Main objective of the analysis that specifies whether your model will be focused on prediction or interpretation.

The main objective of the analysis is building a model that is able to predict the “average” price of an item.

Brief description of the data set you chose and a summary of its attributes.

Eve Online is a Massive Multiplayer Online Role Playing Game. Players are set in the far future and tasked with piloting spaceships to make money (ISK) and do whatever they desire. Players are able to come together to form corporations, then corporations of like minded players form alliances. Eve online has a completely player driver market, that means everything is created and destroyed within the game with very few external injections. The space in Eve online is divided into two main categories, High security space which is safe with an ingame police force and Null security space where outlaws roam free. Each of these groups are then further broken down into regions then constellations, then individual solar systems.

The data analyzed was all market data for January of 2022 in one specific region with the following attributes:

Numerical variables:

Volume: The amount that a specific item was traded in one region.

-Min: 1

-Max: 2,274,592,449.0

-Mean: 380,853.4

Order\_count: How many individual buy and sell requests were placed on the market combined.

-Min: 1

-Max: 1806.0

-Mean: 15.68

Highest: The highest price an item sold for

-Min: 0.01

-Max: 17,790,000,000.0

-Mean: 38,544,630.0

Lowest: The lowest price an item sold for

-Min: 0.01

-Max: 17,790,000,000.0

-Mean: 36,252,960.0

Average: The average price an item was sold for.

-Min: 0.01

-Max: 17,790,000,000.0

-Mean: 37,559,150.0

Date: the day for that type\_id

-From 1/1/2022 to 1/31/2022

Highlow\_difference: The difference between the highest and item sold for and the lowest.

-Min: 0.0

-Max: 2,281,500,000.0

-Mean: 2,291,673.0

Categorical variables:

Type\_id: The specific code associated with each unique item.

-Number of unique values: 9954

TypeName: The ingame name of each item correlating to the type\_id. I.E(Zydrine, Medium Vorton Projector, Ice Harvester Upgrade)

CategoryID: The specific code associated with each unique category.

-Number of unique values: 29

CategoryName: the classification name that encompases all group names. I.E(Module, Blueprint, Commodity).

GroupID: The specific code associated with each unique group.

-Number of unique values: 684

GroupName: The classification name that encompases all type names. I.E (Propulsion Module, Bounty Encrypted Bonds, Armor Plate).

The base dataset consisted of 150,599 variables with 14 features. I chose to keep both the ID and Names for each type/category/group for data exploration purposes and ease of reading but ended up dropping the IDs when it came to modeling.

Brief summary of data exploration and actions taken for data cleaning and feature engineering.

My first issue with the data is that none of my numerical features were normally distributed.

* volume 64.462185
* highlow\_difference 38.083912
* lowest 33.553670
* average 32.633370
* highest 31.897759
* order\_count 11.688379

In order to fix this I performed a log transformation on all skewed features resulting in:

* average -0.499953
* highest -0.542638
* lowest -0.448839
* order\_count 0.937300
* volume 1.640323
* order\_count 0.937300

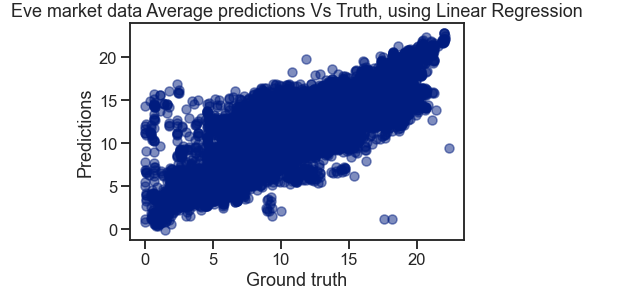
Then I had to deal with the issue of my categorical columns. I chose to use One Hot Encoding here, if I had kept my initial data I would have ended up with 4037 additional columns so I decided to work with a smaller portion of the data, only sales that occurred on the first of January and to eliminate the columns: typeName, categoryID, groupID, and type\_id as they were redundant bringing the total features down to 457. I also elected after the first model run to remove the columns of: Highest, Lowest, and Highlow\_difference as my model was using data that was directly derived from the “average” column and ended up making near perfect predictions.

Finally I ended up using some polynomial features in some of the models but I will cover that when we get to them.

Summary of training at least three linear regression models.

Linear Regression:

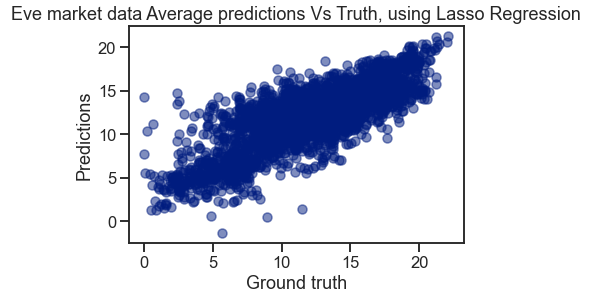
My baseline was Linear Regression with no polynomial features, no cross validation, no scaling, with one train/test split.



