Consumer ABS (Name WIP)

James T. Bishop
Department of Computer Science
Boston University
Boston, MA 02215
jtbishop@bu.edu

Abstract

This paper explores statistical relationships in Consumer Asset Backed Securities including the pricing of individual securities and probability of prepayment or default. As with Mortgage-Backed Securities, the largest risk an investor in Consumer ABS faces is prepayment risk. Identifying relevant factors to accurately quantify this risk is paramount to extracting value from the Consumer ABS market.

Introduction

Asset Backed Securities (ABS) are a class of financial structured product that pool together cash flow generating assets and payout that cash to the securities' holders. Consumer ABS are a category of ABS where the underlying asset pools come from credit card loans, student loans, auto loans, or specific merchant loans like Affirm or Afterpay.

What makes Consumer ABS specifically interesting is how underdeveloped the market is, even relative to the rest of the ABS market or rest of the structured product market. Of the $9.2 trillion strucutured products market, non-mortgage ABS comprised $1.3 trillion, with consumer ABS making up about $500 billion of this .

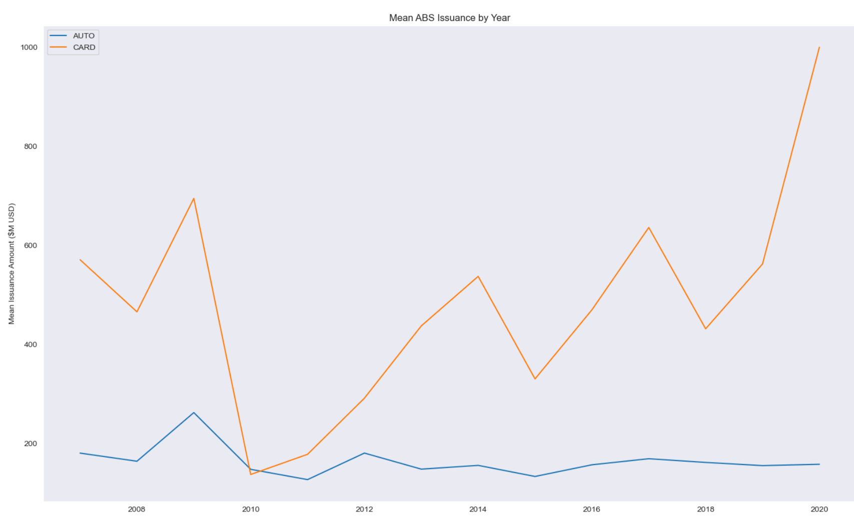

Figure 1: Mean Auto and Card ABS Issuance

Preprint. Under review.

Pricing these securities is difficult for a few reasons. First, while many of these securities trade frequently, there is less transparency than in equity and other fixed income markets. Many consumer ABS deals don't have public information available and might not even involve public companies, and even those that do aren't necessarily part of FINRA's Trade Reporting and Compliance Engine (TRACE), which functions to report trade data similar to what's available to the general public on stock trades. However, this data isn't typically available to retail investors because of a paywall.

Second, most retail investors can't trade consumer ABS through their broker. This means fewer market participants and, as a result, more mispricings.

Finally, ABS are complex products. Modeling their cash flows, prepayment and credit risk, and volatility is difficult, even if one assumes a perfectly efficient market. Due to this, there's a lower level of consistency in price determination between market makers (e.g. banks) which also makes price determination difficult.

The goal of this project is to act as a primer on using modern data science methods to explore the ABS market and make ballpark price estimations.

Data

Data was gathered through Boston University's Bloomberg Terminal subscription as well as the Federal Reserve of St. Louis (FRED) online database.

From Bloomberg, the SRCH <GO> function was used to screen for public ABS deals (i.e. ABS issuances involving publicly traded companies) transacted in US dollars and with collateral in the US. Later, the SRCH <GO> function was also used to find the TRACE eligible subset of those deals. Additionally, individual bond prices for 20 bonds was gathered along with collateral for those bonds using PDI<GO>. The pricing data was gathered on a daily frequency while the collateral data was gathered at monthly frequency.

To to so, I had to login remotely to a virtual machine in BU's library with Bloomberg Terminal then manually build a universe of securities which were then pulled into an Excel file which I then transferred from the VM to my machine. Throughout the process, I managed to exceed the subscription call limits which resulted in 1-2 day waiting periods before being able to pull more data. This was a reoccurring problem.

From FRED, historic yields for the 1, 2, 3, 5, 7, 10, 20, and 30 year treasuries were pulled on a monthly basis via Pandas datareader in Python as well was fredr in R.

For data maps, see .

