edWisor

Cab Fare Prediction

Akshay Rahate

https://github.com/erepr/Taxi-Fare-Prediction/

Index

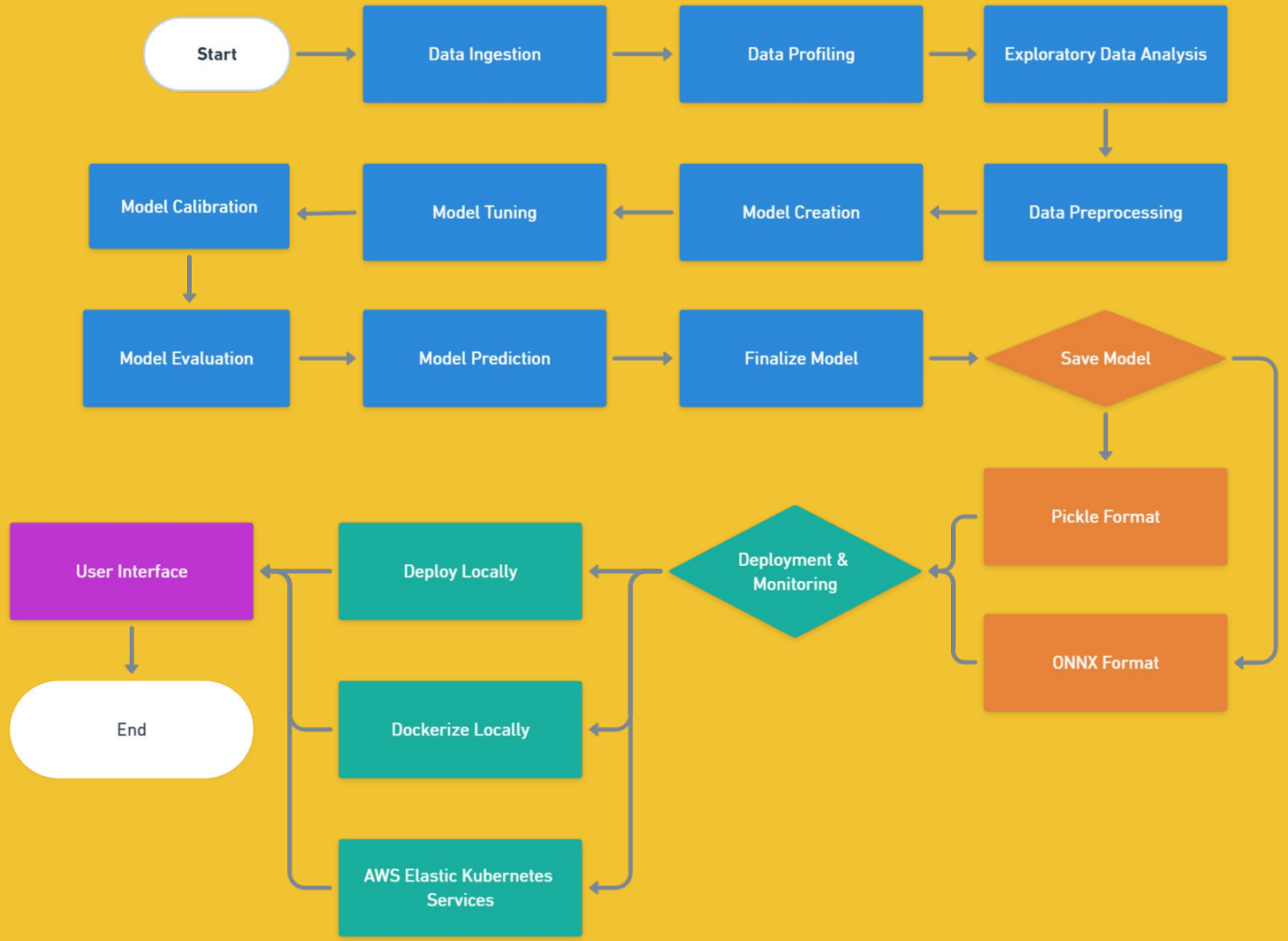
Problem Statement

You are a cab rental start-up company. You have successfully run the pilot project and now want to launch your cab service across the country. You have collected the historical data from your pilot project and now have a requirement to apply analytics for fare prediction. You need to design a system that predicts the fare amount for a cab ride in the city.

