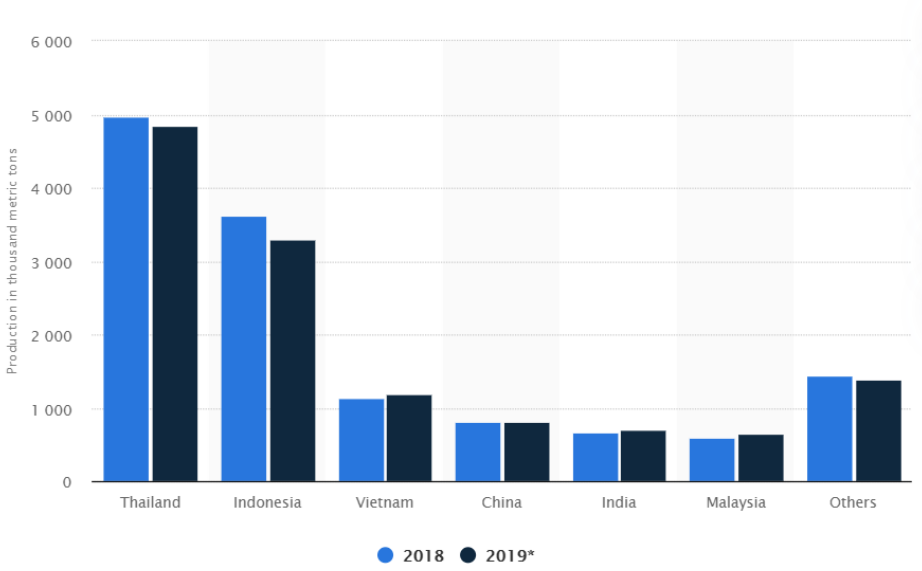
Introduction

Natural rubber comes from the Pará rubber tree, or the sharinga tree, commonly referred to as simply the rubber tree. It has many uses due to being highly waterproof, resilient, and stretchy.

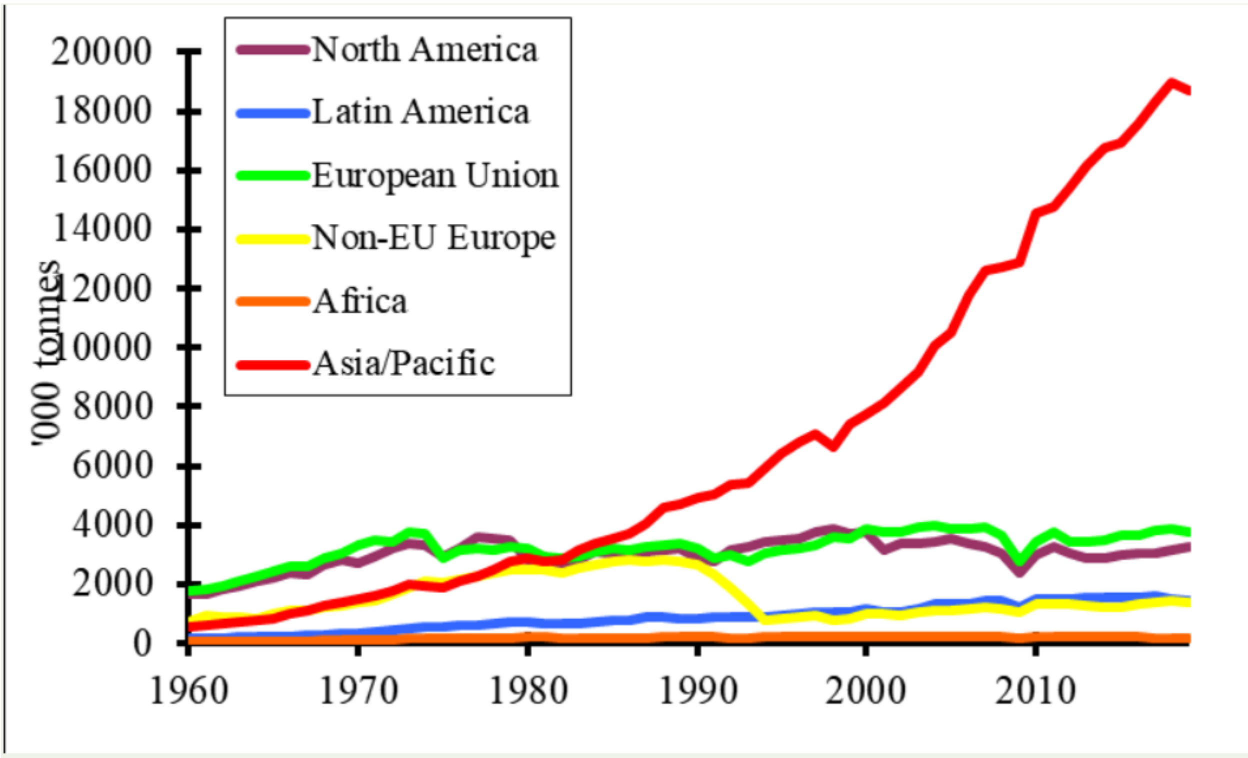
Ribbed Smoked Sheets (RSS) are coagulated rubber sheets processed from fresh field latex sourced from well managed rubber plantations adopting modern processing methods. The grades available are shown in the table. The higher grades RSS 1x to RSS 3 are mainly used for manufacture of products for medical, pharmaceutical and engineering. The lower grades of RSS 4 and 5 are generally used for the manufacture of automobile tyres, re-treading materials and all other general products. RSS 3 and RSS 4 are the preferred raw material for radial tyres. Quality of Ribbed Smoked Sheets is ascertained as laid down in Green Book Standards.

