Project Final Report: Prediction of New Car Prices

Executive Summary

The United States has one of the largest automotive markets in the world. In 2018, U.S. light vehicle sales reached 17.2 million units, the fourth straight year in which sales reached or surpassed 17 million units. Overall, the United States is the world's second-largest market for vehicle sales and production. The prices of new cars in the industry is fixed by the manufacturer with some additional costs incurred by the Government in the form of taxes. So, customers buying a new car can be assured of the money they invest to be worthy.

There are a variety of features of a car like the age of the car, its make, the origin of the car (the original country of the manufacturer), its mileage (the number of kilometers it has run) and its horsepower. Due to rising fuel prices, fuel economy is also of prime importance. Other factors such as the type of fuel it uses, style, braking system, the volume of its cylinders (measured in cc), acceleration, the number of doors, safety index, size, weight, height, paint color, consumer reviews, prestigious awards won by the car manufacturer. Other options such as sound system, air conditioner, power steering, cosmic wheels, GPS navigator all may influence the price as well.

We have used a dataset from Kaggle (https://www.kaggle.com/prassanth/new-cars-price-2019) to perform price prediction of new cars and it is freely available for public use. This dataset has 32,316 rows and 57 columns.

In this study, price prediction was done using different machine learning models. We have evaluated the performance using several machine learning algorithms like linear regression, decision tree, random forest and gradient boost algorithm. Among all these models, random forest classifier proves to perform the best for their prediction task

Problem Statement

For the last century, the entire globe has witnessed car culture. The global automotive industry is in better shape than before. Automobiles are highly differentiated durable goods with variable lifetimes. A car price prediction has been a high-interest research area, as it requires noticeable effort and knowledge of the field experts.

Our project is to understand the factors influencing pricing of cars. Our goal is to predict car prices for new cars using various attributes.

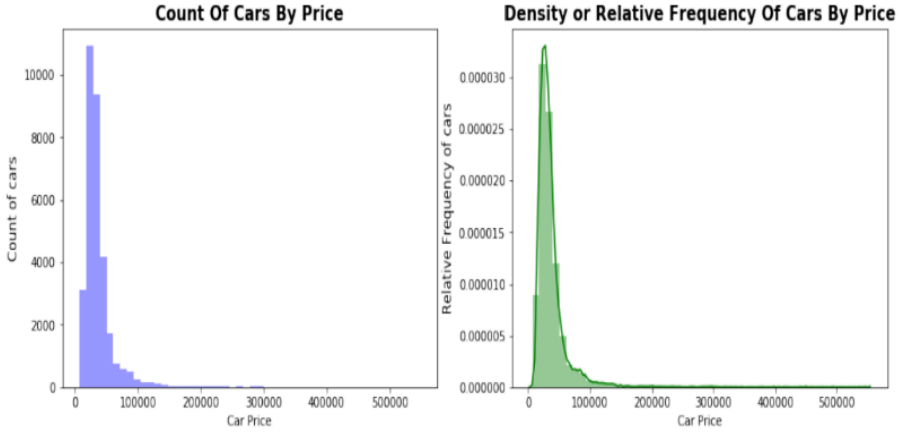
Discussion of Project Scope

- To develop a model for predicting the price of the cars with the available independent variables
- To predict which variables are significant in predicting price of the car
- To understand how exactly the prices vary with the independent variables.
- How well those variables define the price of the car.
- Using SAS Enterprise Miner, perform linear regression, decision tree and gradient boost algorithms to predict price, perform K-fold Cross validation.

Identification of key measures used to evaluate the success of our project:

- MSRP - Selling price in dollars is the response variable and rest all other variables/factors influence the price of the car.
- This dataset has 32,316 rows and 57 columns.

Following are some of the important observations on our target variable "MSRP":



- As shown in the figure the target variable has a positive skew.
- More than 50% of the cars are priced less than 100000.
- Based on the observations and graph on the right side (KDE/green one), it appears that there is only one distribution for cars priced less than 200000.