Attributes Datatypes

- pickup_datetime - object
- fare_amount - object
- pickup_longitude - float64
- pickup_latitude - float64
- dropoff_longitude - float64
- dropoff_latitude - float64
- passenger_count - float64

Architecture



Preliminary Investigation

Dependent Features:- fare_amont, which exists only in train dataset.
Independent Features:- pickup_datetime, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude, passenger_count. And these features are common for both data sets.
fare_amount - convert outlier to NaN and impute

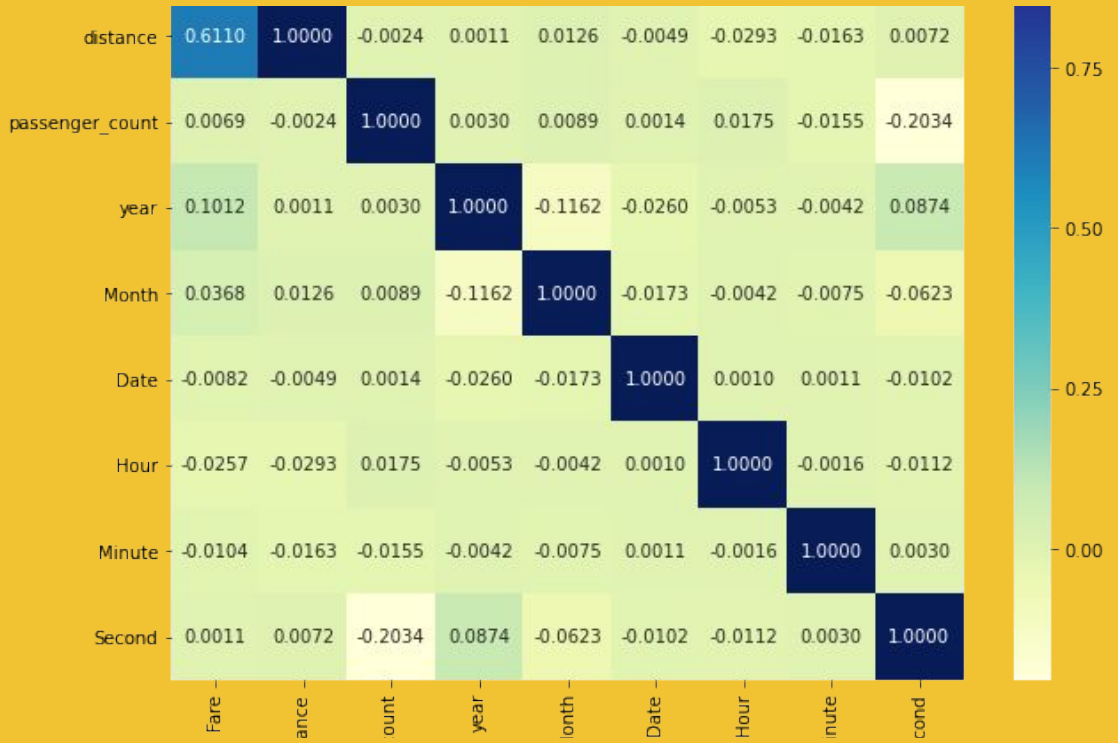
- Extreme large value
- Negative value
- Extreme small values
- Zero ‘0’

pickup_longitude, **pickup_latitude**, **dropoff_longitude**, **dropoff_latitude** - [-0.004093, 0.0335, 401.083332, 0, 0.016852, -7.9866399999999995, 0.01798, 0.728087] convert outlier to NaN and impute
passenger_count - limit count to 6

Splitting datetime

pickup_datetime	year	Month	Date	Hour	Minute	Second
2009-06-15 17:26:21 UTC	2009	6	15	17	26	21
2010-01-05 16:52:16 UTC	2010	1	5	16	52	16
2011-08-18 00:35:00 UTC	2011	8	18	0	35	0
2012-04-21 04:30:42 UTC	2012	4	21	4	30	42
2010-03-09 07:51:00 UTC	2010	3	9	7	51	0
...	...	...	...	...	...	...
2014-12-12 07:41:00 UTC	2014	12	12	7	41	0
2009-07-13 07:58:00 UTC	2009	7	13	7	58	0
2009-11-11 11:19:07 UTC	2009	11	11	11	19	7
2010-05-11 23:53:00 UTC	2010	5	11	23	53	0
2011-12-14 06:24:33 UTC	2011	12	14	6	24	33