TSR which is also known as block rubber is graded according to precise technical parameters such as dirt content, ash content, nitrogen content, volatile matter and properties of the rubber such as its Wallace Plasticity (PO) and its Plasticity Retention Index (PRI). The TSR grades most widely used by the tyre and rubber industry are the TSR-20 and TSR-10 grades from Indonesia, Thailand and Malaysia which are known as SIR20, STR20 and SMR20 respectively. Block rubber can be produced both from field latex as well as from latex coagulum or what is commonly known as cup- lump. Tree lace and unsmoked sheets can also be used in producing block rubber.

Rubber production involves raw materials such as butadiene, crude oil, rubber cup lump, latex, etc. It also depends on external phenomena such as rainfall. Thailand is the world's largest rubber producer, producing around 4.85 million metric tons each year. Indonesia is the second largest producer with around 3.3 million metric tons produced yearly.



Accordingly, the global consumption of natural rubber is considerable. In 1990, natural rubber consumption amounted to 5.2 million metric tons, and in 2019, it reached 13.6 million metric tons, which is nearly tripled consumption in 28 years. China is by far the largest consumer of natural rubber worldwide, consuming a peak of 5.5 million metric tons in 2019. China uses natural rubber for a variety of manufacturing uses, including automobile and tire manufacturing, in particular.



Regional rubber consumption, 1960-2019

Rubber markets function in a systematic way, where farmers sell the raw products to local suppliers who in turn sell them to the larger suppliers. The larger suppliers then directly deal with the companies who utilize the products. This whole process can take up to several months, which can cause the current rubber price to change on the basis of historical value of the factors. To deal with this discrepancy we have utilized lag variables.

Rubber prices are affected both positively and negatively by different factors. An increase in the crude oil price will lead to an increase in the price of natural rubber, but an increase in the demand of synthetic rubber might lead to a decrease in the price of natural rubber. We have tried to construct the model taking in account multiple such relationships.

Rubber is one of the most widely traded commodities in the world. The global rubber market stood at 40.77 billion dollars in 2019. It is estimated to cross 50 billion dollars by 2027. This immense market size has led to the formation of several alternate trading markets. These markets include options, futures, etc and can be extremely difficult to comprehend. Accurate forecasting of the price of different types of rubber makes navigating through these markets easier. The forecasted trends can be used to make decisions on investing or shorting the various derivatives.

We have tried to implement a program to forecast the future prices of TSR20 and RSS4 rubber classes. For the forecast we have used different types of statistical models based on regression. The feature selection for these models was done on the basis of research on the independent variables and statistical significance. The forecast for the independent variables was done using ARIMA and SARIMA models. The forecast on the basis of the predicted independent variables was done using XGBoost, LightGBM and RandomForestRegressor models which were trained on monthly historical data from January 2015 to December 2020. The evaluation of the models was done on the basis of their MAPE and R2 score.

Data Pre-processing

We use feature engineering to construct all of the inputs that will be used to make predictions for future time steps.

After doing the market research, we performed many different types of transforms like lags, ratios, difference and rolling means on our dataset and created new input features which might give good correlation with our output variables.

We performed transformations by taking the ratio and the product of the desirable features to produce new features which helped us to train our model better

ROLLING MEAN

The rolling average or moving average is the simple mean of the last ‘n’ values. It can help us in finding trends that would be otherwise hard to detect. Also, they can be used to determine long-term trends. You can simply calculate the rolling average by summing up the previous ‘n’ values and dividing them by ‘n’ itself. But for this, the first (n-1) values of the rolling average would be Nan.

The value of n which we have selected for rolling mean om our features is 3.

LAG

The lag operator (also known as backshift operator) is a function that shifts (offsets) a time series such that the “lagged” values are aligned with the actual time series. The lags can be shifted any number of units, which simply controls the length of the backshift.

A dependent variable that is lagged in time. For example, if Y_t is the dependent variable, then Y_{t-1} will be a lagged dependent variable with a lag of one period. Lagged values are used in Dynamic Regression modeling.

Lags are very useful in time series analysis because of a phenomenon called autocorrelation, which is a tendency for the values within a time series to be correlated with previous copies of itself. One benefit to autocorrelation is that we can identify patterns within the time series, which helps in determining seasonality, the tendency for patterns to repeat at periodic frequencies. Understanding how to calculate lags and analyze autocorrelation will be the focus of this post.

ACF

ACF is an (complete) auto-correlation function which gives us values of auto-correlation of any series with its lagged values. We plot these values along with the confidence band and tada! We have an ACF plot. In simple terms, it describes how well the present value of the series is related with its past values. A time series can have components like trend, seasonality, cyclic and residual. ACF considers all these components while finding correlations hence it's a 'complete auto-correlation plot'.

Auto Regressive (AR) process

A time series is said to be AR when present value of the time series can be obtained using previous values of the same time series i.e., the present value is weighted average of its past values. Stock prices and global temperature rise can be thought of as an AR processes.

The AR process of an order p can be written as,

Where ϵt is a white noise and $y't_{-1}$ and $y't_{-2}$ are the lags. Order p is the lag value after which PACF plot crosses the upper confidence interval for the first time. These p lags will act as our features while forecasting the AR time series. We cannot use the ACF plot here because it will show good correlations even for the lags which are far in the past. If we consider those many features, we will have multicollinearity issues. This is not a problem with PACF plot as it removes components already explained by earlier lags, so we only get the lags which have the correlation with the residual i.e., the component not explained by earlier lags.

