CUSTOMER LIFE TIME VALUE (CLV) ANALYSIS

HOPMONK Gaming Company

AGENDA

- Problem Statement
- Data Pre Processing and Feature Engineering
- Model Building
- Error Metric Evaluation
- Data Insights
- Summary

OVER VIEW

Objective

Using machine learning techniques analyze the data and predict the **Customer Life Time Value CLV** that will enable Hopmonk to target and acquire customers based on the net potential as profit.

Hopmonk Storage and the Approach

- Hopmonk currently has a million customers on their Enterprise Data Warehouse (EDW) where the data is spread across various tables.
- Hopmonk has implemented both Oracle and Cognos solutions designed to enable their business users to extract and analyze their data.
- Hopmonk feels they have locked away valuable details on consumer behavior, segmentation, demographics, and more.

DATA PRE-PROCESSING AND FEATURE ENGINEERING

Data Pre-Processing

7 Data frames are merged together based on the business problem and a final master data frame is made.

Based on the Variables business importance the missing NA values are been imputed.

Feature Engineering

A new Features are been extracted based on the existing variables.

New Features are:
Frequency, Recency, Average Order Value, Purchase-frequency and Customer life time value.

Frequency=Sum(App+LF)

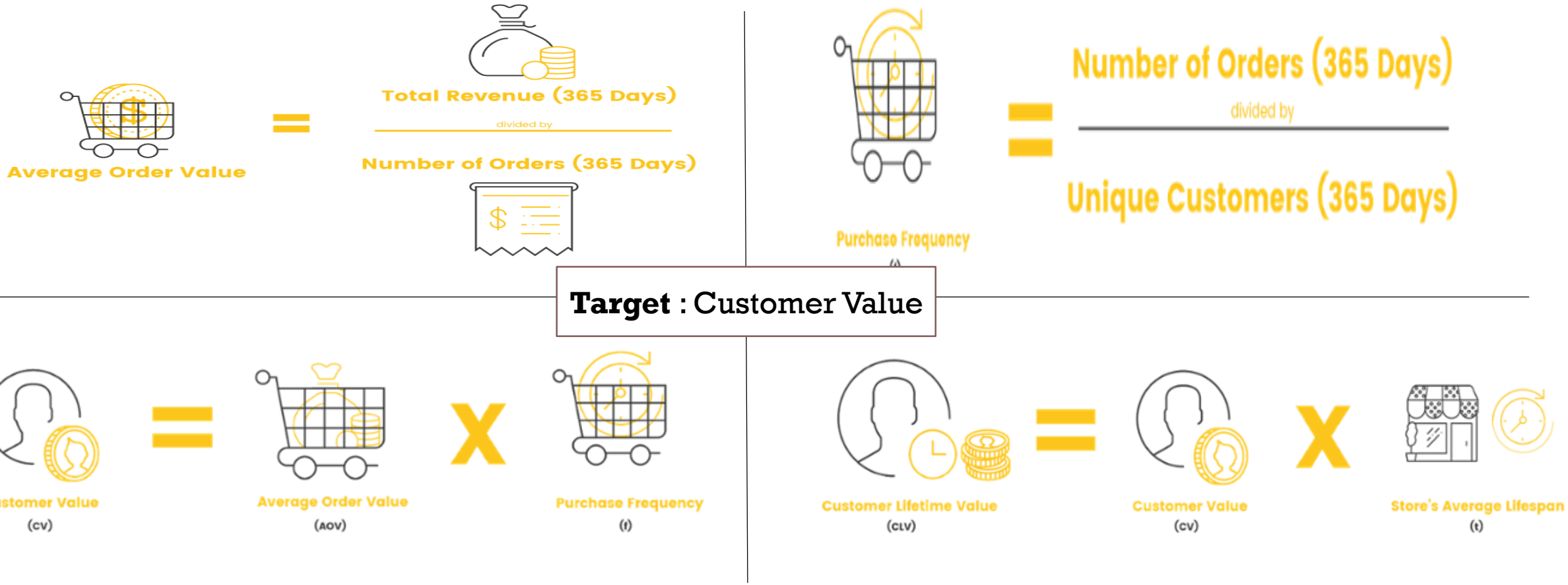
Recency=Recent value(App and LF

Variables Extraction

By using Hetcor function the variables which are highly corelated are been removed.

And using the Step Aic and VIF some of the variables are been removed.

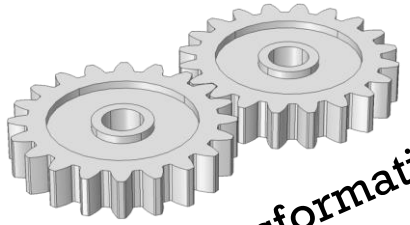
FEATURE ENGINEERING



Variables used:
Total Revenue Generated
Units bought
No. of Unique Customer.

DEMO GRAPHS OF THE FINAL DATA FRAME

Pre-  l Data Frame With 0 by 103



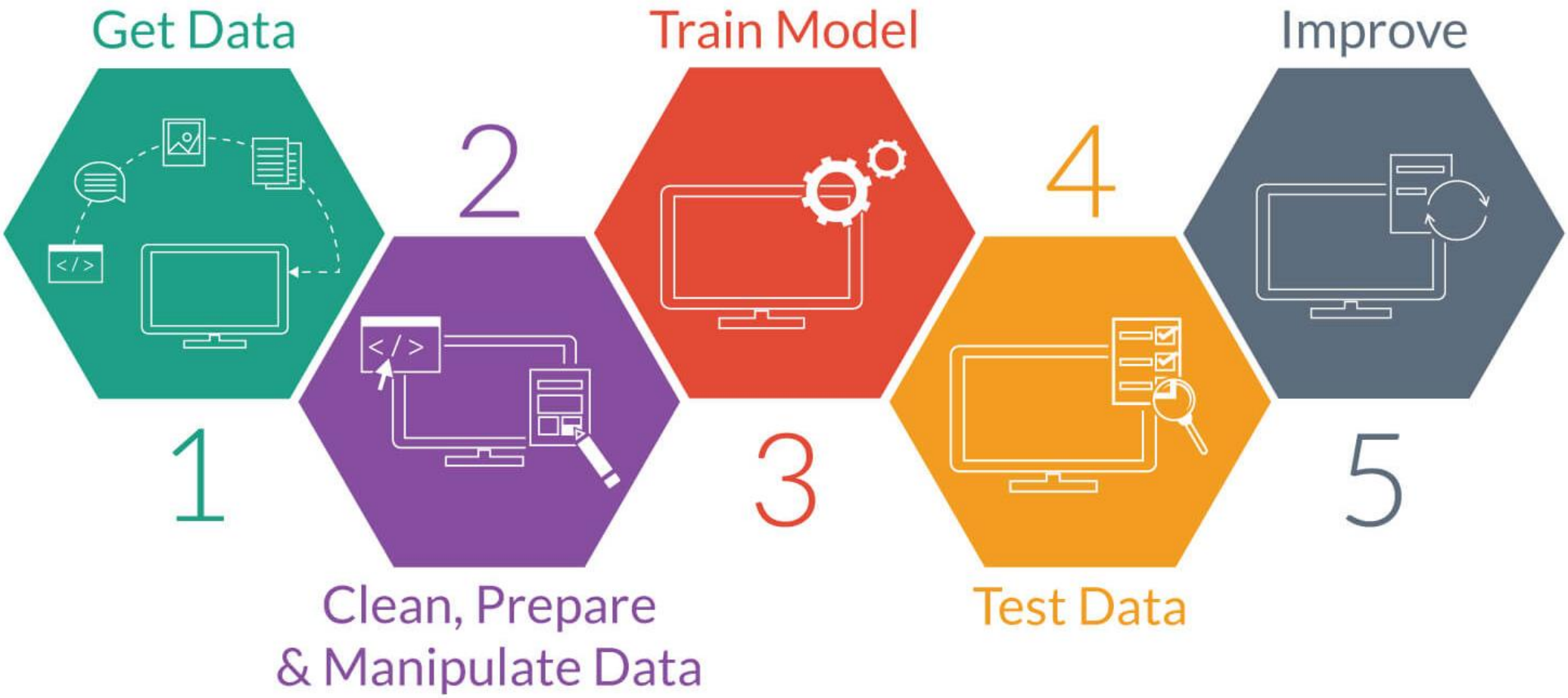
Feature engineered Data Frame With 29271 by 83