Implementation

Initial Analysis & Clustering

To start, I explored the data gathered using SECF <GO> and examined year over year mean issuance by ABS category, distribution of issuances by coupon type within each category, and quantity and average dollar amount of issuances by coupon type. Next, I removed non-TRACE eligible bonds from my dataset since there would be no way for me to examine individual securities that are not TRACE eligible. I then created a correlation matrix of the remaining securities' features in Figure **??**. Following this, I used k-Means clustering with 10 clusters to find 10 general classes of TRACE eligible bonds, based on available features. From here, I identified the bond in each of the 10 clusters' center and randomly selected one other security from a cluster. This gave me a set of 20 TRACE eligible bonds that were most representative of each of the 10 clusters as well as a bond that was in the cluster but possibly less similar to the average bond in the cluster. I chose these bonds as the ones for which I would pull pricing and collateral data since there was no feasible way for me to fetch data on a massive bond universe given the data restrictions imposed by Boston University's Bloomberg subscription.

Classification

Following exploration, I made loader objects to load in my various data sources. These are all in the `preprocessing.py` file. I then began to look at ways to predict prepayments on the bonds in the TRACE Eligible dataset (see Table 2 for a description). I included all feature columns except "Is Mortgage Paid Off", "Next Call Date", "WAC", "Current WAL", and "Amt Out" because "Is Mortgage Paid Off" is our target variable, and the remaining variables' conditional distributions would reveal very clearly whether or not a security had been prepaid. I then used One Hot Encoding on the categorical features, made the date features Gregorian ordinal values, and used a label encoder on the remaining columns.

I trained kNN, SVM, Gaussian Process, Decision Tree, Random Forest, Multilayer Perceptron, AdaBoost, Naive Bayes, and Quadratic Discriminant classifiers on 70% of the data and tested them on 30%. I used 5 fold randomized cross validation on the training set to tune all of my models. Results will be discussed in .

Time Series Analysis

Following the classification task, I attempted to use Generalized Least Squares to predict monthly returns on the 20 securities mentioned earlier. I merged the price and collateral DataFrames with historic treasury yields. Since all of the collateral data was monthly, I calculated mean monthly returns for each security as well as monthly mean treasury curve yields and created a new dataframe that was monthly rather than daily data. I then lagged the exogenous variables by 1 timestep so that my model represented using the current month's data to predict the next month's. In doing so, I also limited the amount of simultaneous causality bias I might've had from my covariates, in particular treasury yields.

Finally, I moved over from Python to R to perform more time series analysis. This time, I only considered excess 3 and 6 month spreads, monthly pay rates, portfolio yield, change offs, and 30, 60 and 90+ day delinquency rates for consumer/card ABS and then WAC, WALA, and WAM for auto ABS. As I'll discuss later, the earlier time series analysis was riddled with multicollinearity issues so instead of using the vanilla exogenous regressors, I orthogonalized these using SVD. I trained these models on 60%, 70%, and 80% of the data then attempted to predict returns. Log returns are stationary, which made this task a bit simpler as I didn't have to worry about finding the appropriate number of lags in my data.

Results

Aside on Clustering

Following my 10 cluster set, I tried cluster sizes between 2 and 20 and decided 5 looked like an appropriate number. I made this decision because I found that a PCA with 5 components explained most of my data, which implied there would be a way to group the bonds into 5 clusters relatively well. Shown in 11 are the general composition of the 5 clusters into which I split my universe. The use case for this is possible risk management techniques when building a portfolio of ABS. That being said, the choice of 5 was more for simplicity than rigorous analysis. Based on silhouette score, 10 clusters performed best in that range and using the elbow method confirms this (Figure 2).

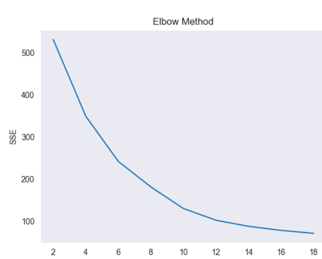

Figure 2: KMeans Elbow Method

Portfolios are often assessed using SVD or similar techniques to build an "orthogonal" portfolio. This will clearly increase diversification in the portfolio, but it's difficult to translate this into some sort of tangible qualitative description of the risks present, or not, in the portfolio. Clustering makes this a bit easier as, for instance, we know that AA rated auto ABS fall into separate clusters than BB rated card ABS. The value add here is that clusters are more naturally understood than matrix factorization or other more rigorous mathematical techniques, meaning pitching a portfolio or strategy that can be explained in terms of clusters may be more appealing to potential clients.

Classification

```
Test: RMSE of DecisionTreeClassifier(criterion='entropy', max_depth=35) is 0.139
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       266
           1       0.93      0.93      0.93        44

    accuracy                           0.98       310
   macro avg       0.96      0.96      0.96       310
weighted avg       0.98      0.98      0.98       310
```

Figure 3: Test Results from Decision Tree Classifier

In classification, I considered RMSE, accuracy, recall, and F1 Score and found that a Decision Tree with a max depth of 35 and splits using entropy/information gain performed best as seen in Figure 3.