Checking Correlation



Pearson's Correlation Test

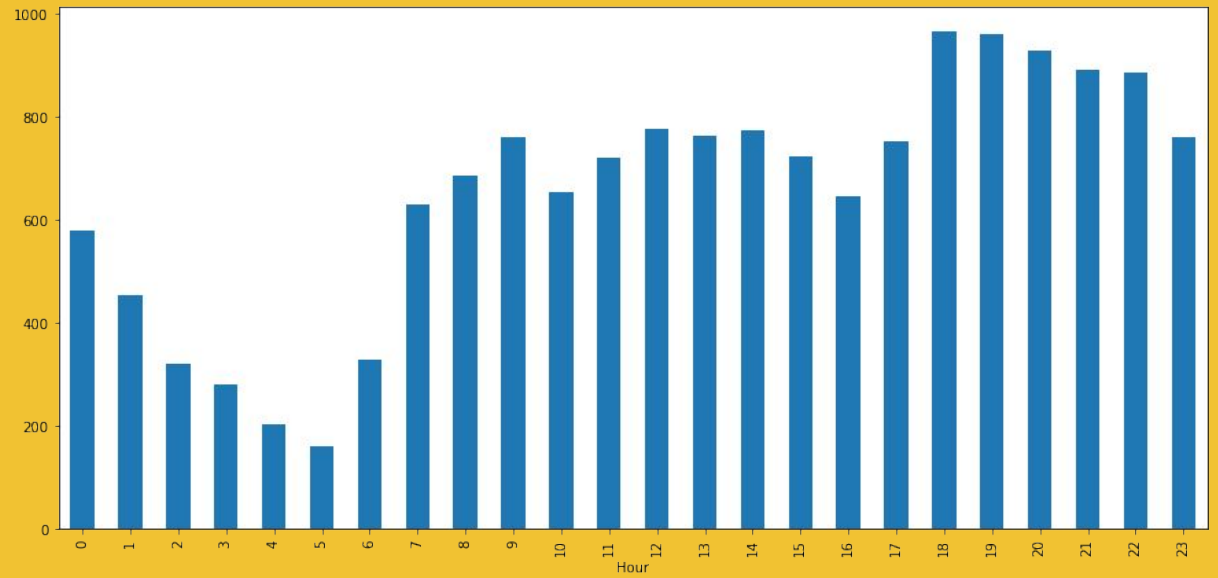
	Fare	distance	passenger_count	year	Month	Date	Hour	Minute	Second
Fare	1	0.611	0.0069	0.1012	0.0368	-0.0082	-0.0257	-0.0104	0.0011
distance	0.611	1	-0.0024	0.0011	0.0126	-0.0049	-0.0293	-0.0163	0.0072
passenger_count	0.0069	-0.0024	1	0.003	0.0089	0.0014	0.0175	-0.0155	-0.2034
year	0.1012	0.0011	0.003	1	-0.1162	-0.026	-0.0053	-0.0042	0.0874
Month	0.0368	0.0126	0.0089	-0.1162	1	-0.0173	-0.0042	-0.0075	-0.0623
Date	-0.0082	-0.0049	0.0014	-0.026	-0.0173	1	0.001	0.0011	-0.0102
Hour	-0.0257	-0.0293	0.0175	-0.0053	-0.0042	0.001	1	-0.0016	-0.0112
Minute	-0.0104	-0.0163	-0.0155	-0.0042	-0.0075	0.0011	-0.0016	1	0.003
Second	0.0011	0.0072	-0.2034	0.0874	-0.0623	-0.0102	-0.0112	0.003	1