PACF

PACF is a partial auto-correlation function. Basically, instead of finding correlations of present with lags like ACF, it finds correlation of the residuals (which remains after removing the effects which are already explained by the earlier lag(s)) with the next lag value hence 'partial' and not ‘complete’ as we remove already found variations before we find the next correlation. So, if there is any hidden information in the residual which can be modeled by the next lag, we might get a good correlation and we will keep that next lag as a feature while modeling. Remember while modeling we don’t want to keep too many features which are correlated as that can create multicollinearity issues. Hence, we need to retain only the relevant features.

Moving Average (MA) process

A Moving Average process is one where the present value of series is defined as a linear combination of past errors. We assume the errors to be independently distributed with the normal distribution. The MA process of order q is defined as,

Here ϵ_t is a white noise. To get intuition of MA process let us consider order 1 MA process which will look like,

Let’s consider y’t as the crude oil price and ϵ_t is the change in the oil price due to hurricane. Assume that c=10 (mean value of crude oil price when there is no hurricane) and $\theta_1=0.5$. Suppose, there is a hurricane today and it was not present yesterday, so y’t will be 15 assuming the change in the oil price due to hurricane as $\epsilon_t=5$. Tomorrow there is no hurricane so y’t will be 12.5 as $\epsilon_t=0$ and $\epsilon_{t-1}=5$. Suppose there is no hurricane day after tomorrow. In that case the oil price would be 10 which means it got stabilized back to mean after getting varied by hurricane. So, the effect of hurricane only stays for one lagged value in our case. Hurricane in this case is an independent phenomenon.

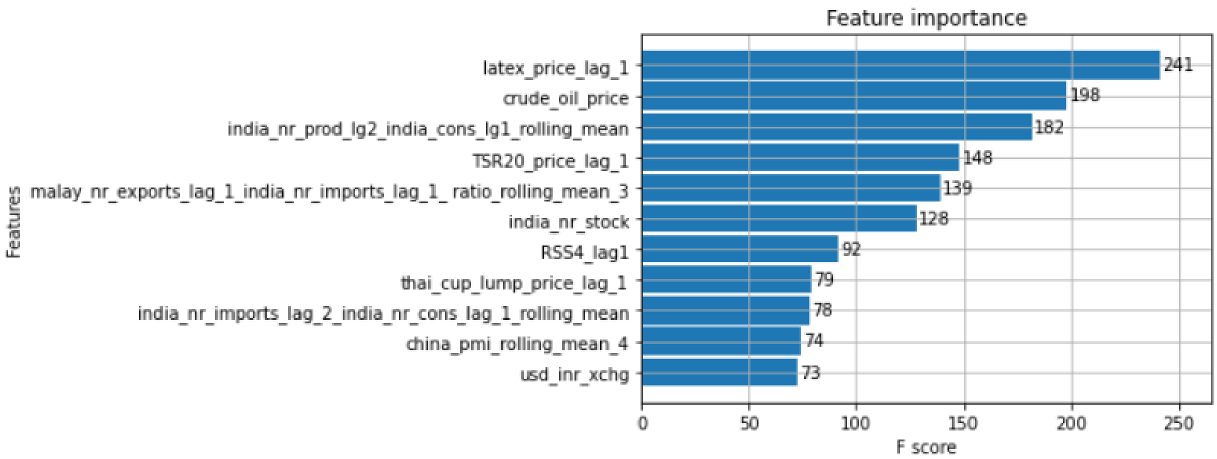
Order q of the MA process is obtained from the ACF plot, this is the lag after which ACF crosses the upper confidence interval for the first time. As we know PACF captures correlations of residuals and the time series lags, we might get good correlations for nearest lags as well as for past lags. Since our series is linear combination of the residuals and none of time series own lag can directly explain its present (since its not an AR), which is the essence of PACF plot as it subtracts variations already explained by earlier lags, its kind of PACF losing its power here! On the other hand being a MA process, it doesn’t have the seasonal or trend components so the ACF plot will capture the correlations due residual components only.One can also think of it as ACF which is a complete plot (capturing trend, seasonality, cyclic and residual correlations) acting as a partial plot since we don’t have trends, seasons, etc.

Hence with the help of ACF and PACF plots we can decide the values of parameters p and q of the ARIMA/SARIMA models which in turn helps us to predict the independent variables.

