Tetelestai Guitars Product Listings Maximizing Price Analysis

Abstract:

The price of a product on Reverb.com is listed by Tetelestai Guitars for various types of products. The goal of this this report is to discover the contributing variables that reflect most strongly on the selling price for a product. Certain makes and brands, specific product categories such as Acoustic Guitar and Electric Guitar, along with the condition of the item and specific description topics were top variables in predicting the label of price, which is the desired outcome to maximize. Main factors explored in the dataset included shipping price, makes of the product, product category, text features (description), and a useful feature on Reverb.com known as “Make an Offer” to determine a best fit model to predict high price. Interestingly, some of the well-known brands and specific descriptors historically would have sold due to brand awareness, but the analysis found evidence to support specific product categories and descriptions rather than specific makes or brands. The analysis includes several portions: descriptive, diagnostic, predictive, and prescriptive analytics, focused on the CRISP-DM process in order to achieve desired results.

Author:

Preston Moore, pmoore2@my.gcu.edu

LAST UPDATED: Wednesday, February 5, 2025

Contents

Initial Data Collection Report

Data source

1. Transaction data: CSV export
    - Format: CSV
    - System: https://reverb.com/my/selling/orders
    - Location: Reverb.com
    - Authentication: anyone with Admin role, logged into their account.
    - Cost and future availability: would need to be made available through API Integration and Token use, but is only **private** sourced.

Data Description Report

Dictionary

- **condition**
    - The condition of the item being sold.
- **make**
    - the brand or make & model of the item being sold.
- **shipping_price**
    - The shipping price on the order of the item.
- **product_type**
    - This represents what category the product falls under.
- **offers_enabled**
    - Whether or not the customer could make an offer less than the asking price or not.
- **description**
    - A Text containing the description of the item listed on the website for public customer viewing. This can include URLs, embedded videos, custom descriptions, stock descriptions, or a combination of all these items. Descriptions can also be left blank.
- **price**
    - The label to predict, which is the selling price of the product.

Data Understanding: EDA, Univariate Analysis

Univariate Statistics

	Count	Unique	Type	Min	Max	25%	50%	75%	Mean	Median	Mode	Std	Skew	Kurt
condition	208	8	object	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
make	209	89	object	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
shipping_price	204	26	float64	0.0	100.0	0.0	8.0	30.0	16.71	8.0	0.0	20.3	1.48	2.03
product_type	209	11	object	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
offers_enabled	209	2	bool	NA	NA	NA	NA	NA	0.87	1.0	True	NA	NA	NA
description	208	208	object	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
price	209	95	float64	0.0	2000.0	40.0	100.0	300.0	222.7	100.0	40.0	300.82	2.97	11.59

Missing Values

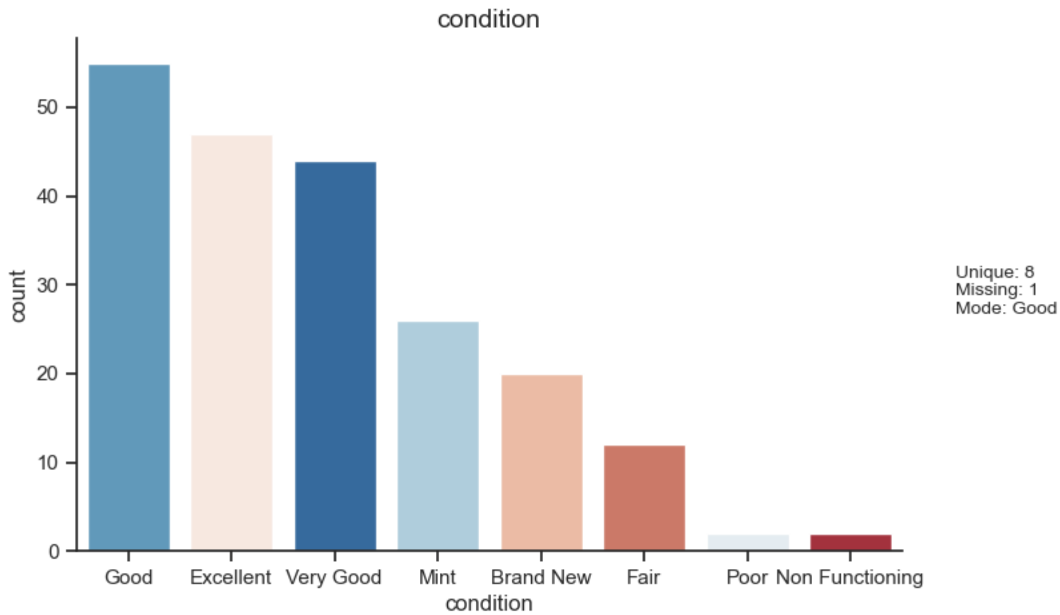
```
#   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   condition       208 non-null     object
 1   make            209 non-null     object
 2   shipping_price  204 non-null     float64
 3   product_type    209 non-null     object
 4   offers_enabled  209 non-null     bool
 5   description     208 non-null     object
 6   price           209 non-null     float64
```