This result is significant because of how well it performed. That being said, this task can be thought of as rather trivial given that it is classifying *ex post facto* rather than predicting if and when a future prepayment will occur. This is also a source of future work and is a much more complex task. I'll likely use Fannie and Freddie Loan Performance Data which is a larger and more robust dataset than the one used in this project since, unfortunately, there is no way for me to obtain the necessary type and quantity of data required to build a robust time dependent prepayment model for consumer ABS.

Time Series Analysis

The GLS model performed well on training data as seen in Figure 4 as well as Figures 12 and 13. The model was statistically significant at the 5% level, although few of the covariates were individually significant due to a high degree of multicollinearity.

The cause of the multicollinearity is rather obvious, given that the treasury curve generally moves together and the other exogenous variables typically have some sort of mathematical relationship, whether that be the fact that yield to maturity is used in part to calculate them or that delinquency rates are correlated.

The ARIMA models had varied performance as seen in 5, which shows the same 2 securities as in Figure 4. The predictions seemed to capture trends relatively well but performed poorly with volatility. This is likely due to the fact that there weren't enough regressors present to fully orthogonalize them, which meant that there was still some degree of multicollinearity. This can be seen in Figure 6, which shows the first and second principal components have strong negative correlation while there's slight positive correlation between components 1 and 4. To further analyze this, we can look at the component loadings in Figure 7.

Unfortunately, with the features available there was no workaround for this that improved prediction accuracy. Mean prediction error ranged from 45.89 to -1174.17 on the consumer dataframes while the auto dataframe predictions had a slightly tighter range between -115.41 and -405.32. Example components and loadings can be seen in Figures 8 and 9.

Some of the securities also had fewer observations than features which limited the orthogonalization's descriptive power.

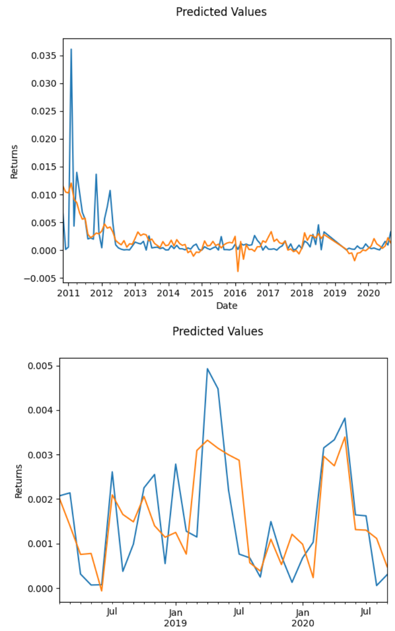

Figure 4: GLS Fitted Plots

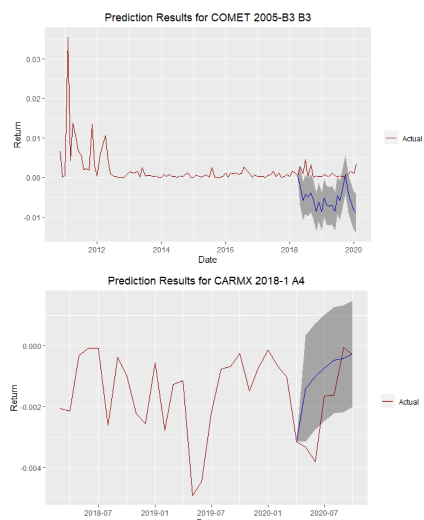

Figure 5: ARIMA Prediction Plots

Challenges

The biggest challenge in this project was data availability. Despite the cleanliness of data provided by Bloomberg, the amount of data with useful features was limited. It was impossible to pull a large universe of securities all at once, even in the screening phase which meant that there were approximately $1,000$ securities used in the classification task. Similarly, it was impossible to fetch a high volume of individual bond prices or collateral at once, and scripting was not an option which meant a small universe of securities for tasks using that data. This also meant large variance in the length of available time series which resulted in some securities being impossible to model.

Another evident challenge is the complexity of ABS. For instance, the LIBOR Market Model (LMM) is a popular tool for pricing structured products, among other things. However, calibrating the

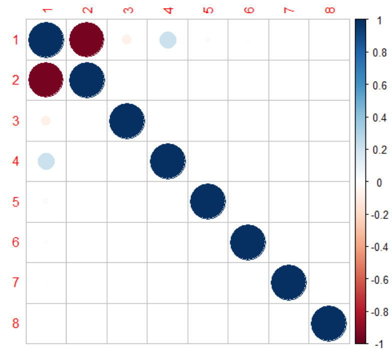

Figure 6: Principal Components for COMET 2005-B3 B3

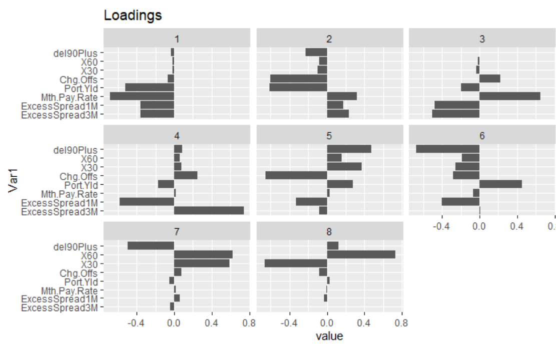

Figure 7: Factor Loadings for COMET 2005-B3 B3

model is a complex optimization process as is applying the model to data, particularly when the data available is low frequency and there isn't a lot available. I attempted to fit a similar model (G2++) but was unable to get the necessary data.