Feature Selection

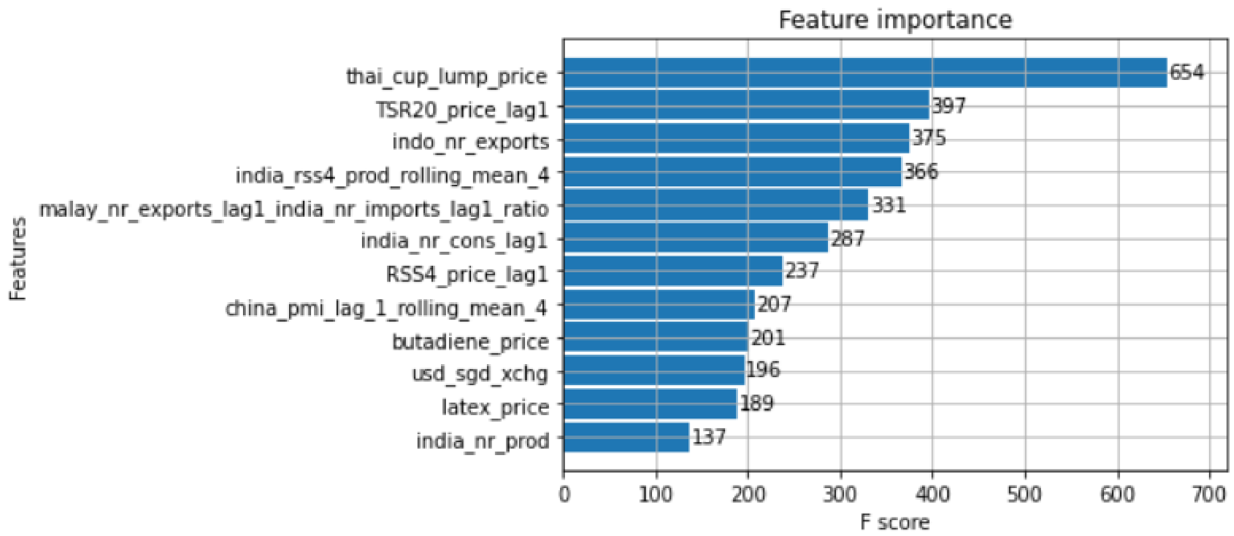
RSS4

Best Transformations	**Correlation with RSS4 price**
latex_price_lag_1	0.685277
india_nr_prod_lg2_india_cons_lg1_rolling _mean	-0.233392
crude_oil_price	0.303067
india_nr_stock	-0.271176
malay_nr_exports_lag_1_india_nr_imports_lag_1_ratio_rolling_mean_3	0.436270
usd_inr_xchg	-0.405750
india_nr_imports_lag_2_india_nr_cons_lag_1_rolling_mean	-0.381684
usd_sgd_xchg	-0.317201
RSS4_lag1	
TSR20_price_lag_1	
thai_cup_lump_price_lag_1	0.658268
china_pmi_rolling_mean_4	0.495750
idr_usd_xchg	0.440211



TSR20

Best Transformations	Correlation
thai_cup_lump_price	0.907024
TSR20_price_lag1	0.874369
latex_price	0.798539
RSS4_price_lag1	0.509689
butadiene_price	0.550838
malay_nr_exports_lag1_india_nr_imports_lag1_ratio	0.501015
china_pmi_lag_1_rolling_mean_4	0.482397
indo_nr_exports	0.448951
india_rss4_prod_rolling_mean_4	0.300590
india_nr_cons_lag1	
india_nr_prod	0.290757
usd_sgd_xchg	



Independent Variable Models

ARIMA

The autoregressive-integrated-moving average (ARIMA) model is discussed in detail in Box and Jenkins (1976) and O'Donovan (1983). Briefly, this technique is a univariate approach which is built on the premise that knowledge of past values of a time series is sufficient to make forecasts of the variable in question. Box and Jenkins (1976) set four steps for this approach: model identification, parameter estimation, diagnostic checking and forecasting.

The identification step involves the comparison of estimated autocorrelation and partial autocorrelation functions of known ARIMA processes.

An ARIMA model is characterized by 3 terms: p, d, q where,

p is the order of the AR term (number of autoregressive terms),

d is the number of differencing required to make the time series stationary,

q is the order of the MA term (number of lagged forecast errors in the prediction equation.),

The forecasting equation is constructed as follows. First, let y denote the dth difference of Y, which means:

If d=0: $y_t = Y_t$

If d=1: $y_t = Y_t – Y_{t-1}$

If d=2: $y_t = (Y_t – Y_{t-1}) - (Y_{t-1} – Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$

Note that the second difference of Y (the d=2 case) is not the difference from 2 periods ago. Rather, it is the first-difference-of-the-first difference, which is the discrete analog of a second derivative, i.e., the local acceleration of the series rather than its local trend.

In terms of y, the general forecasting equation is:

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \cdots - \theta_q e_{t-q}$$

Here the moving average parameters (θ's) are defined so that their signs are negative in the equation, following the convention introduced by Box and Jenkins.

We have chosen the values of p, d, q by using all possible combinations of p, d, q and then calculating the RMSE value of the mode. Then selecting the best model out of all these by checking for lowest RMSE score

Arima models have been used to predict the values of the independent variables of our model.

The independent variables for predicting the price of RSS4 are –

Independent Variable	Best Arima order	RMSE
Crude oil Price	(1,0,1)	6.241
Latex Price Lag 1	(1,0,0)	7.049

(India NR production Lag2 / India consumption Lag1) Rolling mean	(0,0,2)	0.413
(Malaysia NR Exports Lag1 / India NR Imports Lag1) Rolling mean	(1,0,2)	0.156
Indonesia NR Exports	(6,0,2)	28211.655
China NR Consumption Rolling Mean	(8,2,1)	15779.344
Thailand RSS Production	(0,1,0)	11948.398
USD INR Exchange Rate	(2,1,0)	1.009
(India NR Imports Lag2 / India NR Consumption Lag1) Rolling Mean	(0,1,0)	0.558
USD SGD Exchange Rate	(2,0,0)	0.012
TSR20 Price Lag1	(0,1,0)	7.562
Thai Cup Lump Price Lag1	(1,1,2)	2.434
China PMI Rolling Mean	(4,1,0)	1.030
World SR Imports	(2,0,1)	81551.853
IDR USD Exchange Rate	(0,0,1)	0.000