Chi-Square Test

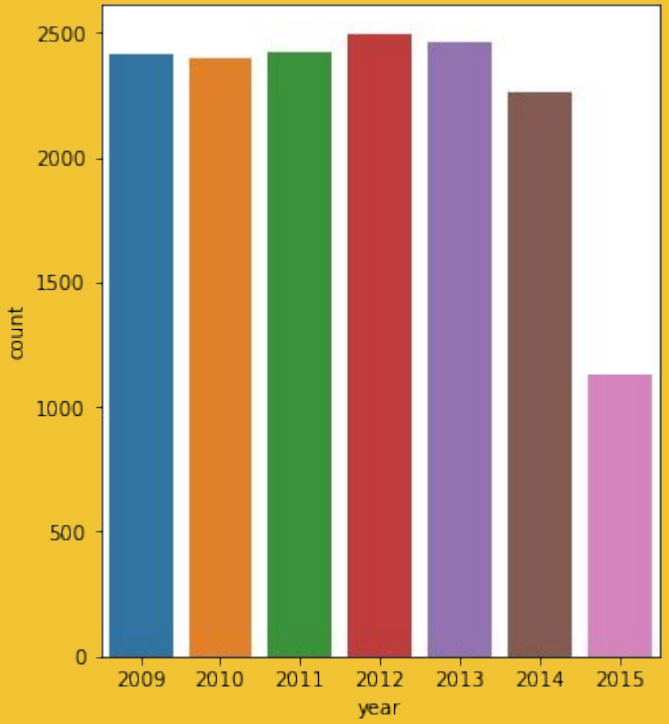
p value is 0.0
Dependent (reject H0)
Since, ***p-value > alpha*** Therefore, we **reject H0**, that is, the variables do have significant relationship.