Customer Centric	Business Centric	Customer Demo graphs
Games Played,	Favourite Source	Age
Time	Favourite Channel	Country
Units		No of Children
Recency, Frequency		

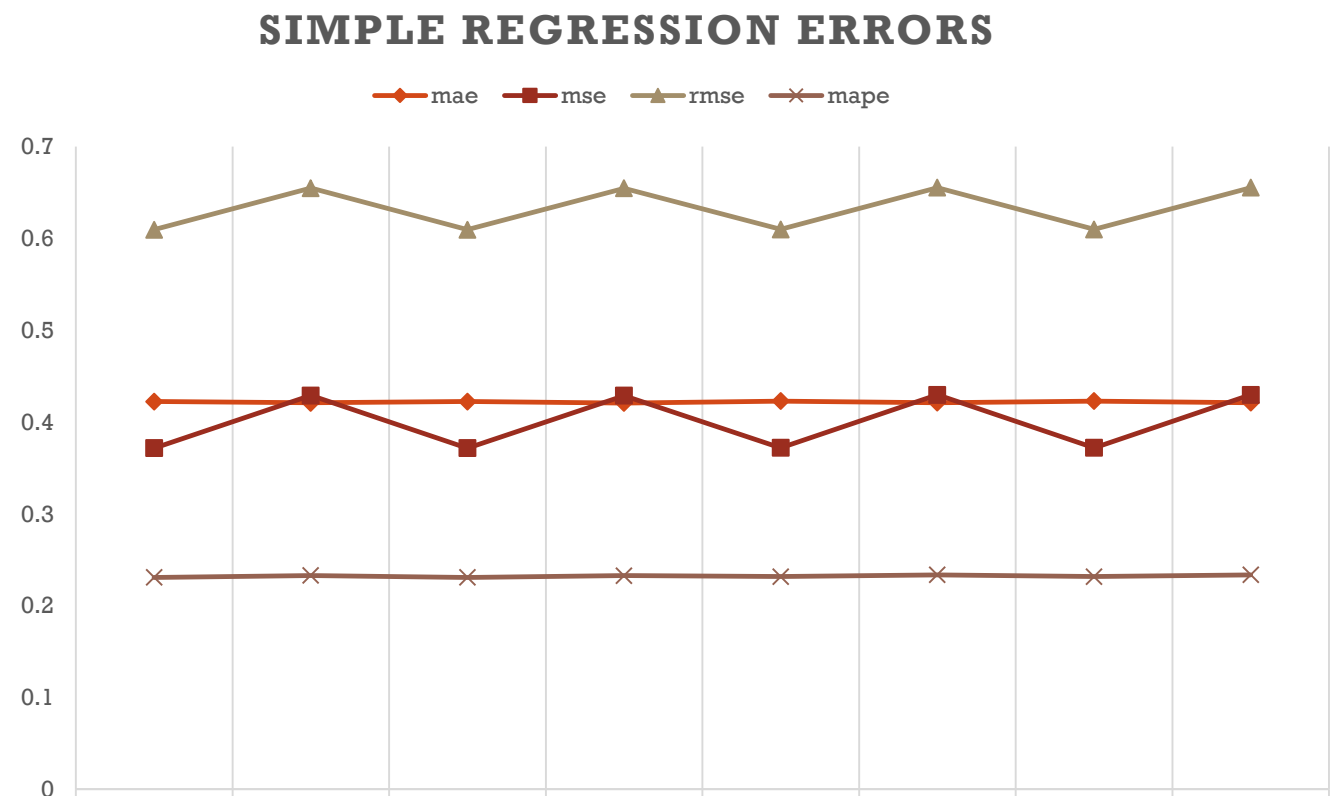
Hopmonk Dataset Summary:

Total Observations : 29271
Total variables : 24
Split : 70-30
Target Variable : Customer_value
Scaling : Based on the model

