature Importance

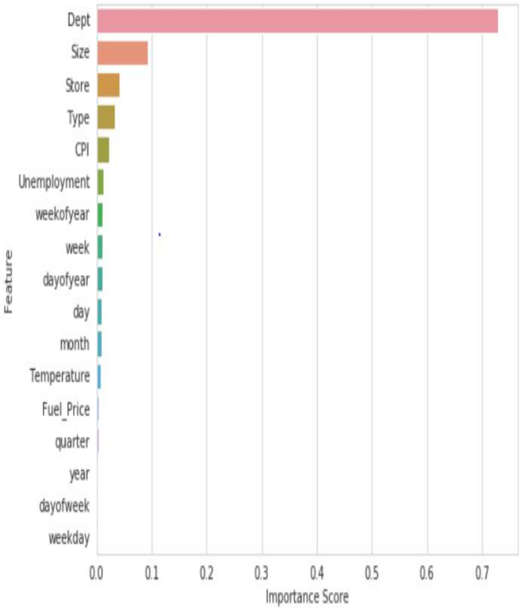
XGBoost



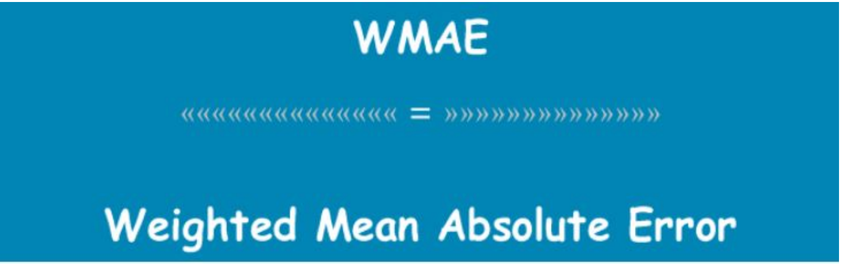
Random Forest



Extra Trees



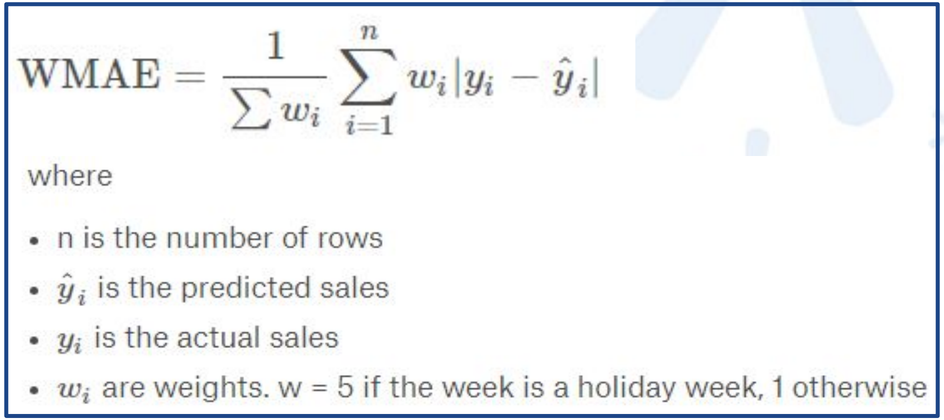
Evaluation Metrics:
Weighted Mean Absolute Error



WMAE

««««««««««««««« = »»»»»»»»»»»»»»»