The independent variables for predicting the price of TSR20 are –

Independent Variable	Best Arima order	RMSE
Latex Price Lag 1	(1,0,0)	7.049
(Malaysia NR Exports Lag1 / India NR Imports Lag1) Rolling mean	(1,0,2)	0.156
Indonesia NR Exports	(6,0,2)	28211.655
USD SGD Exchange Rate	(2,0,0)	0.012

TSR20 Price Lag1	(0,1,0)	7.562
Thai Cup Lump Price Lag1	(1,1,2)	2.434
China PMI LAG1 Rolling Mean	(4,1,0)	1.030
RSS4 Price LAG1	(0,2,0)	71.305
Butadiene Price	(2,0,0)	2.358
India RSS4 production rolling mean	(2,0,0)	2411.469
India NR consumption LAG1	(0,0,1)	28795.976
India NR Production	(10,1,1)	4889.913

SARIMA

SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component.

A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA model. It adds three new hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

There are four seasonal elements that are not part of ARIMA that must be configured; they are:

- P: Seasonal autoregressive order.
- D: Seasonal difference order.
- Q: Seasonal moving average order.
- m: The number of time steps for a single seasonal period.

It is written as follows:

$$\underset{\substack{\uparrow \\ \text{Non-seasonal part of the model}}}{(p, d, q)} \qquad \underset{\substack{\uparrow \\ \text{Seasonal part of the model}}}{(P, D, Q)m}$$

We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model.

The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts of the seasonal period. For example, an ARIMA(1,1,1)(1,1,1)4 model (without a constant) is for quarterly data (m=4), and can be written as

(1 − φ1B) (1 − Φ1B4)(1 − B)(1 − B4)yt=(1+θ1B) (1+Θ1B4) εt.(1 − φ1B) (1 − Φ1 B4)(1 − B)(1 − B4)yt=(1+θ1B) (1+Θ1B4) εt.

The additional seasonal terms are simply multiplied by the non-seasonal terms.

The training data comprised of the values of the independent features from January 2015 up to September 2020. The testing data consisted of the values from October to December 2020.

We applied hyperparameter tuning for all the attributes of the SARIMA model (p,d,q,P,D,Q,m) on all the best features obtained by the process of feature engineering.

By analyzing the results of the above process, we fit the SARIMA Model with the parameter pairs which gave us the least MAPE for each feature separately.

Then, with the help of the model which was trained with the best fit parameters, predictions were made for a period of 3 months from January 2021 to March 2021 for each independent feature.

These final predictions then helped us to generate the prices of RSS4 and TSR20.

The independent variables for predicting the price of RSS4 are –

Independent Variable	Best SARIMA order	Best Seasonal SARIMA order	RMSE
Crude oil Price	(2,1,2)	(1,1,0,6)	12.111
Latex Price Lag 1	(0,1,0)	(0,1,2,6)	9.4.4
(India NR production Lag2 / India consumption Lag1) Rolling mean	(0,1,1)	(0,1,0,6)	0.600
(Malaysia NR Exports Lag1 / India NR Imports Lag1) Rolling mean	(0,2,2)	(2,1,2,6)	0.338

Indonesia NR Exports	(2,1,2)	(1,1,2,6)	25917.191
China NR Consumption Rolling Mean	(2,2,2)	(2,1,0,12)	40767.344
Thailand RSS Production	(1,2,2)	(1,1,0,6)	21727.494
USD INR Exchange Rate	(0,1,0)	(0,1,2,6)	1.77
(India NR Imports Lag2 / India NR Consumption Lag1) Rolling Mean	(0,1,1)	(1,2,0,12)	0.600
USD SGD Exchange Rate	(0,1,0)	(0,1,2,6)	1.77
TSR20 Price Lag1	(2,1,2)	(2,1,1,6)	157.562
Thai Cup Lump Price Lag1	(0,1,0)	(0,1,0,12)	5.774
China PMI Rolling Mean	(2,1,1)	(1,1,1,6)	1.32
World SR Imports	(0,1,0)	(2,2,1,6)	83689.853
IDR USD Exchange Rate	(0,1,0)	(0,1,2,6)	1.77

The independent variables for predicting the price of TSR20 are –

Independent Variable	Best SARIMA order	Best Seasonal SARIMA order	RMSE
Latex Price Lag 1	(1,1,2)	(0,2,0,6)	8.92
(Malaysia NR Exports Lag1 / India NR Imports Lag1) Rolling mean	(2,1,2)	(0,2,2,6)	0.096
Indonesia NR Exports	(2,2,0)	(0,1,1,6)	1566.39
USD SGD Exchange Rate	(2,2,0)	(2,1,1,12)	0.0008
TSR20 Price Lag1	(1,1,1)	(1,2,0,12)	43.012
Thai Cup Lump Price Lag1	(2,2,0)	(2,1,1,6)	1.4812
China PMI LAG1 Rolling Mean	(2,1,0)	(0,2,1,12)	0.02

RSS4 Price LAG1	(0,2,0)	(2,1,0,6)	33.6
Butadiene Price	(1,1,0)	(0,2,0,12)	3.040
India RSS4 production rolling mean	(0,2,2)	(0,1,2,6)	183.62
India NR consumption LAG1	(0,2,0)	(2,2,1,12)	6941
India NR Production	(10,1,1)		4889.913

Dependent Variable Models

The selected independent variables were then predicted by our ARIMA and SARIMA model. These predictions were fed into XGBoost, Random Forest and Light GBM models.

XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The same code runs on major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples.

RSS4

Training – 2015-2019; Testing – 2020

Date	RSS4_price	RSS4 Prediction	error	abs_error
2020-01-01	1896.472072	1823.401978	73.070094	73.070094
2020-02-01	1892.001454	1891.713257	0.288197	0.288197
2020-03-01	1751.976785	1710.326294	41.650491	41.650491
2020-04-01	1640.000000	1595.499512	44.500488	44.500488
2020-05-01	1527.447152	1504.826660	22.620492	22.620492
2020-06-01	1585.969064	1602.213135	-16.244070	16.244070
2020-07-01	1693.446700	1717.224487	-23.777787	23.777787
2020-08-01	1771.509902	1761.987549	9.522354	9.522354
2020-09-01	1825.754211	1848.591064	-22.836853	22.836853
2020-10-01	1932.114331	1742.590698	189.523633	189.523633
2020-11-01	2102.320370	2068.302734	34.017636	34.017636
2020-12-01	2150.980306	2096.719482	54.260824	54.260824

TSR20

Training – 2015-2019; Testing – 2020

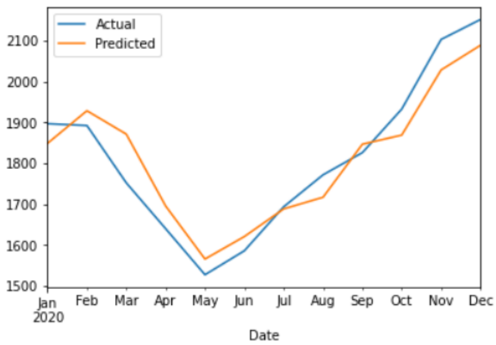
Date	TSR20_price	TSR20 Prediction	error	abs_error
2020-01-01	1465.500000	1455.562256	9.937744	9.937744
2020-02-01	1337.950000	1411.429688	-73.479687	73.479687
2020-03-01	1207.090909	1171.798706	35.292203	35.292203
2020-04-01	1087.666667	1101.009766	-13.343099	13.343099
2020-05-01	1091.222222	1121.209961	-29.987739	29.987739
2020-06-01	1141.227273	1200.547363	-59.320091	59.320091
2020-07-01	1177.142857	1178.913330	-1.770473	1.770473
2020-08-01	1304.500000	1282.015503	22.484497	22.484497
2020-09-01	1360.545455	1347.924316	12.621138	12.621138
2020-10-01	1523.727273	1359.643555	164.083718	164.083718
2020-11-01	1552.666667	1476.337158	76.329508	76.329508
2020-12-01	1558.380952	1507.662109	50.718843	50.718843

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

RSS4

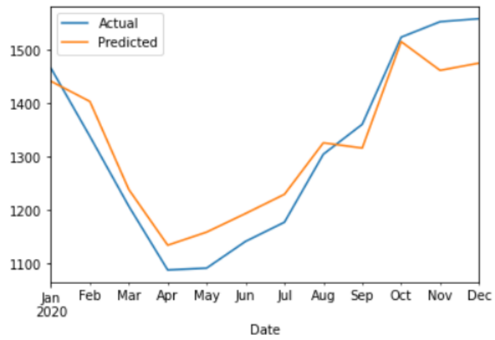
Training – 2015-2019; Testing – 2020



Date	Actual	Predicted	error	abs_error
2020-01-01	1896.472072	1848.574012	47.898060	47.898060
2020-02-01	1892.001454	1928.112005	-36.110551	36.110551
2020-03-01	1751.976785	1870.959822	-118.983037	118.983037
2020-04-01	1640.000000	1695.446447	-55.446447	55.446447
2020-05-01	1527.447152	1565.817231	-38.370079	38.370079
2020-06-01	1585.969064	1620.701726	-34.732662	34.732662
2020-07-01	1693.446700	1688.286275	5.160425	5.160425
2020-08-01	1771.509902	1716.724863	54.785039	54.785039
2020-09-01	1825.754211	1846.387937	-20.633726	20.633726
2020-10-01	1932.114331	1868.614136	63.500195	63.500195
2020-11-01	2102.320370	2027.733718	74.586652	74.586652
2020-12-01	2150.980306	2088.008818	62.971488	62.971488

TSR20

Training – 2015-2019; Testing – 2020



Date	Actual	Predicted	error	abs_error
2020-01-01	1465.500000	1441.156681	24.343319	24.343319
2020-02-01	1337.950000	1403.286050	-65.336050	65.336050
2020-03-01	1207.090909	1238.435895	-31.344986	31.344986
2020-04-01	1087.666667	1134.121519	-46.454852	46.454852
2020-05-01	1091.222222	1158.655863	-67.433641	67.433641
2020-06-01	1141.227273	1193.478760	-52.251488	52.251488
2020-07-01	1177.142857	1229.381721	-52.238864	52.238864
2020-08-01	1304.500000	1325.883611	-21.383611	21.383611
2020-09-01	1360.545455	1316.056669	44.488786	44.488786
2020-10-01	1523.727273	1515.397943	8.329330	8.329330
2020-11-01	1552.666667	1461.592045	91.074622	91.074622
2020-12-01	1558.380952	1475.197755	83.183198	83.183198

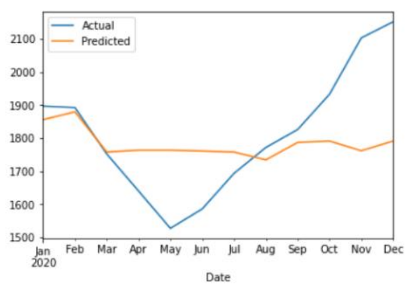
Light GBM

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following advantages:

- Faster training speed and higher efficiency.
- Lower memory usage.
- Better accuracy.
- Support of parallel, distributed, and GPU learning.
- Capable of handling large-scale data.