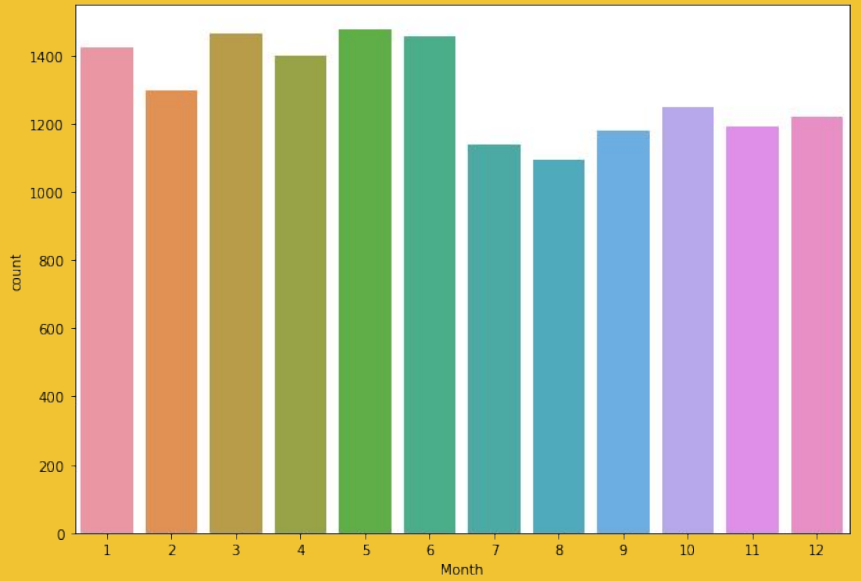
Cab Demand Surge

Demand surge for cabs after
5 AM and 5 PM i.e the office rush hours

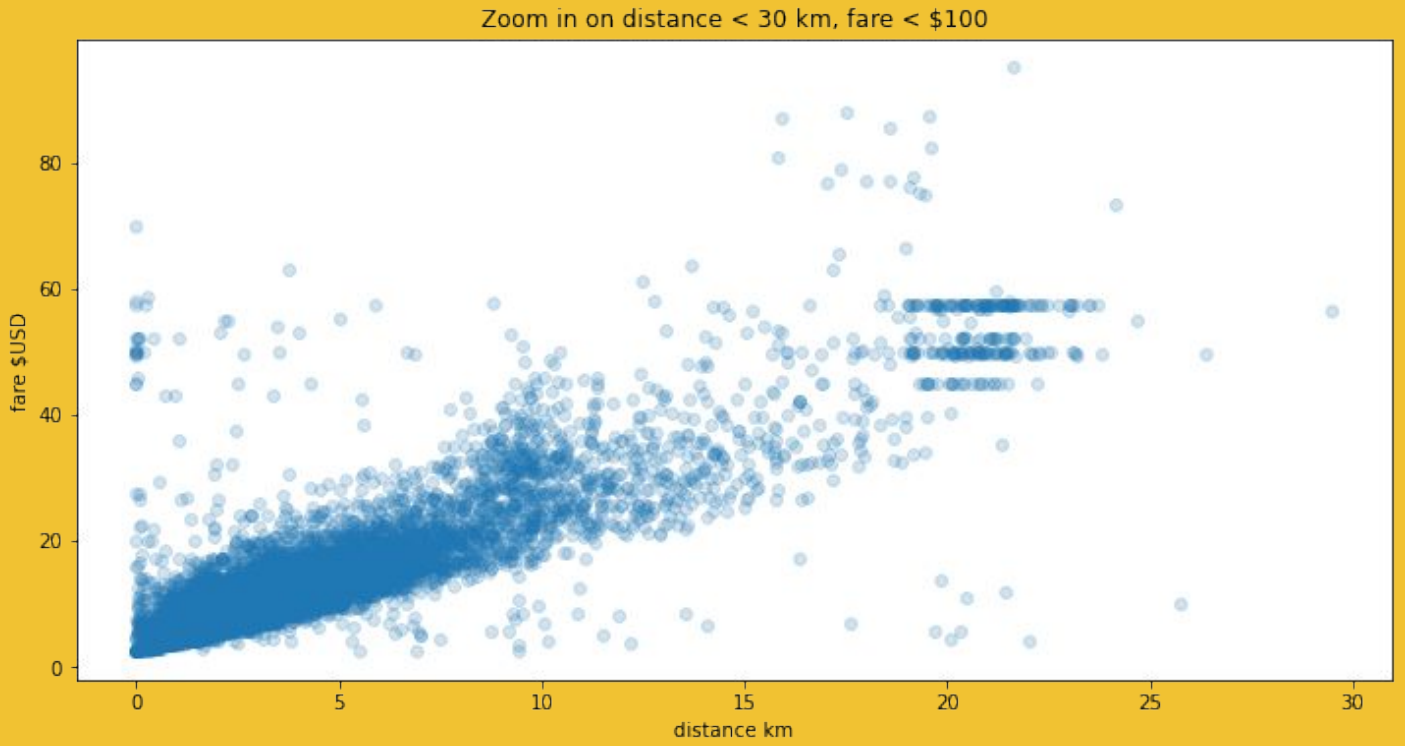


No Demand Surge Yearly & Monthly



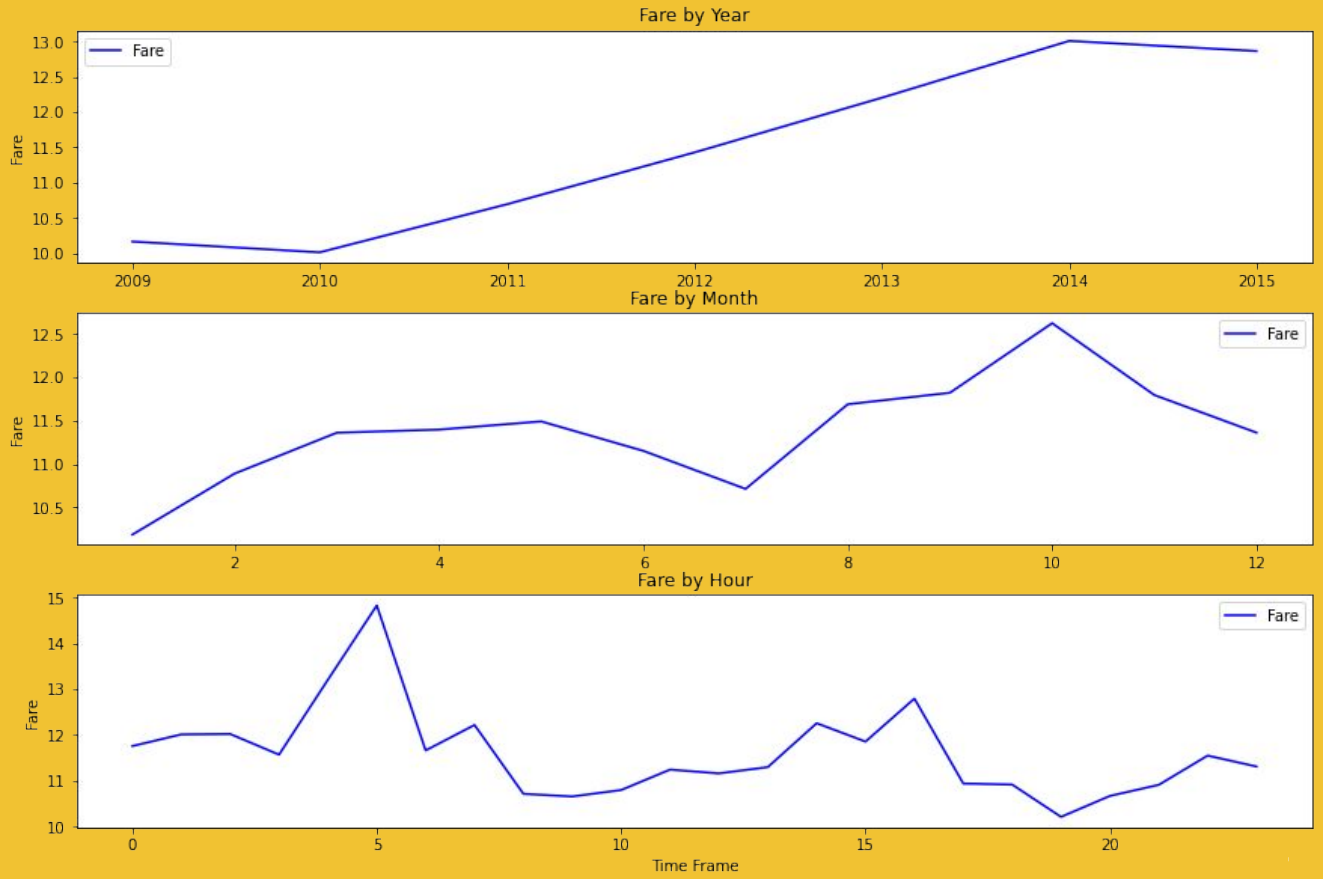


Fare vs Distance



Fare Surcharge

- Surge Fare at 5 AM
- 4-5 PM office goers
- After October / Winter Season / Festivals



Final Dataframe

We drop variables like

- passenger_count,
- year, Date
- Minute
- Second

Because from EDA it does not contribute much to the information neither it explains any seasonality changes

	Fare	distance	Month	Hour
0	4.5	1.030765	6	17
1	16.9	8.450145	1	16
2	5.7	1.389527	8	0
3	7.7	2.799274	4	4
4	5.3	1.999160	3	7
...	...	...	...	...
15580	6.5	0.850046	12	7
15581	16.1	7.867649	7	7
15582	8.5	1.469108	11	11
15583	8.1	2.590040	5	23
15584	8.5	3.898118	12	6

15585 rows × 4 columns

Regression Module in PyCaret

Defining parameters

- Splitting dataset into train-test dataset with 70%-30% ratio.
- Shuffling of rows during train-test split
- Normalization using z-score
- Applying transformation to make data more gaussian-like
- Remove outliers using Singular Value Decomposition technique
- Remove multicollinearity
- Apply transformation on target variable as well
- 10 kfold cross validation strategy

PyCaret is an open source, low-code machine learning library in Python.

It provides high level abstraction on top of Sci-Kit Learn, Matplotlib, Seaborn etc