Finally, because of the way I had to pull data, there were some securities that had collateral and prices but weren't matchable my CUSIP in my main dataframe. This created an issue since I had to pull PDI information by CUSIP and price information by name and had no way to cross reference the two.

Conclusion & Takeaways

The largest takeaway is that a Decision Tree Classifier is very well suited to the task of predicting whether a security has defaulted or not, given this dataset. Obviously, one key issue with this is that we haven't predicted when or how likely that default is, which means that it's difficult to quantify the risk associated with it. This is discussed in greater detail in .

This paper also provides a starting point for time series analysis of ABS prices. By looking at a small number of factors, relatively accurate predictions were able to be made. To really see the performance of the GLS and ARIMA models, however, backtesting would be required, which wasn't possible with the data available. Even when observing proper caution with training, validation, and testing sets, financial models often succumb to issues of overfitting which is disasterous. This is a common pitfall of, for instance, LSTM models which typically perform exceptionally well during a research phase but fail to capture value in practice (on time series, at least). Backtesting would help better reveal potential downside risk associated with these models.

Future Work

I plan to gather incrementally more data and build a larger set of securities with collateral and pricing. In doing so, I'll be able to more produce results with a higher degree of certainty. With more

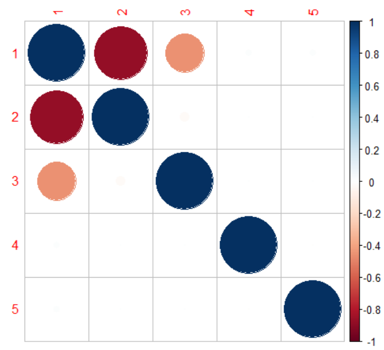

Figure 8: Principal Components for CARMX 2018-1 A4

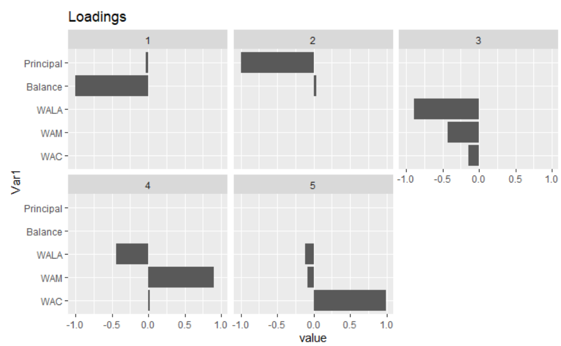

Figure 9: Factor Loadings for CARMX 2018-1 A4

individual securities, I'll be able to construct portfolios and test how well k-Means might work for balancing and examine the explainability/performance trade off associated with this method.

I also intend to examine price at issue data and gather text files of filing documents and news on companies issuing these debt securities to see whether or not sentiment analysis has a place in accurately pricing initial offerings.

As stated earlier, I'm going to work with the Fannie and Freddie loan performance dataset as well to see how powerful the models I built in this project are.

Further, I am working on implementing the LMM. The difficulty lies in the frequency of data I have available because typically the model is used on extremely high frequency data, e.g. 1 minute intervals rather than days or months. I also have to build a way to create a better yield curve. Ordinarily, it's possible to construct a yield curve from currently trading securities (e.g. your 4 year yield becomes whatever treasury matures in 4 years) but building this out backwards is difficult to do given data constraints.

Finally, the Decision Tree Classifier doesn't answer "How risky is this security?" or "When will we likeliest see prepayments occur?". To do so requires a more complex model, and is something I plan to explore in the future, hopefully using properties of the LMM model and the benefits of increased data volume.