Data Cleaning Before Launching SAS Enterprise Miner

Preprocessing of data was done using Python 3.8 and Microsoft Excel.

Quantitative Data (Interval): Following variables were converted to numeric data type:

- MRSP - Selling price in dollars (Target Variable – Had '$' sign in the original dataset which was removed)
- Corrosion Miles/km - Warranty against corrosion
- Drivetrain Miles/Km - Warranty for drivetrain
- EPA Fuel Economy Est - City (MPG)- Mileage in city
- Roadside Assistance Miles/km - Roadside assistance provided in terms of km
- SAE Net Horsepower @ RPM - Horsepower produced at engine crankshaft (without transmission losses)
- SAE Net Torque @ RPM - Net optimum torque at certain range of RPM

Qualitative Variables (Nominal):

- Drivetrain - Types of drivetrain (Conveys power from engine to wheels) used (FWD, RWD, AWD, 4WD)
- Engine - Type of engine used in car
- Front Wheel Material - Material in which wheel is made up of
- Fuel System - Type of fuel injection used in car
- Suspension Type Front - Type of suspension used in car front wheels
- Suspension Type Rear - Type of suspension used in car rear wheels
- Trans Description Cont - Type of transmission used in car

Binary Variables: – Following nominal variables were converted to binary data type (Yes = 1 and No = 0):

- Air Bag Frontal Driver - Drivers Airbag(Y/N)
- Air Bag Frontal Passenger - Front passenger Airbag(Y/N)
- Air Bag-Passenger Switch (On/Off) - Airbags in rear side of the car(Y/N)
- Air Bag-Side Body-Front - Airbags in front side of the car(Y/N)
- Air Bag-Side Body-Rear - Airbags in rear side of the car(Y/N)

- Air Bag Side Head-Front - Side airbags to protect head for front row(Y/N)
- Air Bag Side Head-Rear - Side airbags to protect head for rear row(Y/N)
- Back up Camera - Is reversing camera present (Y/N)
- Brakes -ABS - Is Antilock braking system is present(Y/N)
- Child Safety Rear Door Locks - Child safety door locks(Y/N)
- Daytime Running Lights - (Y/N)
- Fog Lamps - Is fog lamps present(Y/N)
- Night Vision - Thermographic camera to increase a driver's perception in darkness (Y/N)
- Parking Aid - Sensors to monitor nearby obstacles(Y/N)
- Rollover Protection Bar - Is ROPS present (Y/N)
- Stability Control - Is ECS present (Y/N)
- Tire Pressure Monitor - Y/N
- Traction Control - Is TCS available (Y/N)

Creation of New Variables:

The following variables were derived from the variable Model in the original dataset:

- Model Year
- Manufacturer

The following variable was derived from the variable EPA Classification (EPA size classes -Minicompact, Mid-size, Compact, Mid-Size, SUV) in the original dataset:

- Category

The following variables were derived from the variable Front Tire Size in the original dataset:

- Front tire width
- Front tire aspect ratio
- Front tire speed ratings/cons. type
- Front tire rim size