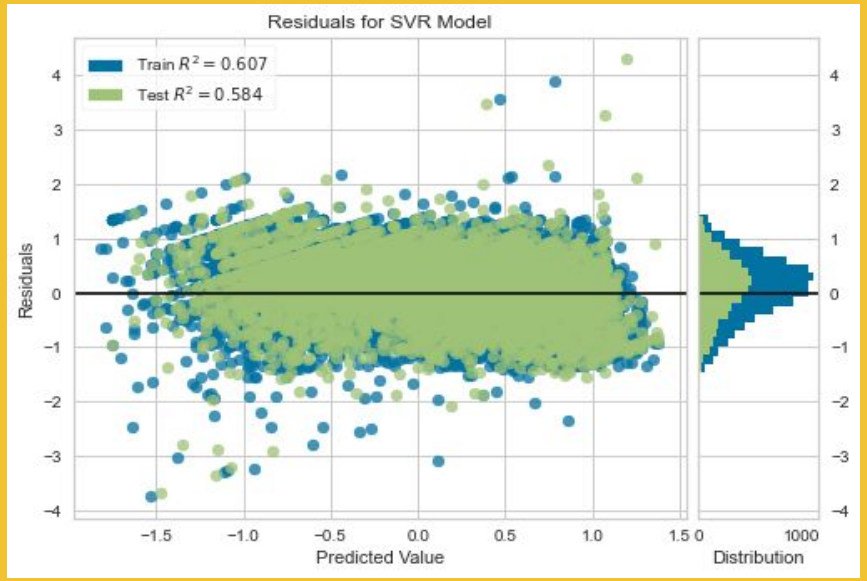
Compare different Models

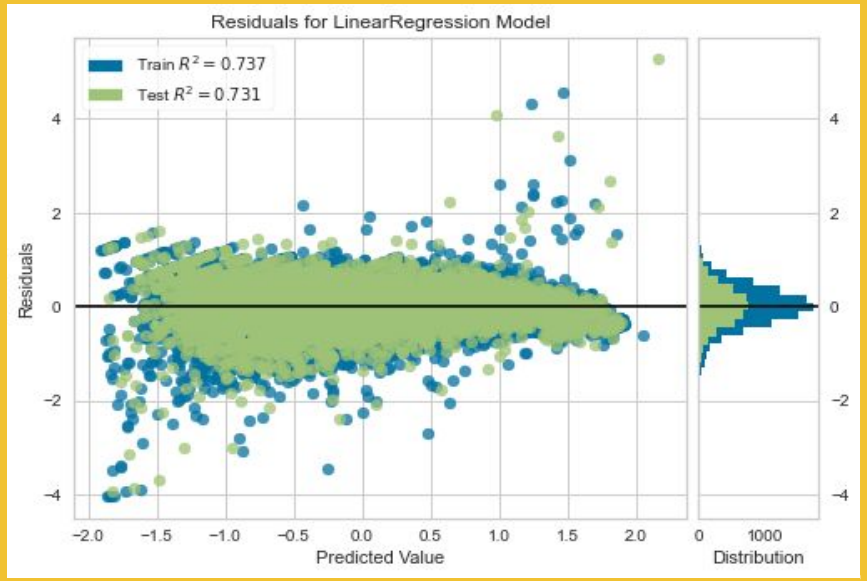
	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
0	Support Vector Machine	2.1191	51.5458	5.9472	0.6419	0.2435	0.1824	8.4923
1	Gradient Boosting Regressor	2.1324	52.0275	5.9978	0.6368	0.2428	0.1859	1.4438
2	Random Forest	2.2731	53.2780	6.1259	0.6206	0.2563	0.2020	1.6923
3	Extra Trees Regressor	2.4134	53.9715	6.2050	0.6093	0.2704	0.2188	1.5564
4	K Neighbors Regressor	2.3388	54.4834	6.2306	0.6062	0.2604	0.2033	0.1382
5	Linear Regression	2.3301	56.3178	6.4392	0.5789	0.2517	0.1917	0.0141
6	Ridge Regression	2.3297	56.3225	6.4391	0.5789	0.2517	0.1917	0.0103
7	AdaBoost Regressor	2.8744	66.8570	7.3425	0.4373	0.2969	0.2327	0.5297
8	Elastic Net	4.1445	96.0414	9.2659	0.0550	0.4366	0.3364	0.0114
9	Lasso Regression	4.7216	105.7366	9.8093	-0.0748	0.5089	0.4072	0.0116
10	Decision Tree	3.1699	105.2629	9.2965	-0.2438	0.3437	0.3080	0.0985

- We select Support Vector Machine, Random Forest and Linear Regression model for further model building.
- Tune hyperparameters.
- Benchmark against metrics like MAE (Mean Absolute Error), MSE (Mean Square Error), RMSE (Root Mean Square Error), R2, RMSLE (Root Mean Squared Log Error) and MAPE (Mean Absolute Percentage Error).

