## Business Problem and Project Proposal

Helmet Headz would like to incorporate predictive analytics into their inventory management system in an attempt to optimize the amount of inventory stored in their warehouse. Having products sit in a warehouse for months on end increases costs to the company. It is far more effective to purchase more of the bike models that will sell faster and thus sit for less time in a warehouse. Also, if Helmet Headz has the correct amount of inventory in their warehouse, they can ship bikes to customers at a faster rate rather than having to backorder the product from the manufacturer. Correctly stocking bikes in the warehouse is undoubtedly a big advantage when competing against big box stores like Amazon and Dick's Sporting Goods.

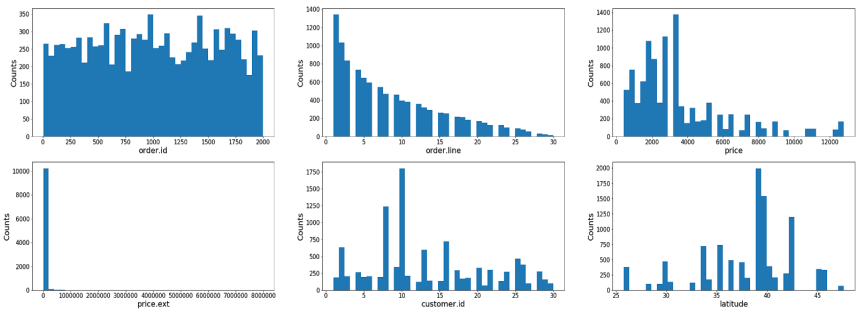
The goal of my project is to predict the amount of bikes that will be sold in the future by Helmet Headz. By using the bike sales data set from Helmet Headz sales database (Github), I will evaluate various data modeling techniques for the possible Helmet Headz projected sales volumes. If Helmet Headz has the ability to enter a bike model into the predictive data model identifying how many bikes are expected to sell before purchasing, this can save franchise owners a large amount of money and keep them competitive.

## Model Implementation

This is a supervised learning problem; the dataset contains sales volumes for the from 2011 to 2016. I will create a model that will be able to provide analytical guidance for the franchise owners based on the historical sales trends for each bike. To solve this business problem, I am going to leverage linear regression. Specifically, I will be putting my data through three different types of linear regression; Standard Linear Regression, Ridge Regression and Lasso Regression, in order to determine which modeling technique fits my data the best. I will perform k-fold cross validation for each model which takes a random sample of the data ensuring the best model is chosen for each feature set. My target variable will be quantity sold and my feature set will be the remaining variables from the data set. Although, some variables are removed in the feature reduction steps highlighted below.

In the beginning of this project, I went through various methods of exploring my data to get a better understanding of what was contained in the data set and ensured there were no errors. I then reviewed each variable to ensure there were no outliers and confirmed all of the data should be included in my future modeling. Also, to ensure I was not guilty of data snooping or identified false correlations in my data, I split my data into a test and training sets.

I then created histograms of all my numerical variables to understand how they were distributed and if the data could provide me any trends or interesting information. There were some interesting observations right out of the gate based on these histograms and allowed me to identify some fascinating trends. In addition to the numerical variables, I took a look at the categorical



variables to determine how many unique values they contained and which had a small number that would allow for informative analysis. Categorical variables with too many unique values are not as useful for modeling.

In order to run the most effective models, I performed various types of dimensionality and feature reduction of my training data set. First, I removed unnecessary columns based on my analysis from step one. After reviewing the data, correlations and relationships I determined that order.id, order.line, customer.id, latitude, longitude, order.date and model were not required in my data set. These variables do not give me any useful data or were redundant fields based on data from other variables (i.e. latitude and longitude are repetitive columns because I have collected city and state already in my data set).

To ensure my categorical variables would be able to be used in my linear regression modeling, I transformed them via one hot encoding. This one hot encoding massively enlarged my data set from 21 columns to 117 columns. This many columns makes my data set an ideal candidate for feature reduction to see if I could remove any redundancy/noise and allow for the modeling time to speed up. I performed multiple feature reduction strategies including Principal Component Analysis (PCA), Matrix Factorization, and Thresholding Numerical Feature Variance. For PCA, first I standardized the features so that they have the properties of a standard normal distribution with a mean of zero and a standard deviation of one. I then created the PCA with the n_components to 99% so that the variance of the original features would be retained. Whiten is set to true which transforms the values of each principal component so they have a zero mean and unit variance. I then ran the standardized features through the PCA to identify how many features I can reduce my data set by. The PCA stated I could reduce my features by a whopping 72!