MODEL BUILDING



SIMPLE LINEAR REGRESSION

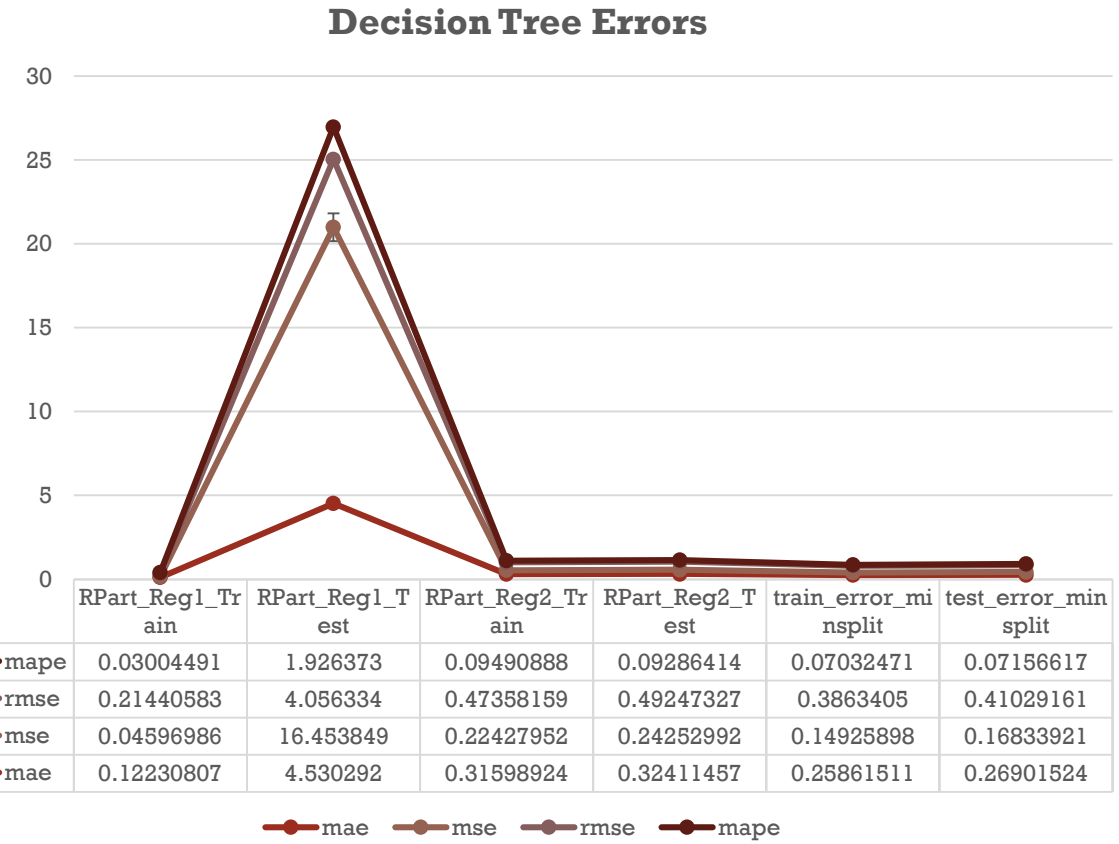


	train_error	test_error	AIC_train_error	AIC_test_error	etrain_lasso	etest_lasso	etrain_ridge	etest_ridge
mae	0.4224153	0.4208756	0.4224147	0.4208535	0.4228082	0.4210808	0.4228082	0.4210808
mse	0.3715517	0.4289595	0.3715654	0.4287616	0.3719724	0.4296397	0.3719724	0.4296397
rmse	0.6095504	0.65495	0.6095616	0.6547989	0.6098954	0.6554691	0.6098954	0.6554691
mape	0.2308342	0.232828	0.2308084	0.2327639	0.2316745	0.2335838	0.2316745	0.233583

Simple Linear Regression:

- The errors were computed based on the simple linear regression.
- Mape stood almost same for various types of regression models.
- The Adjusted R2 is Around 0.90 so even the model is good.
- There is some non-linearity in the data as the data is heavily skewed.
- Did the transformation on the target variable such as sinh(x),Log(x), x2 and Sqrt(x).
- The model gave optimum values over the transformation of sqrt over other transformation.

DECISION TREE



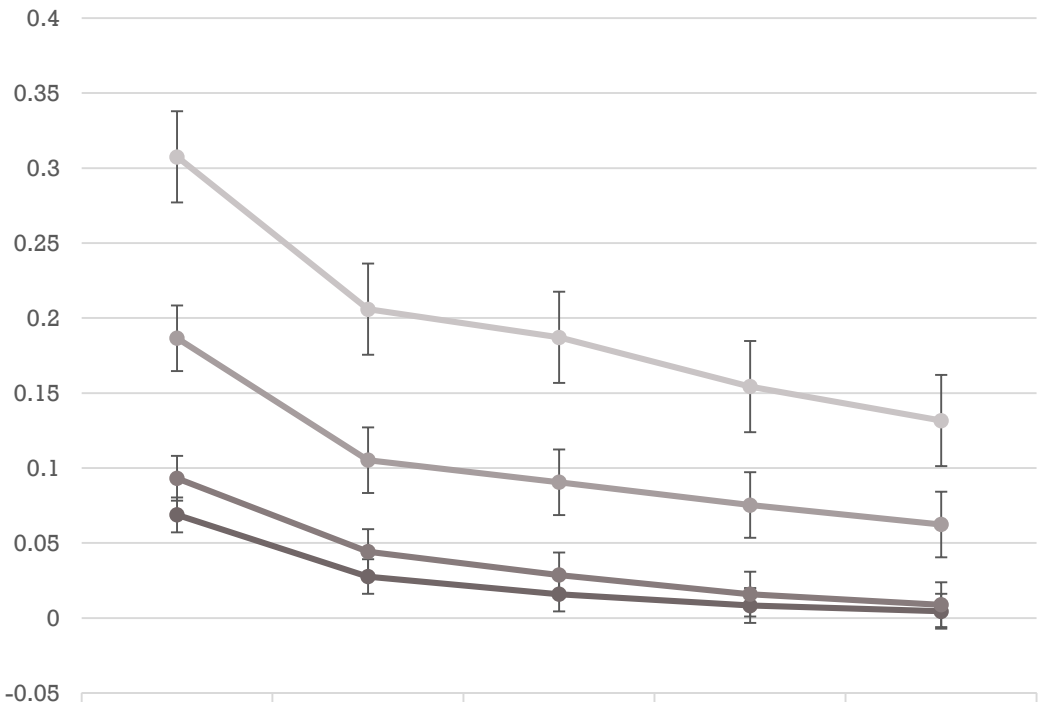
	RPart_Reg1_Train	RPart_Reg1_Test	RPart_Reg2_Train	RPart_Reg2_Test	train_error_minsplit	test_error_minsplit
mape	0.03004491	1.926373	0.09490888	0.09286414	0.07032471	0.07156617
rmse	0.21440583	4.056334	0.47358159	0.49247327	0.3863405	0.41029161
mse	0.04596986	16.453849	0.22427952	0.24252992	0.14925898	0.16833921
mae	0.12230807	4.530292	0.31598924	0.32411457	0.25861511	0.26901524

Decision Tree Model:

- The tree was made to grow without any constraints due to which the model got overfit.
- Then went for the optimal cp value based on the x–relative error.
- Then tuning the number of minimum split gave the least rmse and mape and stable over both train and test.
- As decision tree uses works with the greedy approach some sort o information might be left to get so going for the other models.