Appendix

Table 1: ABS Trace Issuance Datasets

Label	Type	Description
Amt Out	float64	Amount outstanding (USD)
BBG Composite	object	Bloomberg Composite credit rating
CUSIP	object	
Cpn	float64	Coupon (%)
Current WAL	float64	Current Weighted Average Life
Day Count	object	Day count convention (30/360, 30/365, or Actual)
Delinquency Rate 60+ Days	float64	
Delinquency Rate 90+ Days	float64	
Issue Date	datetime64[ns]	
Issuer Name	object	
Maturity	datetime64[ns]	
Mid Price	float64	Mid price between bid and ask price
Mortgage Original Amount	int64	
Next Call Date	datetime64[ns]	Only applicable to callable bonds
Next Coupon Date	datetime64[ns]	
Original Maximum Loan Size	float64	Size of largest loan in the pool at issue
Price at Issue	float64	
Security Name	object	
Ticker	object	
Category	object	CARD, AUTO, or CONSUMER
Original WAL	float64	Original Weighted Average Life
isCallable	float64	

Table 2: TRACE Eligible Loans Dataset

Label	Type	Description
CUSIP	object	
Security Name	object	
Mortgage Original Amount	float64	
Cpn	float64	Coupon (%)
Current WAL	float64	
Amt Out	float64	Amount outstanding (USD)
BBG Composite	object	Bloomberg Composite credit rating
Day Count	object	Day count convention (30/360, 30/365, or Actual)
Delinquency Rate 60+ Days	float64	
Delinquency Rate 90+ Days	float64	
Is Mortgage Paid Off	int64	
Issue Date	datetime64[ns]	
Maturity	datetime64[ns]	
Next Call Date	datetime64[ns]	
Ticker	object	
Price at Issue	float64	
Benchmark Spread at Issue	float64	Spread to the benchmark rate at issue
PSA Since Issuance	float64	Public Securities Association prepayment calculation
WAC	float64	Weighted Average Coupon
Category	object	CARD, AUTO, or CONSUMER
isCallable	float64	
Original WAL	float64	Original Weighted Average Life

Table 3: Collateral for Credit Card ABS

Label	Type	Description
ExcessSpread3M	float64	
ExcessSpread1M	float64	
Mth Pay Rate	float64	
Port Yld	float64	
Chg Offs	float64	
30	float64	30 day delinquency rate
60	float64	60 day delinquency rate
del90Plus	float64	90 day delinquency rate

Table 4: Collateral for Auto and Consumer ABS

Label	Type	Description
WAC	float64	Weighted Average Coupon
WAM	int64	Weighted Average Maturity
WALA	int64	Weighted Average Loan Age
Balance	int64	
Principal	float64	