RSS4

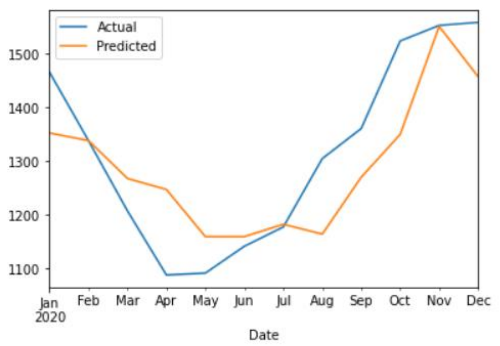
Training – 2015-2019; Testing – 2020



	Actual	Predicted	error	abs_error
Date				
2020-01-01	1896.472072	1855.409261	41.062811	41.062811
2020-02-01	1892.001454	1879.007782	12.993672	12.993672
2020-03-01	1751.976785	1757.642196	-5.665411	5.665411
2020-04-01	1640.000000	1763.345567	-123.345567	123.345567
2020-05-01	1527.447152	1763.345567	-235.898415	235.898415
2020-06-01	1585.969064	1760.676107	-174.707043	174.707043
2020-07-01	1693.446700	1757.642196	-64.195496	64.195496
2020-08-01	1771.509902	1734.043676	37.466226	37.466226
2020-09-01	1825.754211	1786.944087	38.810124	38.810124
2020-10-01	1932.114331	1790.942580	141.171751	141.171751
2020-11-01	2102.320370	1761.640689	340.679681	340.679681
2020-12-01	2150.980306	1790.942580	360.037726	360.037726

TSR20

Training – 2015-2019; Testing – 2020



	Actual	Predicted	error	abs_error
Date				
2020-01-01	1465.500000	1352.154207	113.345793	113.345793
2020-02-01	1337.950000	1338.127715	-0.177715	0.177715
2020-03-01	1207.090909	1267.529365	-60.438456	60.438456
2020-04-01	1087.666667	1246.924910	-159.258243	159.258243
2020-05-01	1091.222222	1159.111071	-67.888849	67.888849
2020-06-01	1141.227273	1159.111071	-17.883798	17.883798
2020-07-01	1177.142857	1182.219187	-5.076330	5.076330
2020-08-01	1304.500000	1163.898397	140.601603	140.601603
2020-09-01	1360.545455	1269.932076	90.613379	90.613379
2020-10-01	1523.727273	1349.647972	174.079300	174.079300
2020-11-01	1552.666667	1550.878290	1.788377	1.788377
2020-12-01	1558.380952	1457.092075	101.288878	101.288878

Results

Our work consists of predicting the prices of two different grades of rubber – RSS4 and TSR20. While RSS4 is the most widely used and produced variant of rubber in India, TSR20 is its counterpart in Thailand.

We built three different models for each variant using different variables for each. The three models used were XGBoost, Random Forest and Gradient Boosting Machine (GBM). To evaluate our models, we used two metrics – the Mean Absolute Percentage Error (MAPE) and the R2 score.

ARIMA and SARIMA models were implemented to project the values of features that were used to predict the prices. Their MAPE was calculated for a three-month period – October, November and December 2020 – and the models were then used to predict the features for the months of January, February and March 2021.

TSR20

Model	Type	(Oct/Nov/Dec) MAPE-VAL
XGBoost	ARIMA	6.16
	SARIMA	5.2
Random Forest	ARIMA	6.83
	SARIMA	5.69
GBM	ARIMA	3.65
	SARIMA	5.91

RSS4

Model	Type	(Oct/Nov/Dec) MAPE-VAL
XGBoost	ARIMA	6.71
	SARIMA	5.96
Random Forest	ARIMA	5.3
	SARIMA	3.18
GBM	ARIMA	11.78
	SARIMA	5.22

The XGBoost, Random Forest and GBM models were trained on data from 2015 to 2020 (excluded). Their MAPE and R2 score was calculated against the entire data from 2020. The predicted feature values were finally fed into the trained XGBoost, Random Forest and GBM models to get the final prices for January, February and March 2021.

TSR20

Model	(2020) MAPE-VAL	R2
XGBoost	3.36	0.865
Random Forest	3.53	0.896
GBM	6.035	0.675

RSS4

Model	(2020) MAPE-VAL	R2
XGBoost	2.38	0.875
Random Forest	2.799	0.9
GBM	7.119	0.079