RANDOM FOREST

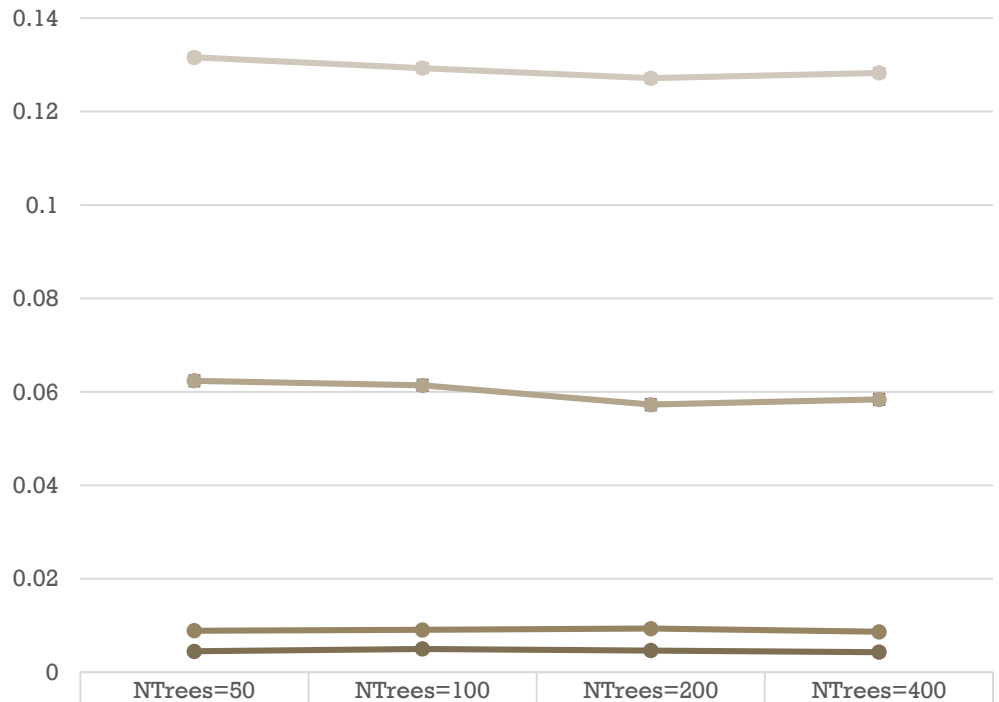
Errors at Various Mtrys and Ntrees Kept Constant



	Mtry=3	Mtry=5	Mtry=6	Mtry=8	Mtry=10
Train_Mape	0.06874258	0.02766684	0.01593094	0.008346876	0.004462201
Test_Mape	0.09316026	0.04421091	0.02869938	0.01590581	0.008871985
Train_RMSE	0.18653861	0.10525012	0.09055661	0.075292897	0.062366102
Test_RMSE	0.30743742	0.20587073	0.18712968	0.15431601	0.131596873

Train_Mape Test_Mape Train_RMSE Test_RMSE

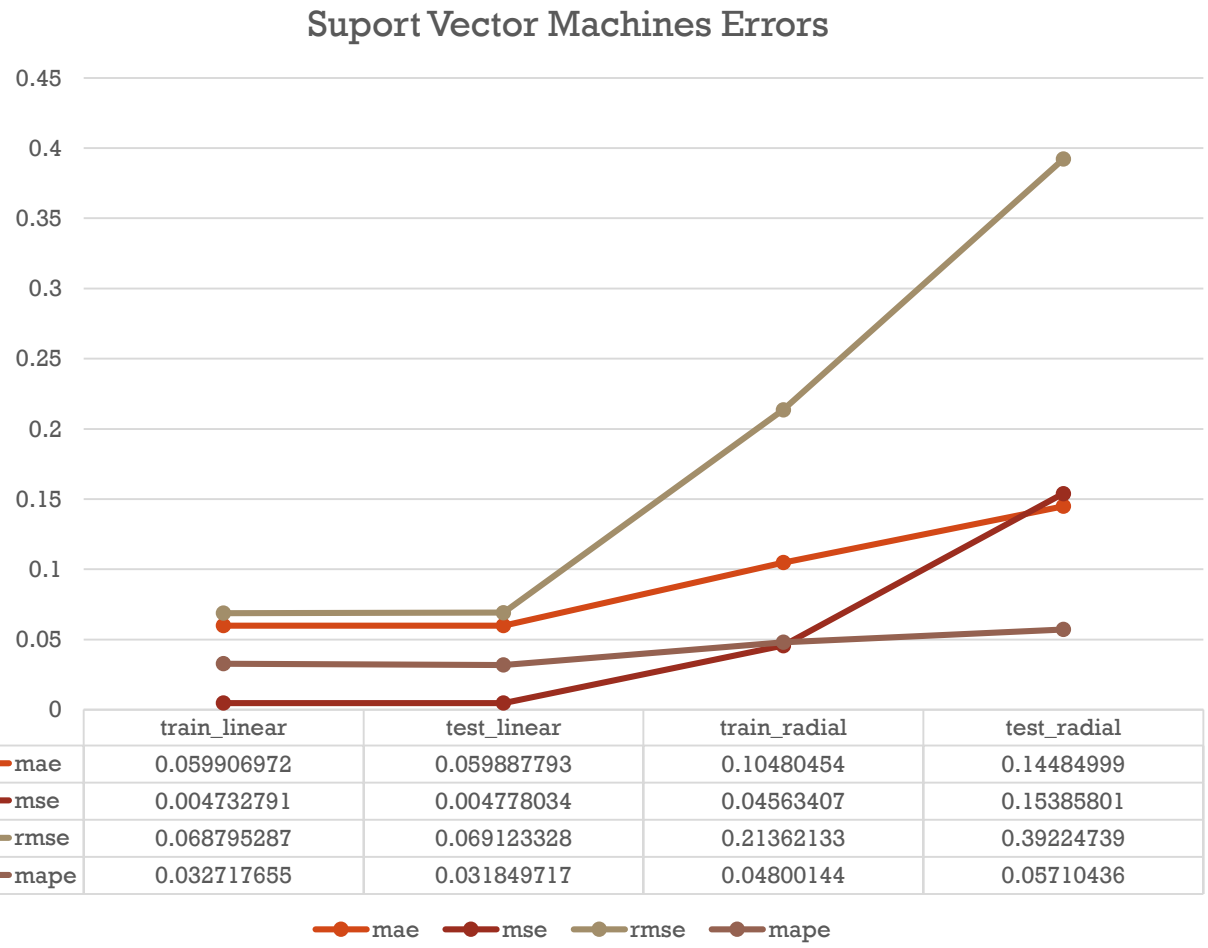
Errors at Various Ntrees and Mtrys Kept Constant



	NTrees=50	NTrees=100	NTrees=200	NTrees=400
Train_Mape	0.004462201	0.004955279	0.004628257	0.004278595
Test_Mape	0.00887198	0.009037344	0.009309786	0.008645399
Train_RMSE	0.062366102	0.061375367	0.057256029	0.058371539
Test_RMSE	0.131596873	0.129303579	0.127114068	0.128250921