The next dimensionality reduction strategy I chose to implement was the Matrix Factorization method. Matrix Factorization creates, fits and applies non-negative matrix factorization(NMF) to reduce the dimensionality of features.Since the best NMF results achieved are through trial and error, I wanted to test out multiple different types of NMF. I created multiple feature reduction sets with n_components set to 50,40,20,10 and 5. The final feature reduction strategy I tested was Thresholding Numerical Feature Variance. This strategy removed the features with a low variance, i.e. likely containing little information and thus causing unwanted noise in your modeling. I created a variance threshold of 0.5. The variance threshold calculates the variance for each feature, and drops any feature that does not have a threshold greater than 0.5.

## Modeling Results

I chose to go with a multiple linear regression model to predict the quantity of bike models that will be sold in the future. "Linear models make a prediction using a linear function of the input features"[1]. As mentioned above, I chose to run my features through 3 different types of linear regression; Standard Linear Regression, Ridge Linear Regression and Lasso Linear

[1] Boyle, T. (2020, February 7). Linear Regression Models in Python | Towards Data Science. Retrieved November 10, 2020, from https://towardsdatascience.com/linear-regression-models-4a3d14b8d368

regression. The two metrics I used to evaluate my models are $R^2$ and mean square error (MSE). $R^2$, or the coefficient of determination, is how much variance in the target variable that is explained by our model[1]. MSE is a measurement of the squared sum of all distances between predicted and true values. The higher the value of MSE, the greater the total squared error, and thus the worse the model[2].

| Model | Feature | R2_Score | RMSE_Score |
|---|---|---|---|
| **Linear_Regression** | NMF_50 | 0.6439 | -981.4728 |
| **Ridge_Regression** | NMF_50 | 0.6416 | -987.6563 |
| **Linear_Regression** | NMF_40 | 0.6377 | -998.5820 |
| **Lasso_Regression** | NMF_50 | 0.6375 | -998.7420 |
| **Ridge_Regression** | NMF_40 | 0.6360 | -1003.3517 |
| **Linear_Regression** | All_Features | 0.6357 | -1004.2251 |
| **Linear_Regression** | NMF_20 | 0.6350 | -1005.9889 |
| **Ridge_Regression** | All_Features | 0.6350 | -1006.1698 |
| **Lasso_Regression** | NMF_40 | 0.6349 | -1006.4297 |
| **Linear_Regression** | NMF_10 | 0.6345 | -1007.3926 |

Each regression type has their benefits. Standard Regression takes all the features as is and does not apply any penalty. Ridge regression uses L2 regularization to minimize the magnitude of the coefficients. It reduces the size of the coefficients and helps reduce model complexity[1]. Finally, Lasso regression uses L1 regularization to force some coefficients to be exactly zero. Lasso can be a good model choice when we have a large number of features but expect only a few to be important[1]. To ensure I was using the best model possible, I ran every feature set through all three models and calculated their $R^2$ and RMSE. To the right are the results from the top 10 models.

## Conclusion

After running all the feature sets through all three models, I determined my best model would be the standard linear regression model with the NMF 50 feature set based on the best $R^2$ and RMSE scores. I then ran the test data through the model and calculated the $R^2$ score to be 0.7. This indicates that 71% of the variance in the quantity of bike models sold can be determined by the features used in the linear regression model. This does not mean the model will predict down to the exact bike how many will be sold. However, a high $R^2$ value like 0.71 indicates this will be able to provide us with an accurate estimate that the franchise owners can use when purchasing inventory for their warehouses.

In the future, I would like to improve this model by incorporating more data and potentially weighing more recent data higher than data from 2011. Customer shopping trends change quickly, and data from 2011 may not be as helpful as data from six months ago. I would like to look at additional data points like marketing and sales promotions to improve the $R^2$ and RMSE scores for this model and provide better predictions.

---

[2] Albon, C. (2018). Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning(1st ed.). Sebastopol, CA: O'Reilly Media.