References

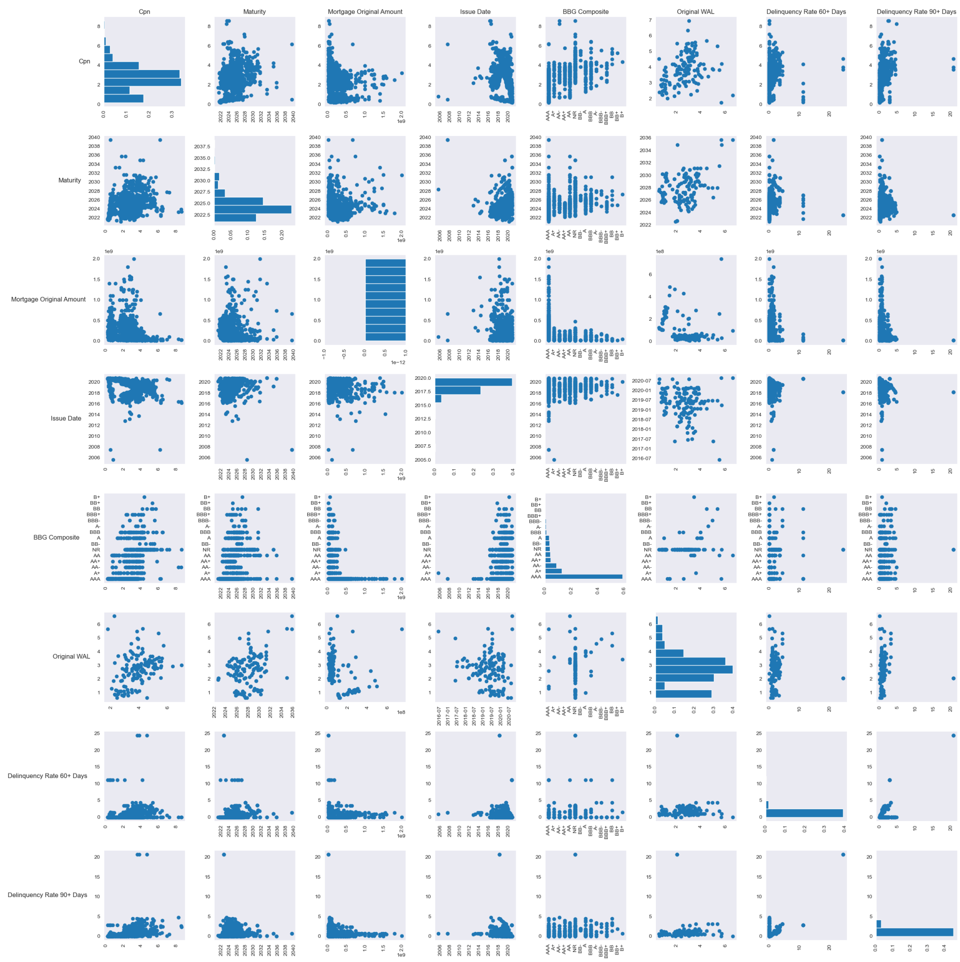

Figure 10: Correlation Matrix of TRACE Eligible Bonds

Figure 11: Attributes of 5 Clusters

Results for CARMX 2018-1 A4

GLS Regression Results

Dep. Variable:	Price	R-squared (uncentered):	0.829
Model:	GLS	Adj. R-squared (uncentered):	0.712
Method:	Least Squares	F-statistic:	7.076
Date:	Fri, 27 Nov 2020	Prob (F-statistic):	8.33e-05
Time:	09:44:51	Log-Likelihood:	179.65
No. Observations:	32	AIC:	-333.3
Df Residuals:	19	BIC:	-314.3
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
WAC	-0.0247	0.016	-1.564	0.134	-0.058	0.008
WAM	0.0027	0.001	2.775	0.012	0.001	0.005
WALA	0.0029	0.002	1.331	0.199	-0.002	0.008
Balance	1.9e-12	4.49e-11	0.042	0.967	-9.2e-11	9.58e-11
Principal	-8.48e-11	1.64e-10	-0.519	0.610	-4.27e-10	2.57e-10
DGS1	0.0020	0.011	0.178	0.861	-0.022	0.026
DGS2	-0.0182	0.055	-0.333	0.742	-0.133	0.096
DGS3	0.0095	0.069	0.137	0.892	-0.135	0.154
DGS5	0.0032	0.041	0.077	0.940	-0.083	0.089
DGS7	-0.0159	0.033	-0.483	0.635	-0.085	0.053
DGS10	0.0405	0.024	1.690	0.107	-0.010	0.091
DGS20	-0.0339	0.030	-1.139	0.269	-0.096	0.028
DGS30	0.0139	0.022	0.624	0.540	-0.033	0.060

Omnibus:	1.544	Durbin-Watson:	2.084
Prob(Omnibus):	0.462	Jarque-Bera (JB):	1.018
Skew:	-0.437	Prob(JB):	0.601
Kurtosis:	2.987	Cond. No.	4.15e+11

Figure 12: GLS Regression Results on CARMX

Results for AMXCA 2018-3 A

GLS Regression Results

Dep. Variable:	Price	R-squared (uncentered):	0.804
Model:	GLS	Adj. R-squared (uncentered):	0.544
Method:	Least Squares	F-statistic:	3.084
Date:	Fri, 27 Nov 2020	Prob (F-statistic):	0.0271
Time:	09:44:51	Log-Likelihood:	181.82
No. Observations:	28	AIC:	-331.6
Df Residuals:	12	BIC:	-310.3
Df Model:	16		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
ExcessSpread3M	-0.0004	0.001	-0.566	0.582	-0.002	0.001
ExcessSpread1M	3.192e-05	0.001	0.053	0.959	-0.001	0.001
Mth Pay Rate	2.215e-05	0.000	0.141	0.891	-0.000	0.000
Port Yld	0.0002	0.000	0.473	0.645	-0.001	0.001
Chg Offs	-7.907e-05	0.001	-0.061	0.952	-0.003	0.003
30	-0.0128	0.011	-1.206	0.251	-0.036	0.010
60	0.0036	0.016	0.229	0.822	-0.031	0.038
del90Plus	0.0063	0.008	0.825	0.426	-0.010	0.023
DGS1	0.0022	0.009	0.244	0.811	-0.018	0.022
DGS2	-0.0262	0.037	-0.701	0.497	-0.108	0.055
DGS3	0.0288	0.050	0.576	0.575	-0.080	0.138
DGS5	0.0130	0.027	0.476	0.643	-0.047	0.073
DGS7	-0.0217	0.031	-0.705	0.494	-0.089	0.045
DGS10	0.0002	0.021	0.009	0.993	-0.045	0.046
DGS20	-0.0015	0.019	-0.075	0.941	-0.044	0.041
DGS30	0.0054	0.021	0.262	0.798	-0.040	0.051

Omnibus:	14.666	Durbin-Watson:	2.219
Prob(Omnibus):	0.001	Jarque-Bera (JB):	26.520
Skew:	0.924	Prob(JB):	1.74e-06
Kurtosis:	7.395	Cond. No.	4.49e+04