Train_Mape Test_Mape Train_RMSE Test_RMSE

SVM

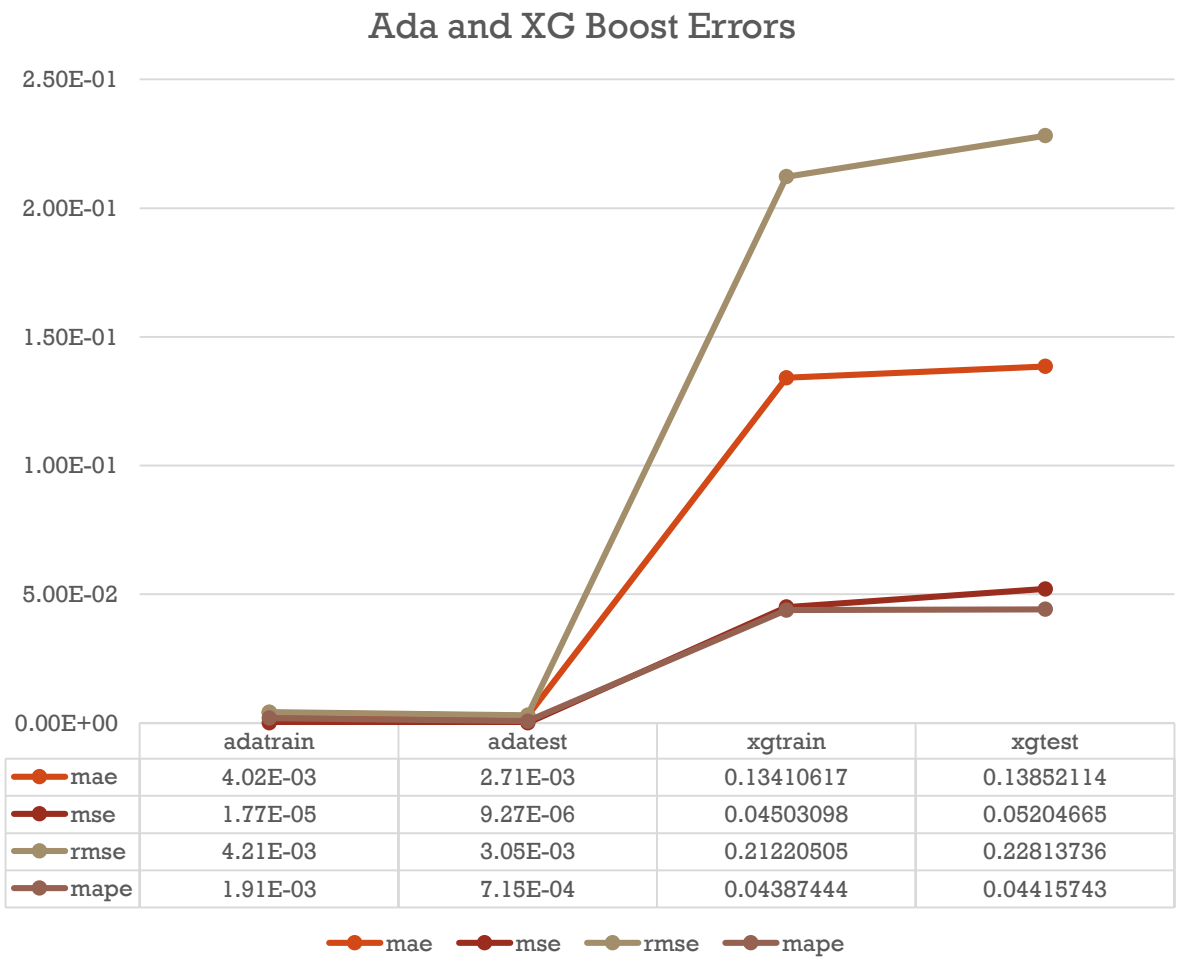


	train_linear	test_linear	train_radial	test_radial
mae	0.059906972	0.059887793	0.10480454	0.14484999
mse	0.004732791	0.004778034	0.04563407	0.15385801
rmse	0.068795287	0.069123328	0.21362133	0.39224739
mape	0.032717655	0.031849717	0.04800144	0.05710436

SVM Regression:

- The values are separable well in the linear kernel and even the rmse of train and test is stable.
- The Tuning Parameter were not taken in account at the cost of time to tune in spite of errors being stable over train and test.
- But, in radial in hyperplane its not generalising well as the data might be linearly separable in linear SVM

ADA AND XG BOOST



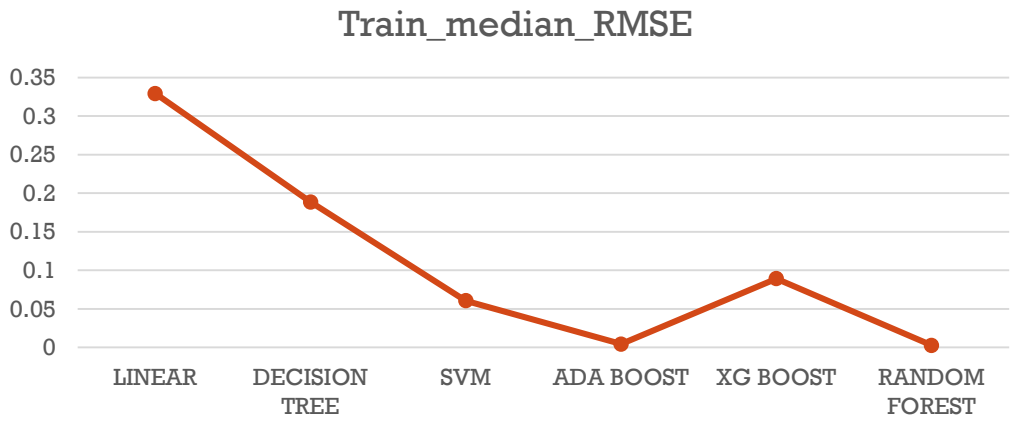
	adatrain	adatest	xgtrain	xgtest
mae	4.02E-03	2.71E-03	0.13410617	0.13852114
mse	1.77E-05	9.27E-06	0.04503098	0.05204665
rmse	4.21E-03	3.05E-03	0.21220505	0.22813736
mape	1.91E-03	7.15E-04	0.04387444	0.04415743

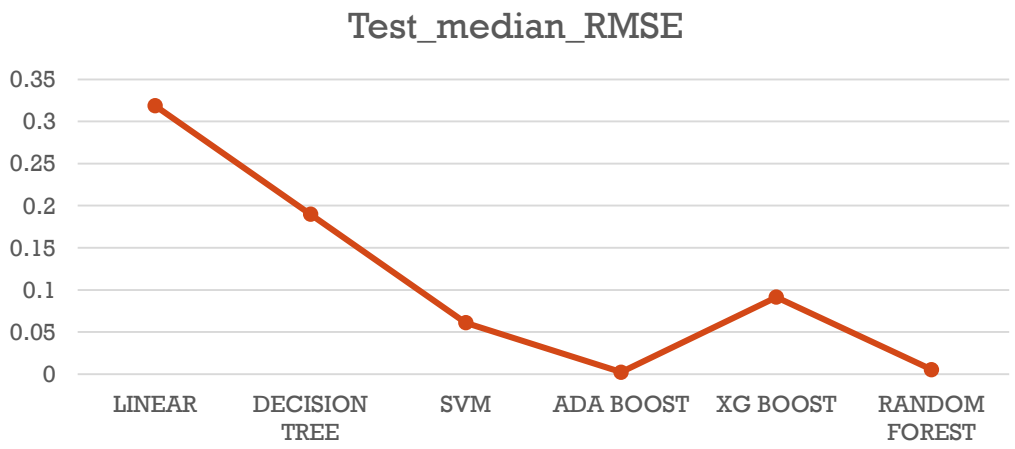
ADA and XG Boost:

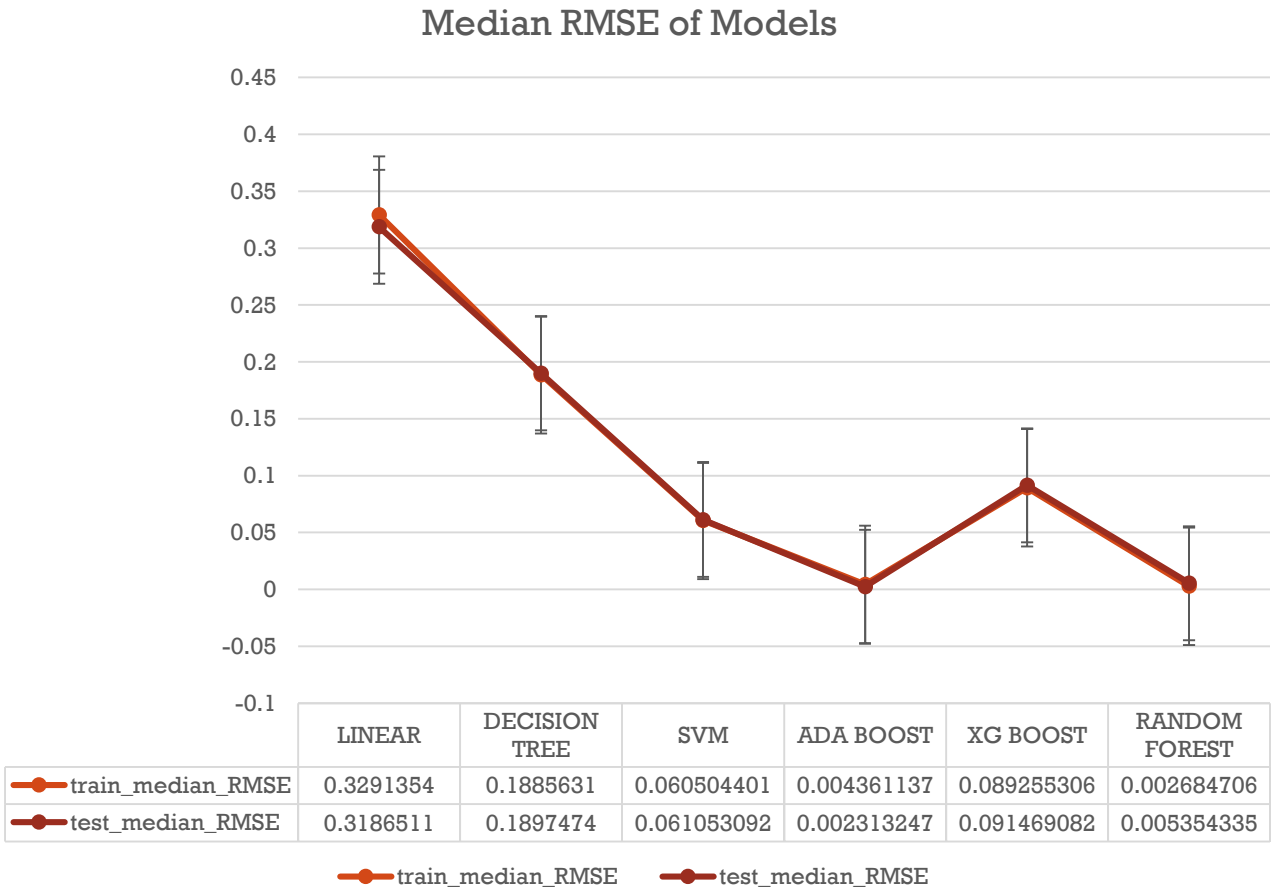
- As ADA Boost take the class vectors did the range normalisation and predicted the probabilities.
- Then did denormalization of the probabilities and got the train and test error nearly about zero.

- XG Boost errors mape is generalised on train and test but the error deviation is high on both train and test due to which rmse is high when compared to other models.