Number of Missing Values by Column Name

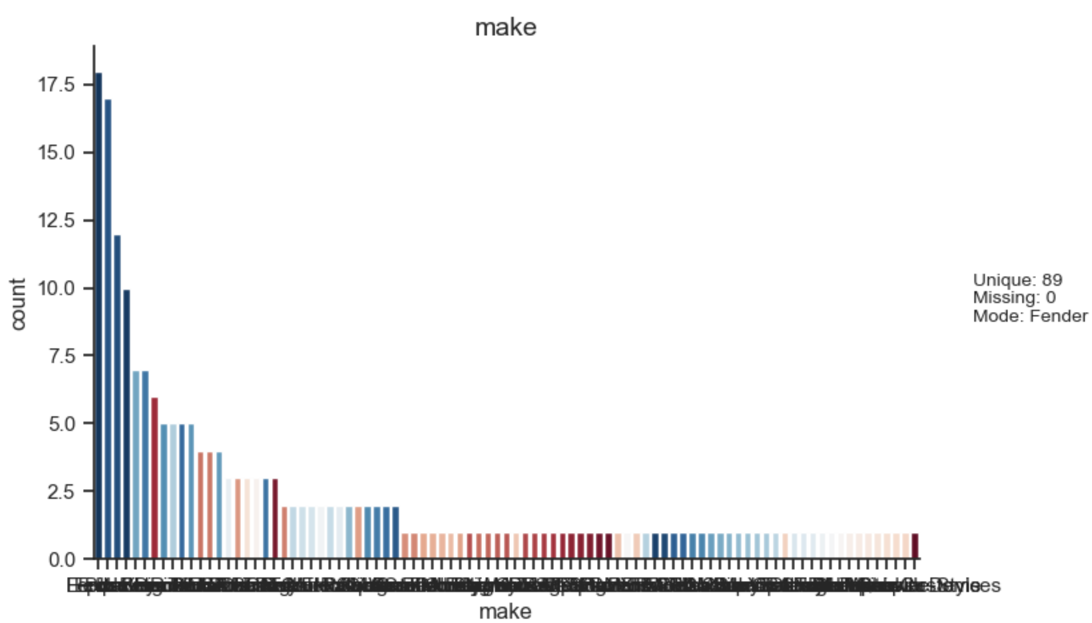
Variable/Column	# Missing Values
condition	1
make	0
shipping_price	5
product_type	0
offers_enabled	0
description	1
price	0