Figure 13: GLS Regression Results on AMXCA

```
[1] "AMXCA 2018-3 A"
Series: y_train
Regression with ARIMA(1,0,0) errors

Coefficients:
          ar1  intercept  xreg1   xreg2  xreg3   xreg4   xreg5    xreg6
      -0.7130     0.0166  3e-04  -3e-04  7e-04  -3e-04  -2e-03  -0.0086
s.e.   0.1169     0.0021  0e+00   3e-04  3e-04   3e-04   3e-04   0.0029

sigma^2 estimated as 3.089e-07:  log likelihood=138.29
AIC=-258.59   AICc=-243.59   BIC=-248.77

Training set error measures:
                        ME        RMSE          MAE      MPE     MAPE      MASE       ACF1
Training set -2.375755e-05 0.000443363 0.0003364926 45.89149 481.6124 0.6016139 0.02863622
                        ME        RMSE          MAE      MPE     MAPE      MASE       ACF1
Training set -2.375755e-05 0.000443363 0.0003364926 45.89149 481.6124 0.6016139 0.02863622
[1] "CCCIT 2007-A3 A3"
Series: y_train
Regression with ARIMA(0,0,0) errors

Coefficients:
       xreg1  xreg2   xreg3   xreg4   xreg5    xreg6    xreg7    xreg8
      -1e-04  3e-04  -1e-04  -2e-04  0.0023  -0.0008  -0.0029  -0.0013
s.e.   1e-04  1e-04   2e-04   5e-04  0.0010   0.0018   0.0027   0.0036

sigma^2 estimated as 5.686e-06:  log likelihood=410.74
AIC=-803.47   AICc=-801.16   BIC=-781.18

Training set error measures:
                       ME        RMSE         MAE       MPE     MAPE      MASE         ACF1
Training set 8.274751e-06 0.002273604 0.001624946 -343.1032 375.2572 0.8692663 -0.003484608
                       ME        RMSE         MAE       MPE     MAPE      MASE         ACF1
Training set 8.274751e-06 0.002273604 0.001624946 -343.1032 375.2572 0.8692663 -0.003484608
[1] "CCCIT 2014-A5 A5"
Series: y_train
Regression with ARIMA(0,0,0) errors

Coefficients:
      intercept   xreg1  xreg2   xreg3   xreg4   xreg5  xreg6   xreg7    xreg8
        -0.0062  -2e-04  2e-04  -5e-04  -1e-04  -1e-04  1e-04  0.0009  -0.0072
s.e.     0.0032   2e-04  2e-04   3e-04   2e-04   4e-04  4e-04  0.0016   0.0019

sigma^2 estimated as 3.095e-07:  log likelihood=363.32
AIC=-706.64   AICc=-702.05   BIC=-685.86

Training set error measures:
                        ME        RMSE          MAE       MPE     MAPE      MASE        ACF1
Training set -3.159898e-16 0.000512163 0.0004021081 -191.8324 224.7446 0.7046651 -0.04026818
                        ME        RMSE          MAE       MPE     MAPE      MASE        ACF1
Training set -3.159898e-16 0.000512163 0.0004021081 -191.8324 224.7446 0.7046651 -0.04026818
[1] "COMET 2005-B3 B3"
Series: y_train
Regression with ARIMA(0,0,0) errors

Coefficients:
      intercept  xreg1  xreg2  xreg3    xreg4   xreg5    xreg6    xreg7    xreg8
         0.0388  9e-04  5e-04  9e-04  -0.0014  -7e-04  -0.0076  -0.0100  -0.0090
s.e.     0.0209  5e-04  9e-04  3e-04   0.0015   4e-03   0.0045   0.0075   0.0139

sigma^2 estimated as 1.434e-05:  log likelihood=378.91
AIC=-737.82   AICc=-735.04   BIC=-712.83

Training set error measures:
                       ME        RMSE         MAE  MPE MAPE      MASE       ACF1
Training set 3.847297e-18 0.003591922 0.001828966 -Inf  Inf 0.9626893 -0.1028105
```

```
[1] "AMXCA 2018-3 A"
Series: y_train
Regression with ARIMA(1,0,0) errors

Coefficients:
          ar1  intercept  xreg1   xreg2  xreg3   xreg4   xreg5    xreg6
      -0.7130     0.0166  3e-04  -3e-04  7e-04  -3e-04  -2e-03  -0.0086
s.e.   0.1169     0.0021  0e+00   3e-04  3e-04   3e-04   3e-04   0.0029

sigma^2 estimated as 3.089e-07:  log likelihood=138.29
AIC=-258.59   AICc=-243.59   BIC=-248.77

Training set error measures:
                        ME        RMSE          MAE      MPE     MAPE      MASE       ACF1
Training set -2.375755e-05 0.000443363 0.0003364926 45.89149 481.6124 0.6016139 0.02863622
[1] "CCCIT 2007-A3 A3"
Series: y_train
Regression with ARIMA(0,0,0) errors

Coefficients:
        xreg1  xreg2   xreg3   xreg4   xreg5    xreg6    xreg7    xreg8
      -1e-04  3e-04  -1e-04  -2e-04  0.0023  -0.0008  -0.0029  -0.0013