Model Evaluation

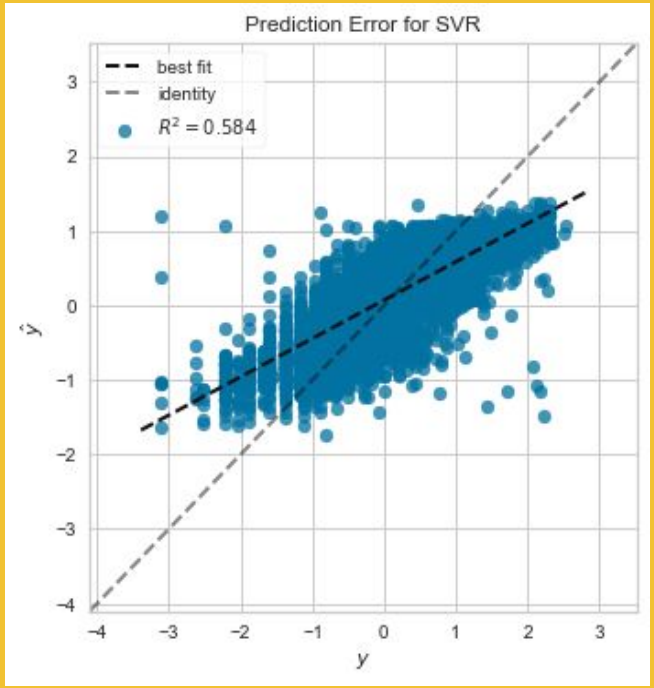
A **Residual Plot** has the Residual Values on the vertical axis; the horizontal axis displays the independent variable. If the points in a residual plot are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data; otherwise, a nonlinear model is more appropriate.

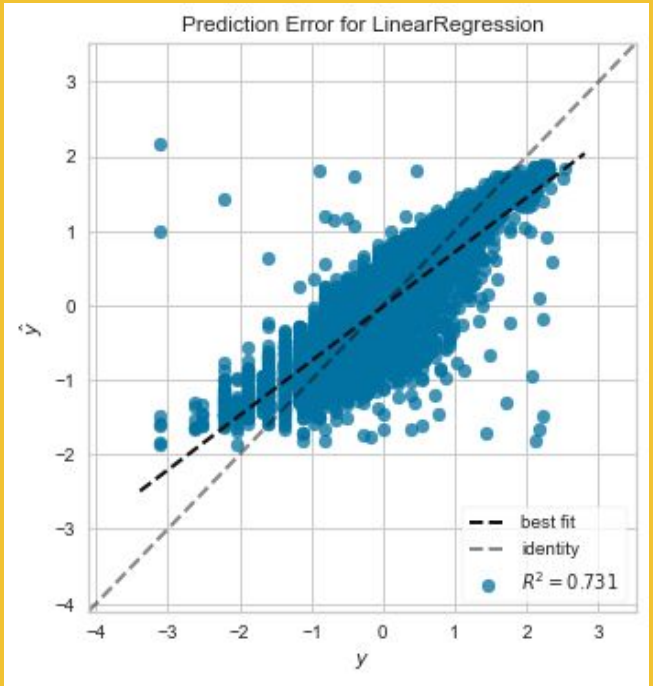




Model Evaluation

A **Prediction Error Plot** shows the actual targets from the dataset against the predicted values generated by our model. This allows us to see how much variance is in the model.





Metrics

Model	R2 Score	MAPE
SVM	0.6413	0.1829
LR	0.5769	0.1919
Tuned SVM	0.2979	0.2696
Tuned LR	0.5789	0.1917
Predict on Test SVM	0.3582	0.2736
Predict on Test LR	0.6884	0.1968
Final SVM	0.3931	0.2670
Final LR	0.6885	0.1966

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. The higher the R-squared, the better the model fits your data.

The MAPE is a statistical measure of how accurate a model is. It can be calculated as the average absolute percent error for each time period minus actual values divided by actual values.

$$\mathrm{M} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|,$$

where A_t is the actual value and F_t is the forecast value.

Save Model

Python pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc.) into byte streams (0s and 1s) is called pickling or serialization or flattening or marshalling. We can converts the byte stream (generated through pickling) back into python objects by a process called as unpickling.

Thank You!