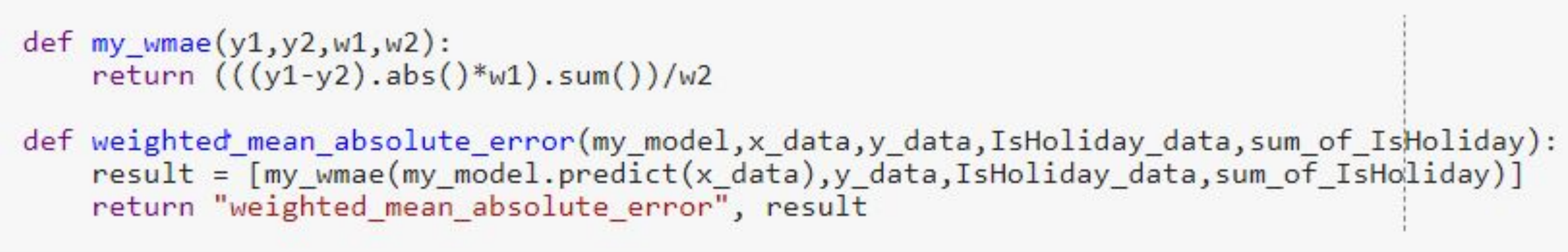
Weighted Mean Absolute Error



$$\text{WMAE} = \frac{1}{\sum w_i} \sum_{i=1}^{n} w_i |y_i - \hat{y}_i|$$

where

- n is the number of rows
- $\hat{y}_i$ is the predicted sales
- y_i is the actual sales
- w_i are weights. w = 5 if the week is a holiday week, 1 otherwise



```
def my_wmae(y1,y2,w1,w2):
    return (((y1-y2).abs()*w1).sum())/w2

def weighted_mean_absolute_error(my_model,x_data,y_data,IsHoliday_data,sum_of_IsHoliday):
    result = [my_wmae(my_model.predict(x_data),y_data,IsHoliday_data,sum_of_IsHoliday)]
    return "weighted_mean_absolute_error", result
```

Machine Learning Algorithms

- For a given **K** and a prediction point **xo**, KNN regression first identifies the **K** training observations that are close to **xo**, represented by **No**.

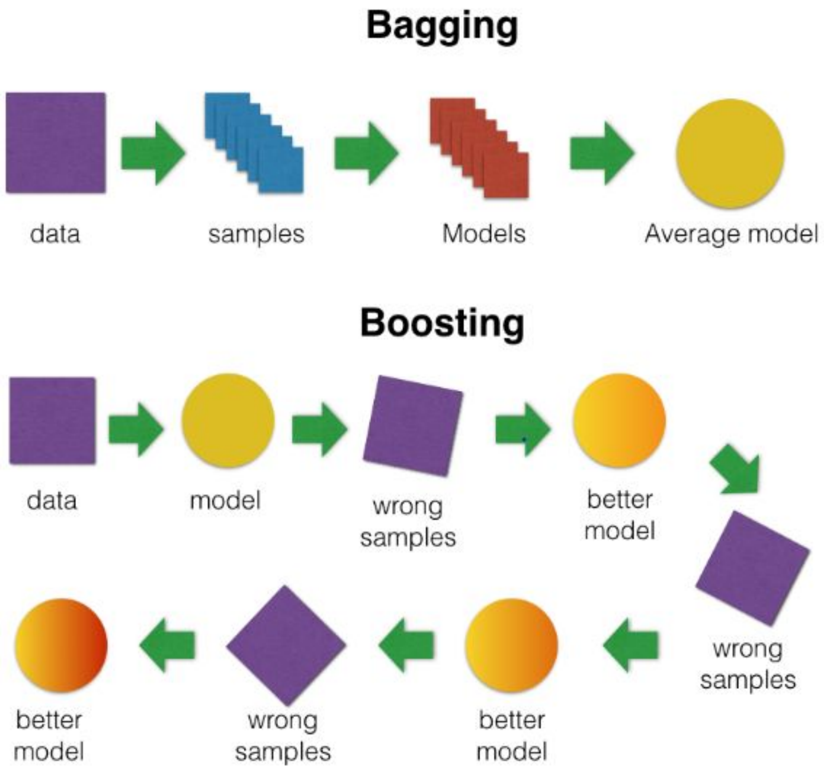
- Estimates **f(xo)** using the average of all the training responses in **No.**

Equation of KNN regression:

$$\hat{f}\cdot(x_0) = \frac{1}{K} \sum_{x_i \in N_0} y_i$$

k-nearest neighbors regression

Machine Learning Algorithms



Extra Trees Regression

Random Forest Regression

XGBoost Regression

Bagging:

- Handles overfitting
- Reduce variance

Eg: Random Forest Regression, Extra Trees Regression.

Boosting:

- Can lead to overfitting
- Reduce bias and variance

Eg: Gradient Boosting

XGBoost is an implementation of Gradient Boosting Machine.