s.e.   1e-04  1e-04   2e-04   5e-04  0.0010   0.0018   0.0027   0.0036

sigma^2 estimated as 5.686e-06:  log likelihood=410.74
AIC=-803.47   AICc=-801.16   BIC=-781.18

Training set error measures:
                       ME        RMSE         MAE       MPE     MAPE      MASE         ACF1
Training set 8.274751e-06 0.002273604 0.001624946 -343.1032 375.2572 0.8692663 -0.003484608
[1] "CCCIT 2014-A5 A5"
Series: y_train
Regression with ARIMA(0,0,0) errors

Coefficients:
      intercept   xreg1  xreg2   xreg3   xreg4   xreg5  xreg6   xreg7    xreg8
        -0.0062  -2e-04  2e-04  -5e-04  -1e-04  -1e-04  1e-04  0.0009  -0.0072
s.e.     0.0032   2e-04  2e-04   3e-04   2e-04   4e-04  4e-04  0.0016   0.0019

sigma^2 estimated as 3.095e-07:  log likelihood=363.32
AIC=-706.64   AICc=-702.05   BIC=-685.86

Training set error measures:
                        ME        RMSE          MAE       MPE     MAPE      MASE        ACF1
Training set -3.159898e-16 0.000512163 0.0004021081 -191.8324 224.7446 0.7046651 -0.04026818
[1] "COMET 2005-B3 B3"
Series: y_train
Regression with ARIMA(0,0,0) errors

Coefficients:
      intercept  xreg1  xreg2  xreg3    xreg4   xreg5    xreg6    xreg7    xreg8
         0.0388  9e-04  5e-04  9e-04  -0.0014  -7e-04  -0.0076  -0.0100  -0.0090
s.e.     0.0209  5e-04  9e-04  3e-04   0.0015   4e-03   0.0045   0.0075   0.0139

sigma^2 estimated as 1.434e-05:  log likelihood=378.91
AIC=-737.82   AICc=-735.04   BIC=-712.83

Training set error measures:
                       ME        RMSE         MAE  MPE MAPE      MASE       ACF1
Training set 3.847297e-18 0.003591922 0.001828966 -Inf  Inf 0.9626893 -0.1028105
[1] "DCENT 2019-A1 A1"
Series: y_train
Regression with ARIMA(0,0,0) errors

Coefficients:
      xreg1   xreg2   xreg3
```

```
[1] "CARMX 2018-1 A4"
Series: y_train
Regression with ARIMA(0,0,0) errors

Coefficients:
      xreg1  xreg2  xreg3
      0e+00  0e+00  0e+00
s.e.  2e-04  2e-04  2e-04

sigma^2 estimated as 1.87e-06:  log likelihood=136.17
AIC=-264.34   AICc=-262.43   BIC=-259.31

Training set error measures:
                        ME        RMSE         MAE       MPE     MAPE      MASE      ACF1
Training set -4.356296e-07 0.001286029 0.001060353 -249.1545 281.9593 0.6410831 0.3112379
[1] "CARMX 2019-2 B"
Series: y_train
Regression with ARIMA(0,0,0) errors

Coefficients:
      intercept  xreg1  xreg2   xreg3
        -0.1277  0e+00  0e+00  0.0026
s.e.     0.2246  3e-04  3e-04  0.0042

sigma^2 estimated as 1.086e-05:  log likelihood=62.5
AIC=-115   AICc=-107.5   BIC=-111.81

Training set error measures:
                       ME        RMSE         MAE       MPE     MAPE      MASE       ACF1
Training set 1.294847e-17 0.002785408 0.002139661 -344.4364 375.4423 0.4799007 -0.1082409
[1] "FCAT 2019-1 D"
Series: y_train
Regression with ARIMA(0,0,0) errors

Coefficients:
      xreg1  xreg2   xreg3
      0e+00  0e+00  -2e-04
s.e.  3e-04  3e-04   3e-04

sigma^2 estimated as 8.022e-06:  log likelihood=63.96
AIC=-119.91   AICc=-115.47   BIC=-117.36

Training set error measures:
                       ME      RMSE         MAE       MPE     MAPE      MASE        ACF1
Training set 5.181069e-08 0.0025105 0.002060333 -191.3868 224.5557 0.7648521 -0.07307939
[1] "NAROT 2018-A A4"
Series: y_train
Regression with ARIMA(1,0,0) errors

Coefficients:
         ar1  intercept  xreg1  xreg2  xreg3
      0.5008     0.0531      0      0  1e-03
s.e.  0.0408     0.0290      0      0  7e-04

sigma^2 estimated as 1.023e-06:  log likelihood=139.58
AIC=-267.16   AICc=-262.49   BIC=-259.85

Training set error measures:
                        ME         RMSE          MAE       MPE     MAPE      MASE       ACF1
Training set -4.084858e-05 0.0009046631 0.0007513882 -217.7163 266.8406 0.6155783 -0.0573368
[1] "PART 2016-2A D"
Series: y_train
Regression with ARIMA(0,0,0) errors
```

```
Coefficients:
      intercept  xreg1  xreg2   xreg3
```