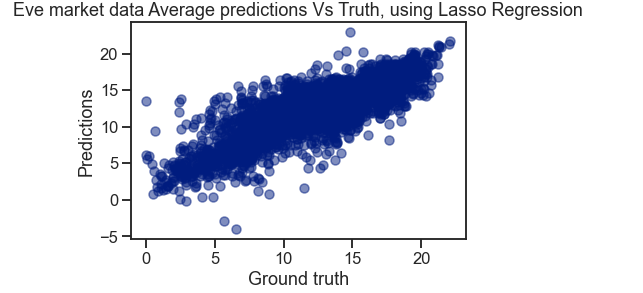
* R2: 0.7151
* RMSE: 2.235
* SME: 4.9963

Lasso Regression:

With Lasso we introduced multiple train/test splits using kFold, alpha modification over multiple runs to determine .022 was our best performing Alpha value before the addition of Polynomial features.



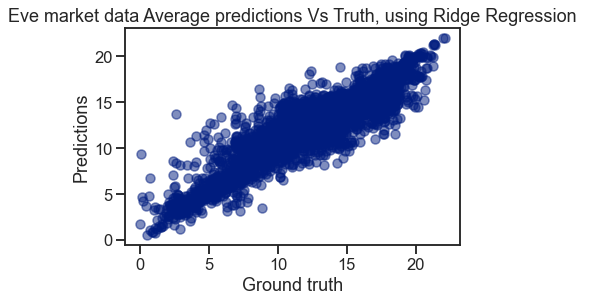
* R2: 0.6493
* RMSE: 3.3390
* SME: 11.1490

With the addition of two Polynomial features (unable to perform three due to computer limitations) we went to 107,416 features and achieved:

* R2: 0.6778
* RMSE: 2.4034
* SME: 5.7767

Ridge Regression:

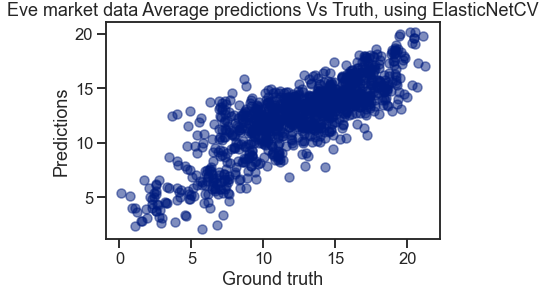
With Ridge we did the same steps as Lasso, we performed our log transformation, One Hot encoded our categorical features, tested multiple Polynomial features with multiple different Alpha values landing on two for our Polynomial Features and an Alpha value of 30.



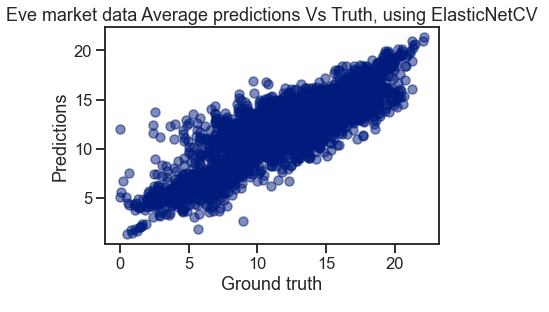
* R2: 0.8148
* RMSE: 1.8223
* SME: 3.3209

ElasticNet CV:

With ElasticNet we did the same steps as Ridge and Lasso, we performed our log transformation, and One Hot encoded our categorical features. This time we didn’t use our StandardScaler or Polynomial Features before running the model and landed on an Alpha value of: 0.01 and an l1\_ratio value of 0.1 and produced the following results.



* R2: 0.6729
* RMSE: 2.4248
* MSE: 5.8798

With the addition of StandardScaler and Polynomial Features degree 1 we achieved:

* R2: 0.7254
* RMSE: 2.2190
* MSE: 4.9242

A paragraph explaining which of your regressions you recommend as a final model that best fits your needs in terms of accuracy and explainability.

After running all models Ridge gets out final recommendation, the main objective of analysis was to get the best prediction for our average. With an R2 score at 0.81 I was satisfied with the model's ability to do that. We didn’t have a need for explainability so we disregarded that portion for all models.

Summary Key Findings and Insights, which walks your reader through the main drivers of your model and insights from your data derived from your linear regression model.

With so many categorical features and subfeatures within our data, I realized that the more we could create the better our model would predict. After that it was testing each model and seeing which would perform the best in a reasonable amount of time with the most features. One Hot Encoding allowed us to really enhance our models ability and then when we could do multiple polynomial features it got even better.

With our goal of accuracy over explability the process was simple, just find which model gave us the best R2 score, we didn’t care about which features caused the greatest impact just that we had the features in the model.

Finally the model performed better than expected, starting with only 14 features I didn’t believe we would have enough features to see an actionable pattern that a model could use. I believe with some fine tuning we could do even better.

Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model adding specific data features to achieve a better explanation or a better prediction.

The next big step for analyzing the data would be first to do some feature engineering to save space. That way we could include data from multiple days instead of one. Then if we could add more Polynomial Features to our elastic net I believe we it would be able to perform better, the limit of not being able to perform Polynomial Features at degree 2 I believe hurt model performance since we were only able to create 462 features instead of 107,416.

Also for practical applications I believe we need to change our prediction feature. Creating a new feature that better predicted the profit margin on an item would be more useful to the player than just being able to predict an average given some variables.