Data Understanding: EDA, Univariate Visualizations

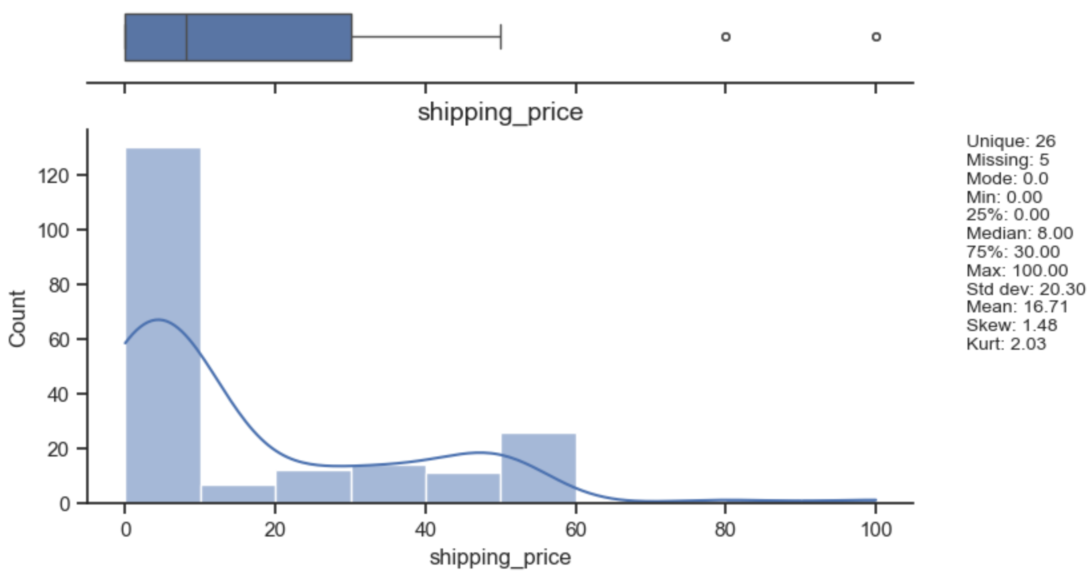
Condition



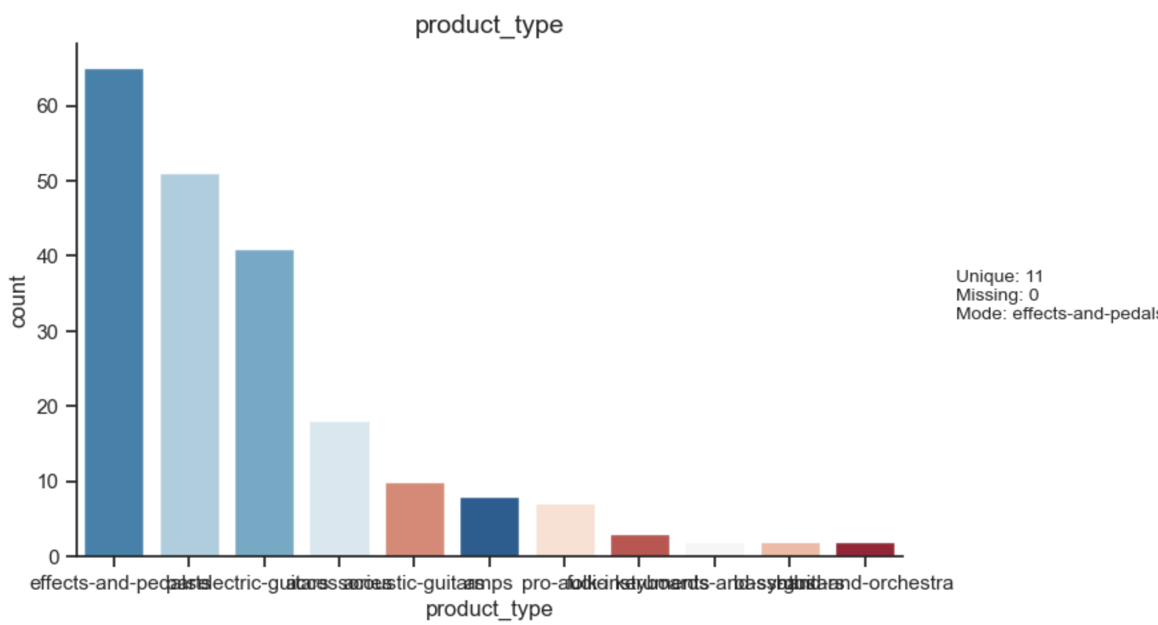
Make