Machine Learning Algorithms

- Multiple independent variables(input features) contributing to the dependent variable(Target feature)

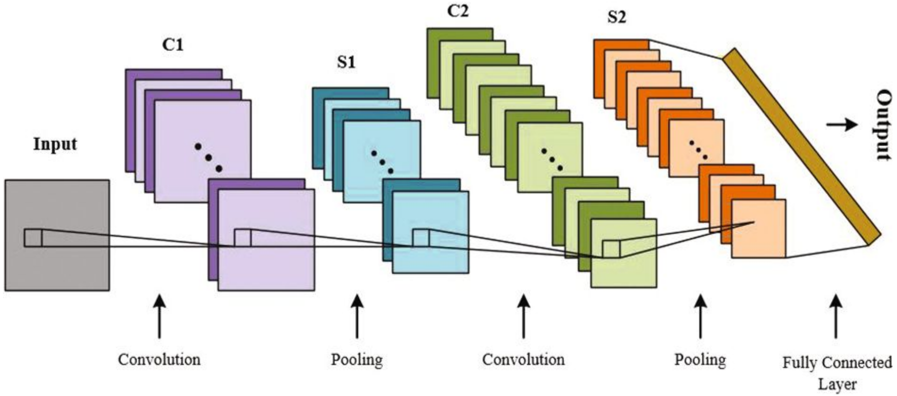
- Similar to Linear Regression

Multivariate linear

Regression

Deep Learning

CNN Architecture



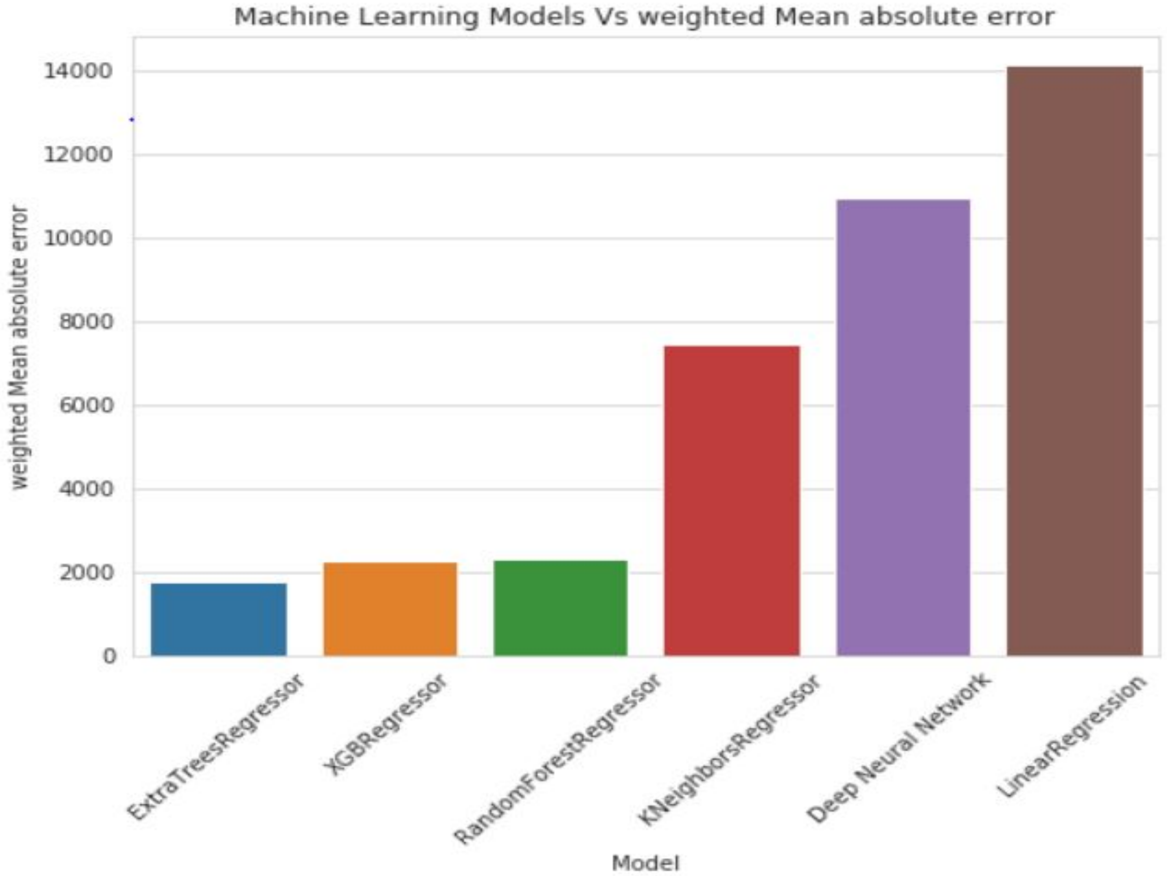
Deep Neural Networks

- Outperform traditional machine learning algorithms if the data size is large.

- Deep Learning algorithms try to learn high-level features from data in an incremental manner.

- Eliminates the need of domain expertise and hard core feature extraction.

Model Performance



Thank you