The final prices calculated were as follows –

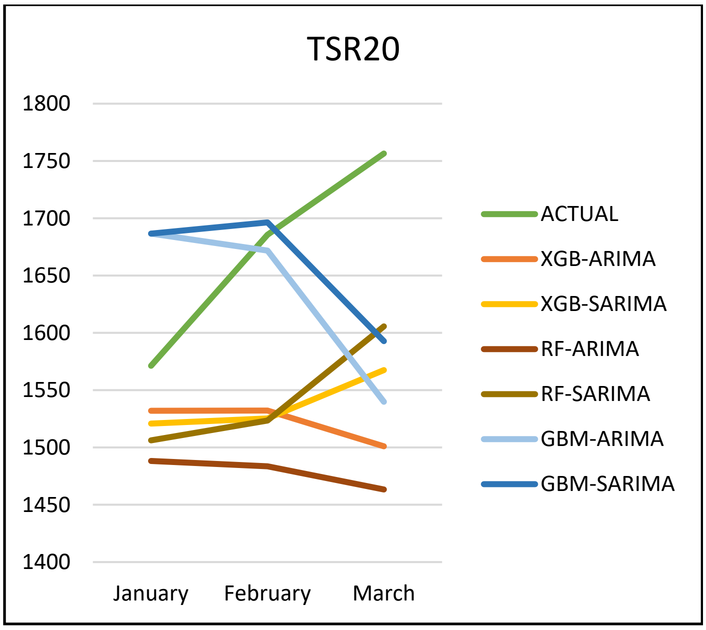
TSR20

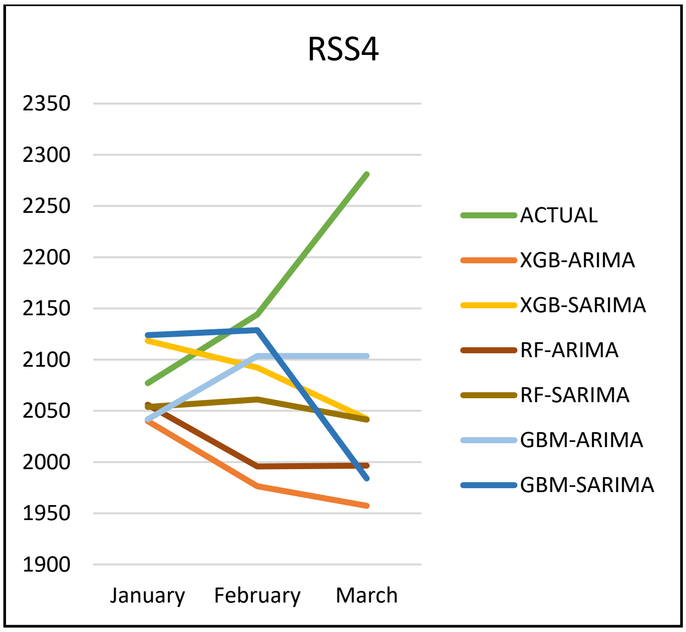
Model	Type	January	February	March
XGBoost	ARIMA	1532.073	1532.203	1501.022
	SARIMA	1520.855	1525.357	1567.521
Random Forest	ARIMA	1488.223	1483.523	1463.331
	SARIMA	1506.213	1523.556	1605.665
GBM	ARIMA	1686.645	1671.777	1539.861
	SARIMA	1686.645	1696.406	1592.796
	Actual Price	1571.3	1685.6	1756.5

RSS4

Model	Type	January	February	March
XGBoost	ARIMA	2039.968	1976.361	1957.244
	SARIMA	2118.404	2092.217	2042.166
Random Forest	ARIMA	2055.926	1995.693	1996.548
	SARIMA	2053.78	2061.051	2041.373
GBM	ARIMA	2041.688	2103.535	2103.535
	SARIMA	2123.909	2128.769	1983.932
	Actual Price	2077	2144	2281

The graphs below display the predictions of the models for the months of January-February-March 2021 against the actual RSS4 and TSR20 prices of the respective months. This helps us see how accurate our models are and assess how they will perform in the wild.





Using these metrics, we can conclude that for **TSR20** the best model is the **Random Forest-SARIMA** model. It gives the lowest MAPE and a relatively comparable R2 score as well. It is also able to predict the trend for the January-February-March data very well.

For **RSS4**, the best model is again the **Random Forest-SARIMA** model. It has an extremely high R2 score and a relatively low MAPE at the same time. However, here, it is unable to predict the trend for the price for January-February.

It is easy to see why the GBM-ARIMA and SARIMA models for both, TSR20 and RSS4 are not adequate. They have abysmal MAPE and R2 scores, and cannot predict the trend either. Hence, they rank last in terms of model quality.

Another important observation is that of understanding how far out do predictions remain accurate from the models created. The below tables show how the Average Absolute Errors accrue over each quarter of 2020 (since the model was trained on data till 2019).

TSR20 Models Errors Cross Tab

	Date (tsr!models) 2020			
	Q1	Q2	Q3	Q4
Avg. abs error GBM (..	57.99	81.68	78.76	92.39
Avg. abs error RF (ts..	36.51	53.12	30.62	65.11
Avg. abs error XGB (..	39.57	34.22	12.29	97.04

RSS4 Models Errors Cross Tab

	Date (rss!models) 2020			
	Q1	Q2	Q3	Q4
Avg. abs error GBM	19.9	178.0	46.8	280.6
Avg. abs error RF	67.7	42.8	26.9	67.0
Avg. abs error XGB	38.3	27.8	18.7	92.6

Conclusion

Based on the results of the presented analysis, we can conclude that the engineered statistical and machine learning models show promising results. They are comprehensive in outlook and factor in the environment surrounding the rubber industry in their calculations.

The **Random Forest-SARIMA** model created to predict TSR20 prices displays an **R2 score of 0.896** and a **mean absolute percentage error of 3.53**.

Similarly, the **Random Forest-SARIMA** model created to predict RSS4 prices displays an **R2 score of 0.9** and a **mean absolute percentage error of 2.799**.

This work can be taken forward by developing stronger models through further feature engineering. It can also be expanded to predict prices of other similar commodities in the global marketplace.