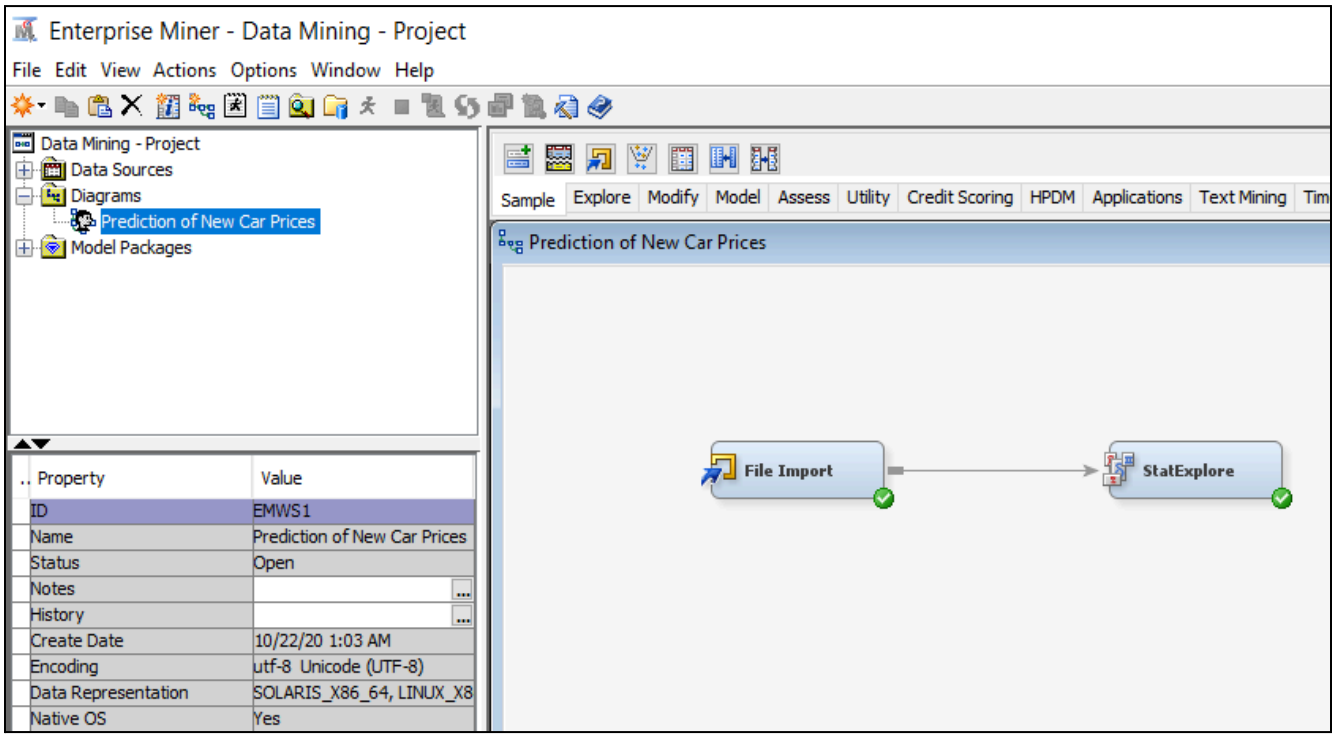
Dropped Variables: Following variables were dropped as the information in them was used to create new variables as explained above.

- Model
- Front Tire Size
- Transmission (Text)
- Body Style

Launching SAS Enterprise Miner

A new project and a new diagram were created using SAS Enterprise Miner. File Import Node was used to import the cleaned file into SAS Enterprise Miner. StatExplore Node was run to gain insights into the data.

Figure-1: File import and StatExplore nodes



The following are the results of StatExplore Node:

Figure -2: Class variables

Results - Node: StatExplore Diagram: Prediction of New Car Prices

File Edit View Window

Output

Class Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	Air_Bag_Frontal_Driver	INPUT	2	0	1	97.19	0	2.81
TRAIN	Air_Bag_Frontal_Passenger	INPUT	2	0	1	95.26	0	4.74
TRAIN	Air_Bag_Passenger_Switch__On_Off	INPUT	2	0	0	89.42	1	10.58
TRAIN	Air_Bag_Side_Body_Front	INPUT	2	0	1	69.45	0	30.55
TRAIN	Air_Bag_Side_Body_Rear	INPUT	2	0	0	94.52	1	5.48
TRAIN	Air_Bag_Side_Head_Front	INPUT	2	0	1	64.04	0	35.96
TRAIN	Air_Bag_Side_Head_Rear	INPUT	2	0	1	55.04	0	44.96
TRAIN	Back_Up_Camera	INPUT	2	0	0	68.24	1	31.76
TRAIN	Brakes_ABS	INPUT	2	0	1	91.77	0	8.23
TRAIN	Category	INPUT	5	1716	Car	42.79	Pic	23.98
TRAIN	Child_Safety_Rear_Door_Locks	INPUT	2	0	1	58.71	0	41.29
TRAIN	Daytime_Running_Lights	INPUT	2	0	1	55.65	0	44.35
TRAIN	Drivetrain	INPUT	5	1716	FWD	28.37	RWD	28.35
TRAIN	Engine	INPUT	14	1975	14	30.86	V8	26.65
TRAIN	Fog_Lamps	INPUT	2	0	0	51.42	1	48.58
TRAIN	Front_Wheel_Material	INPUT	5	1983	Aluminum	61.81	Steel	22.85
TRAIN	Front_tire_speed_ratings_cons_ty	INPUT	6	0	R	49.54	NA	18.83
TRAIN	Front_tire_width	INPUT	44	0	235	13.17	245	12.23
TRAIN	Fuel_System	INPUT	4	2830	SFI	38.88	DI	32.94
TRAIN	Manufacturer	INPUT	43	0	Ford	12.95	Chevrolet	9.21
TRAIN	Night_Vision	INPUT	2	0	0	99.86	1	0.14
TRAIN	Parking_Aid	INPUT	2	0	0	81.81	1	18.19
TRAIN	Roadside_Assistance_Years	INPUT	7	21027		65.07	3	11.72
TRAIN	Rollover_Protection_Bars	INPUT	2	0	0	96.33	1	3.67
TRAIN	Stability_Control	INPUT	2	0	1	67.24	0	32.76
TRAIN	Suspension_Type___Front	INPUT	10	0	MacPherson Strut	37.81	Double Wishbone	19.67
TRAIN	Suspension_Type___Rear	INPUT	10	0	Link type	55.04	Leaf type	9.35
TRAIN	Tire_Pressure_Monitor	INPUT	2	0	1	68.80	0	31.20
TRAIN	Traction_Control	INPUT	2	0	1	70.93	0	29.07
TRAIN	Trans_Description_Cont_	INPUT	4	0	Automatic	59.06	Manual	29.70
TRAIN	Trans_Type	INPUT	10	1981	6	35.43	5	23.08

Figure-3: Interval Variables:

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
Base_Curb_Weight__lbs_	INPUT	3610.225	695.1866	19456	12860	1808	3497	8591	0.571347	0.537205
Basic_Miles_km	INPUT	42739.97	14332.2	30199	2117	24000	36000	150000	4.591368	29.24604
Basic_Years	INPUT	3.398755	0.662203	30199	2117	2	3	6	1.308795	1.151892
Corrosion_Miles_km	INPUT	133587.3	27212.32	29312	3004	50000	150000	150000	-1.41581	1.047433
Corrosion_Years	INPUT	6.341056	2.460939	29350	2966	2	5	12	1.665726	1.217638
Displacement	INPUT	3.542469	1.412912	30169	2147	0.65	3.5	7.5	0.413659	-0.98477
Drivetrain_Miles_km	INPUT	66368.12	22712.74	29528	2788	24000	60000	150000	1.054367	1.079041
Drivetrain_Years	INPUT	5.070069	1.81561	29528	2788	2	5	20	3.096149	16.64827
EPA_Fuel_Economy_Est___City__MPG	INPUT	19.5682	5.589965	27027	5289	9	18	66	2.058189	8.407641
Front_tire_aspect_ratio	INPUT	59.16397	11.56089	30304	2012	30	60	85	-0.28535	-0.88848
Front_tire_rim_size	INPUT	17.24248	1.453867	27817	4499	15	17	22	0.508696	0.003671
Fuel_Tank_Capacity__Approx__gal	INPUT	22.027	7.017269	15615	16701	1.9	19.5	48	0.685508	-0.374
Height__Overall__in_	INPUT	66.44527	9.372474	15628	16688	44.3	67.5	110.1	0.183355	-0.2566
Model_year	INPUT	2009.535	7.198832	32316	0	1990	2011	2019	-0.70824	-0.23193
Passenger_Capacity	INPUT	4.762904	1.824018	32316	0	0	5	15	-0.4357	2.723524
Passenger_Doors	INPUT	3.263399	1.155039	32316	0	0	4	4	-1.35219	0.855727
Roadside_Assistance_Miles_km	INPUT	79394.54	45007.96	25293	7023	25000	60000	150000	0.623517	-1.20762
SAE_Net_Horsepower___RPM	INPUT	258.4166	98.64659	30302	2014	40	252	808	0.914731	1.469596
SAE_Net_Torque___RPM	INPUT	269.1232	102.7461	30249	2067	65	260	935	0.756812	1.027629
Stabilizer_Bar_Diameter___Front	INPUT	1.133912	0.220238	11207	21109	0.63	1.13	1.68	-0.25017	-1.19899
Track_Width__Front__in_	INPUT	61.68542	2.627705	20130	12186	50	61.6	69.4	0.168479	0.385334
Turning_Diameter___Curb_to_Curb	INPUT	39.76493	5.674009	28885	3431	20.5	38.1	93.6	1.796364	8.423493
VAR8	INPUT	100.4615	21.93942	16756	15560	21.28	97.5	188.4	1.089519	2.781327
Wheelbase__in_	INPUT	117.8425	18.06418	30301	2015	73.5	111.5	178	1.086019	0.45237
MSRP	TARGET	37707.46	32392.38	32262	54	6929	30555	548800	6.315598	60.58142