Error Metric Evaluation Based on Median RMSE





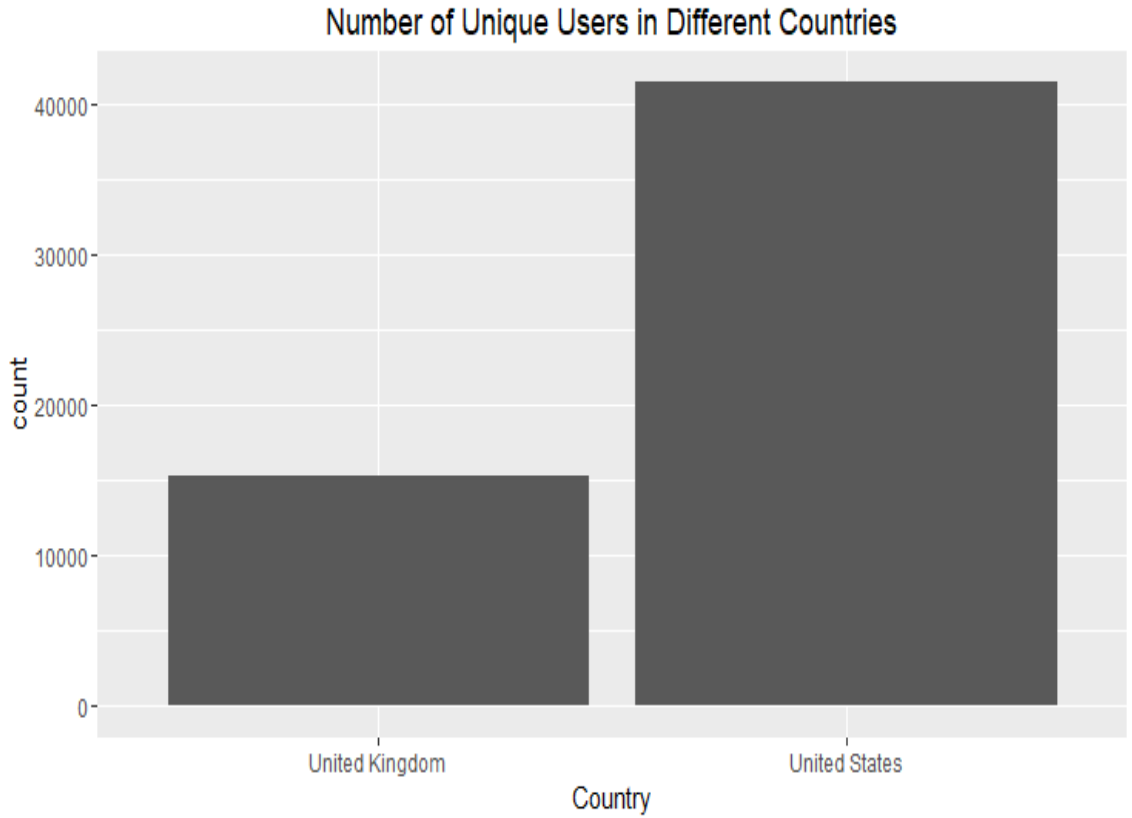


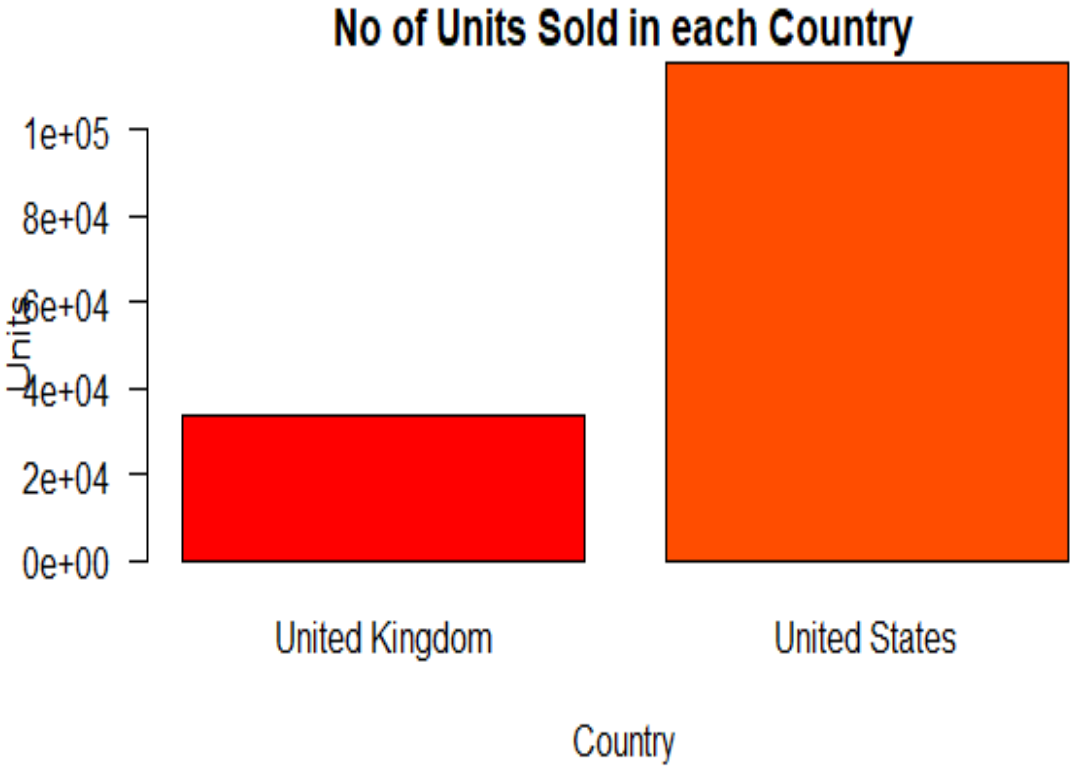
	LINEAR	DECISION TREE	SVM	ADA BOOST	XG BOOST	RANDOM FOREST
train_median_RMSE	0.3291354	0.1885631	0.060504401	0.004361137	0.089255306	0.002684706
test_median_RMSE	0.3186511	0.1897474	0.061053092	0.002313247	0.091469082	0.005354335