From the results of StatExplore, the following variables role was set to rejected as they had more than 35% missing values:

- Base Curb Weight - Total weight of the vehicle in pounds – 12, 860 missing values
- Fuel Tank Capacity, Approx.(gal) - Fuel tank capacity in gallon – 16,701 missing values
- Height Overall - Overall height of the car in inches – 16,688 missing values
- Stabilizer Bar Diameter - Front (in) - Diameter of the front sway bar – 21,109 missing values
- Var-8. i.e. Passenger Volume - Volume of space available for passengers – 15,560 missing values

Graph Explore Node and Multiplot Node

Graph Explore Node was used to visualize various distributions of data. Multiplot Node was used to graphically explore large volumes of data, observe data distributions, and to examine relationships among the variables.

Rationale for Any Deviation from Pre-specified Analysis Plan Performed

Initially we split the data in the ratio 40:30:30 i.e. train(40%), validate (30%) and test(30%). We have now used the splitting ratio of 60:20:20 i.e.train(60%), validate(20%) and test(20%). Our model performs better with lower average squared error.

Moreover, our target variable is continuous in nature. So we have used Average Squared Error as our selection statistic instead of misclassification rate.

Addition of Random Forest machine learning technique for model comparison. We found that Random Forest performs better than all other machine learning techniques for our dataset.

Model Evaluation

Entire data was split into three sets, one is used to train the model upon, the second set is used for validation and the third set is kept as a holdout set which is used to test how the model behaves with completely unseen data.

Data Partition Node was used to split the data into train (60%), validate (20%) and test(20%).

Data Cleaning After Launching SAS Enterprise Miner

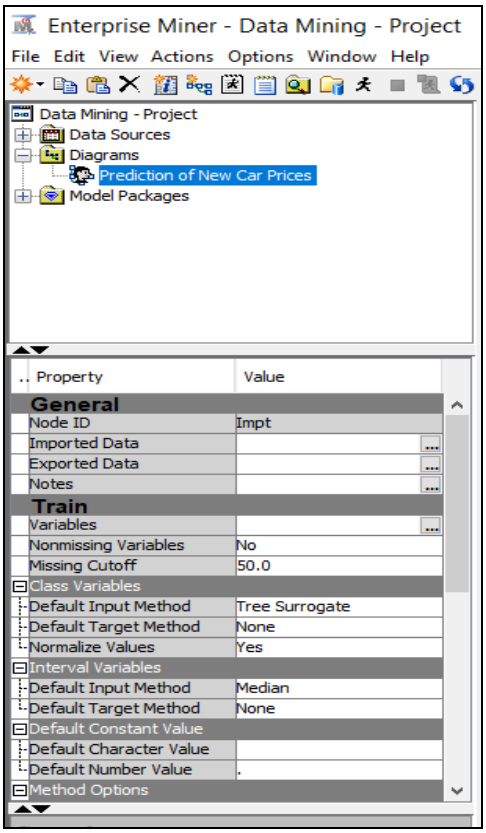
Handling Missing Values Using Impute Node in SAS Enterprise Miner:

Figure -4: Imputation method

For continuous variables, we used median to impute missing values. Median was used instead of mean as outliers are present in the dataset. Class variables were imputed using the tree surrogate method.

Variable Transformation

Transform Variable Node was run to transform the interval variables. Interval variables were on different scales in the dataset and so it is important to transform them to bring them on a similar scale. We tested standardization and log transform methods. Log transformation was more effective to stabilize variances, remove nonlinearity, improve additivity, and correct nonnormality in variables.



Model Development

We tested and compared performance of linear regression, gradient boosting and decision tree models on imputed and transformed variables (*see Figure -5: Model Flowchart figure below*). Based on root average squared error metrics, we found that gradient boosting performed slightly better than other models. Unlike linear regression, the gradient boosting and decision tree algorithms are relatively not affected by missing values and normality assumptions. Therefore, we also tested both tree-based algorithms on raw variables, i.e. before imputation and log transformation. Root average squared error did not change significantly after testing tree-based algorithms on non-transformed variables. Figure - 6 shows comparison of model performance based on root average squared errors.

Linear Regression

Linear regression attempts to predict the value of an interval target as a linear function of one or more independent inputs. Linear Regression was one of the methods used for model analysis. The Regression node belongs to the Model category in the SAS data mining process of Sample, Explore, Modify, Model, Assess (SEMMA). A regression node was used to fit a linear regression model to a predecessor data set in a SAS Enterprise Miner process flow.

The finalized regression model has high F-value and its p-value $<<0.05$, t-statistic p-value for independent variables <0.05 and R^2 >0.88. We arrived at this model by rejecting variables whose t-statistic p-value was greater than 0.05.