Final Model :SVM Model

SUMMARY AND DATA INSIGHTS

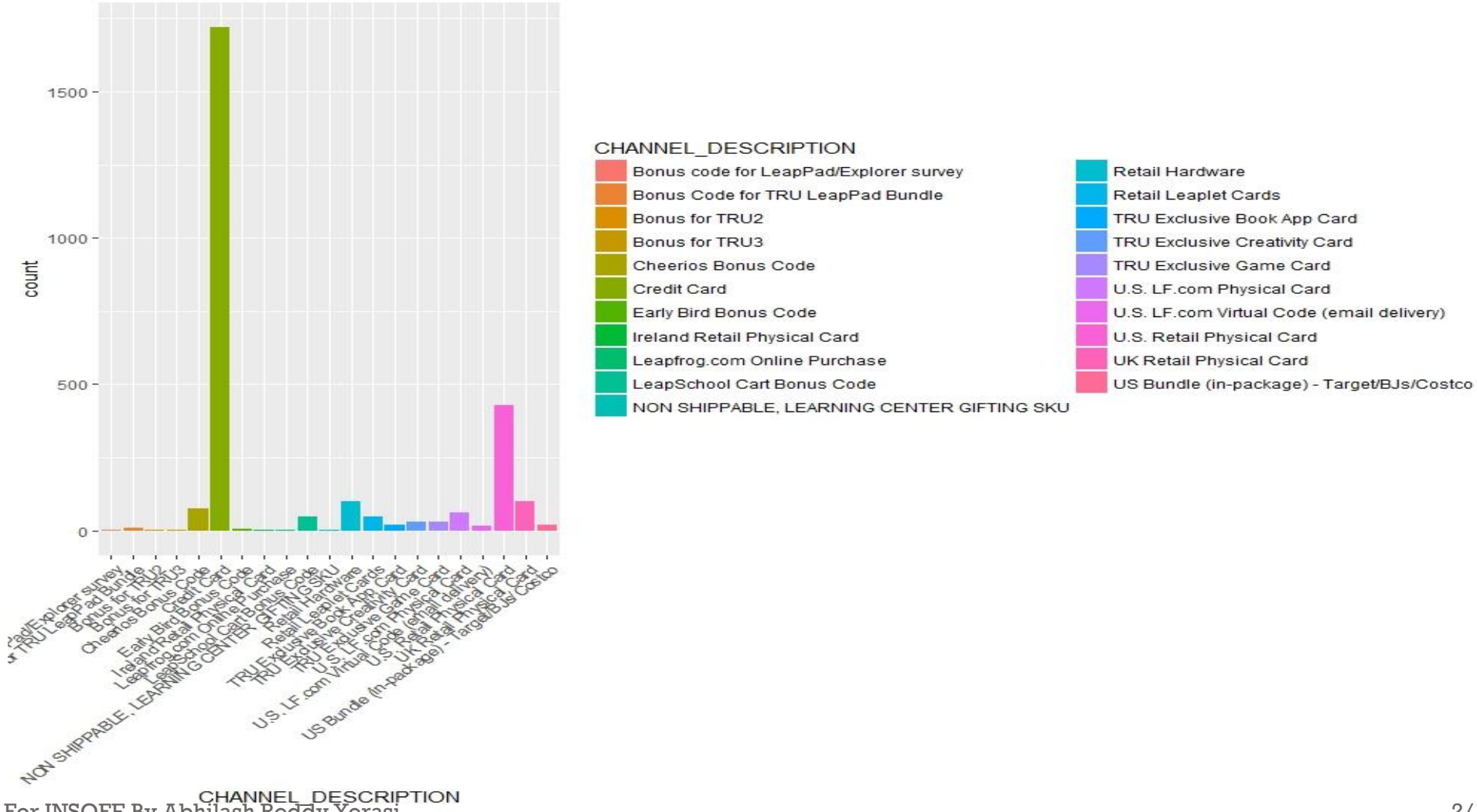
DATA INSIGHTS BEFORE PRE-PROCESSING



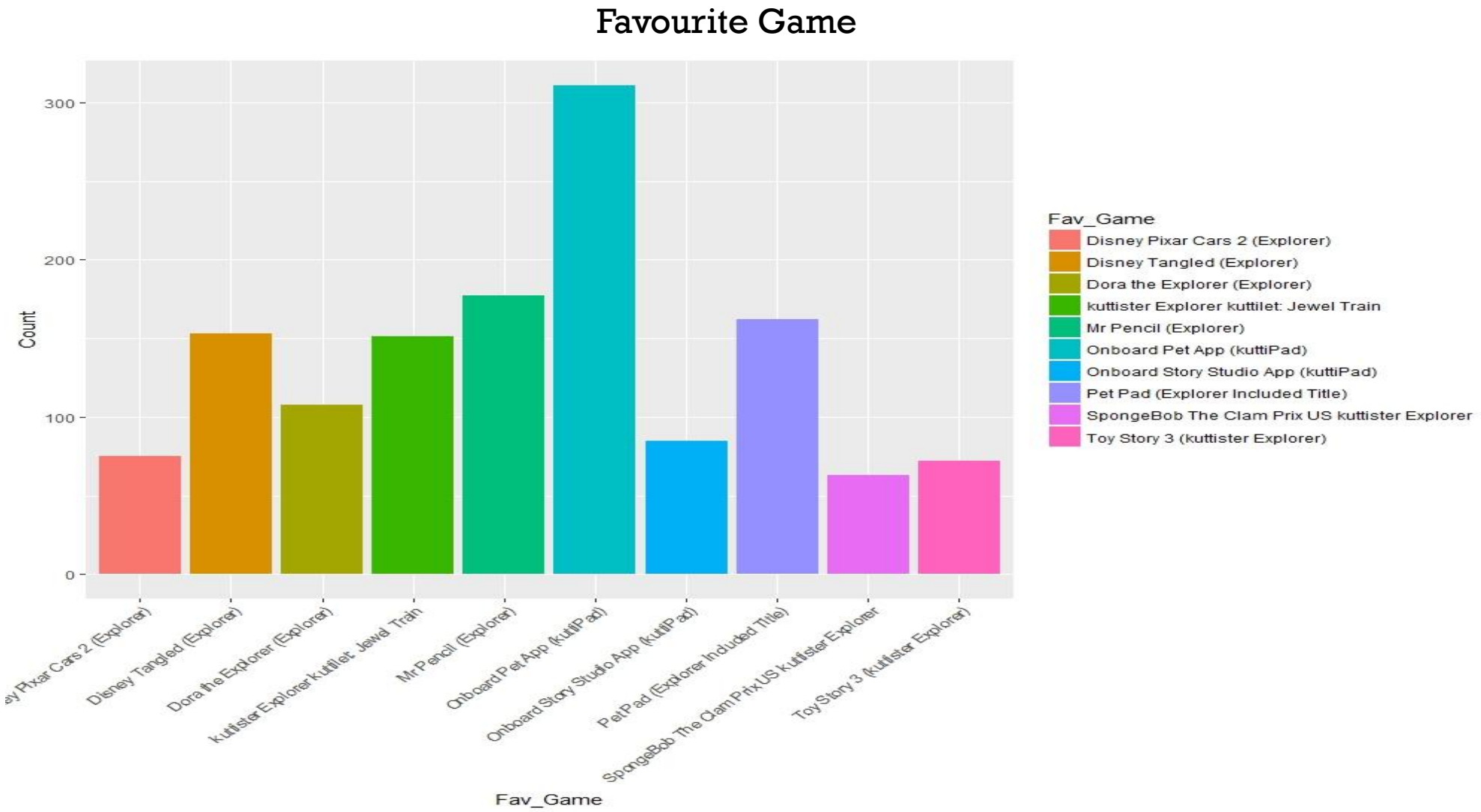


Data Insights Before Pre-Processing

Favourite Channel of Transactions



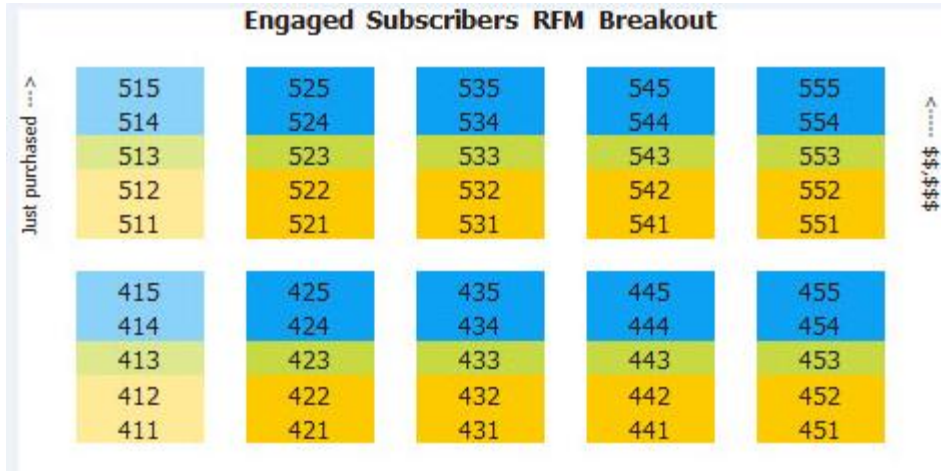
DATA INSIGHTS BEFORE PRE-PROCESSING



DATA INSIGHTS BEFORE PRE-PROCESSING

RFM Analysis Table

RFM555	RFM455	RFM554	RFM545	RFM515	RFM245
87890060	89620432	90600609	89435870	87838511	88431015
88686514		96136780	90511634	87975165	
88721453		96934816	90863436	88089825	
90343573		98825541	90896541	88388798	
91409769			91003460	88422776	
91730072			91577505	88650948	
91838002			96392554	89017661	
92139318			96571486	89165084	
106253219				89171459	
				89515450	
				90823467	
				91004998	
				96149752	
				96633987	
				99678091	
				99802660	
				101022250	



SUMMARY

- As per the business constraints the Customer Life Time value has to be predicted and the SVM model is stable over others when compared in terms of Errors.
- As there is huge difference in the markets of US and UK the marketing strategies has to be improve a lot in UK market to scale up the profitability.
- As more customers are been inclined to the credit card as the channel of transaction company should make the portal highly secured.
- As there is a game named Disney Pixar car2 which is played a lot when compared to others a gamed around that themes will make the company to generate more revenue.
- There should be customers centric marketing strategies to retain the them and increase the revenue.

Q&A

THANK YOU