Gradient Boosting

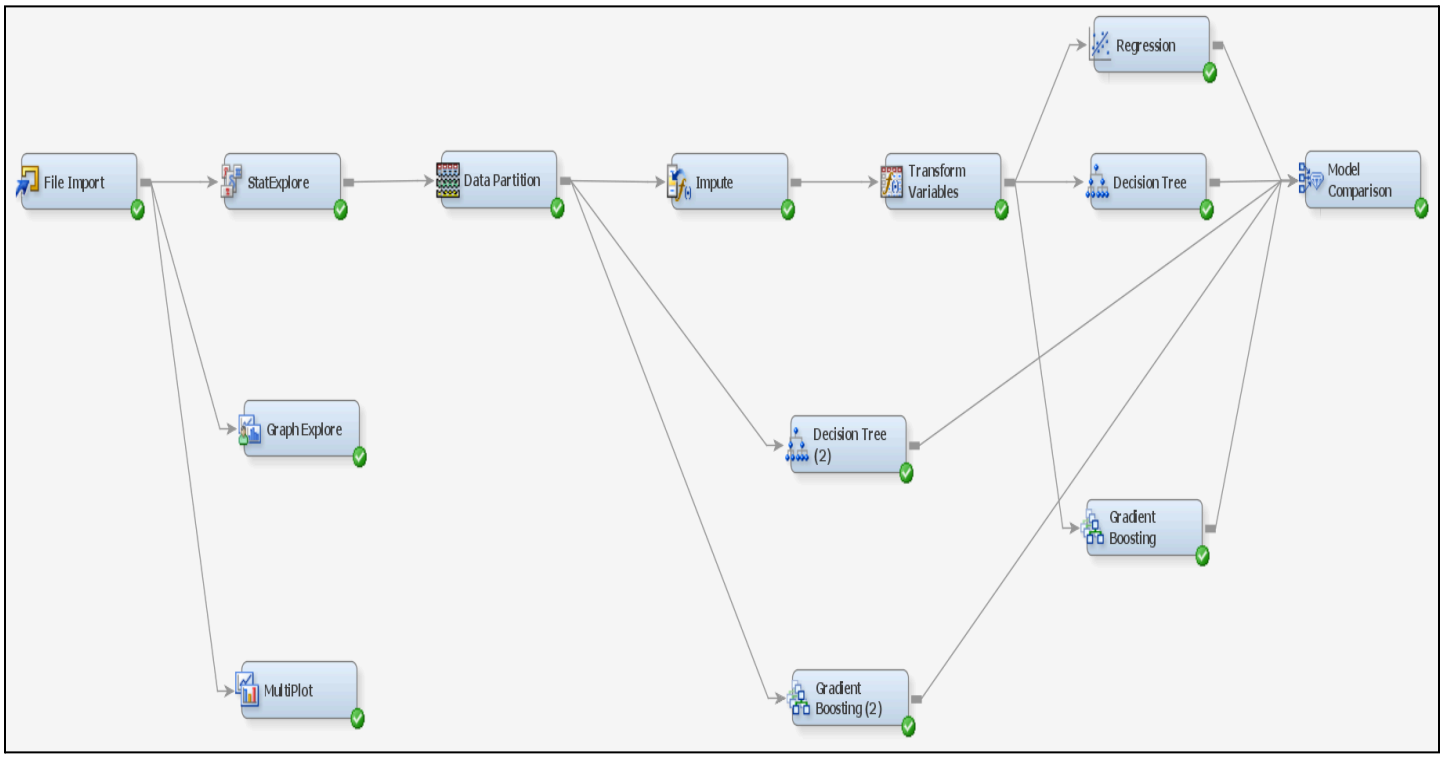
Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing

optimization of an arbitrary differentiable loss function. We used the default settings in SAS-EM for Gradient Boosting.

Decision Tree

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving both regression and classification problems. The goal of using a Decision Tree is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from training data. Similar to gradient boosting, we used default SAS-EM settings for the decision tree algorithm.

Figure -5: Model Flowchart



Conclusion:

Figure -6: Model performance comparison based on root average squared error

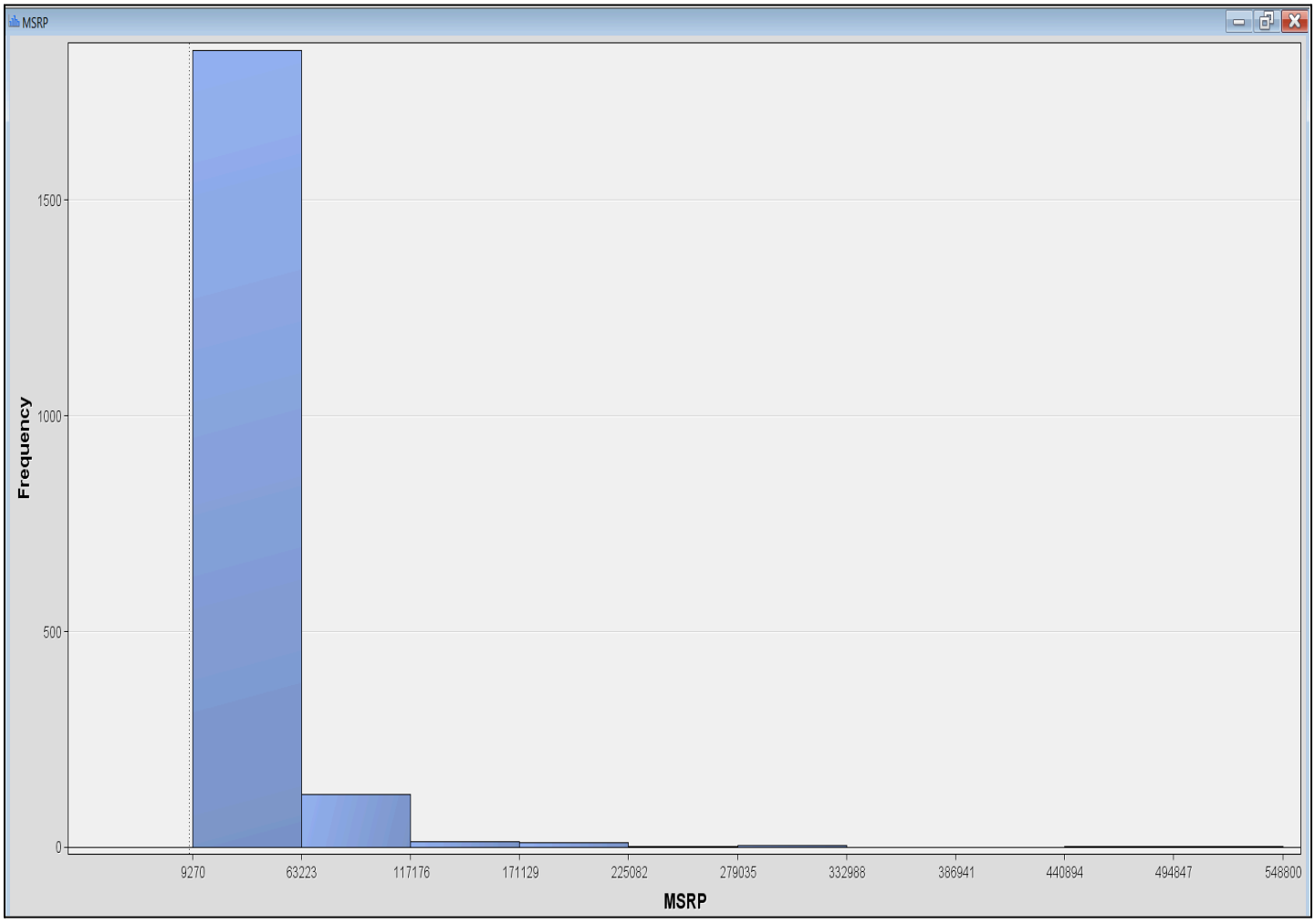
	Train	Validation	Test
Model Description	Root Average Squared Error	Root Average Squared Error	Root Average Squared Error
Gradient Boost 2	9102.13	8949.81	9969.17

Gradient Boost 1	9256.42	9126.31	10031.60
Decision Tree	9149.83	9454.17	9944.20
Decision Tree 2	9256.50	9872.81	10250.18
Linear Regression	10709.63	10723.64	10981.43

In our early attempts to develop a predictive model for car price prediction, we have tested and compared three supervised learning algorithms (figure -6). Our preliminary assessment is that the Decision Tree model gives the best performance and therefore we will use it to develop our final predictive model. We will apply optimization techniques, specific to Decision Tree, to develop a more robust predictive model.

Limitations:

Figure -6: Frequency of car MSRP



In all of the three models, the mean squared error and subsequently root average squared error metrics are large. Upon closer inspection, we found that all of our models were better at predicting car prices less than $64,000. The model predictions become poor with increasing car price. The reason for this could be that most of the cars in our dataset (Figure -6, frequency 29,580) are less than $64,000 and thus they have more influence during

training. Compared to that, there are only 2737 cars that are priced over $64,000. There are not enough observations of expensive cars which could affect a model's ability to learn specific patterns. Thus, our models' error for price prediction in expensive cars is high which could contribute to high values of error metrics. To circumvent this limitation, it may be reasonable to exclude cars that are over $64,000 and create a separate model for them.

We have not performed K-Fold cross validation on our dataset. We will be using this at a later stage in our project.