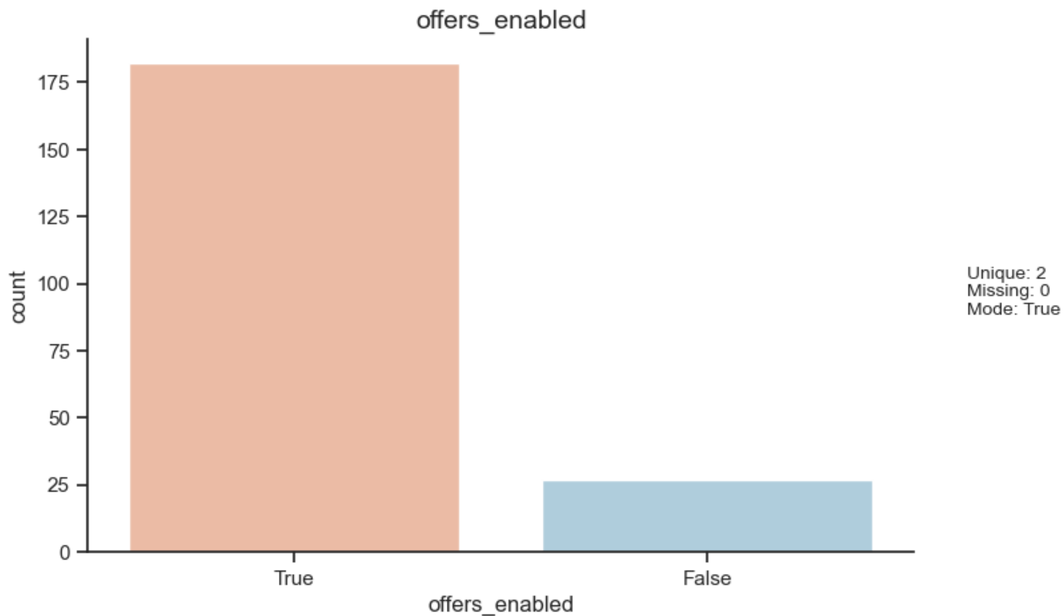
Shipping Price



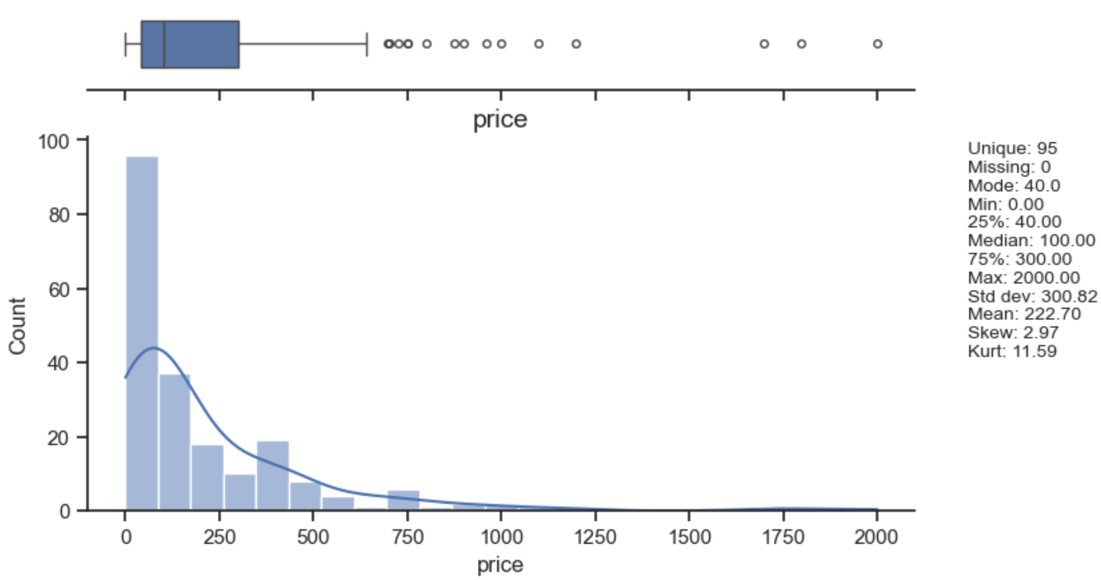
Product Type



Offers Enabled



Price



Data Understanding: EDA, Bivariate Analysis

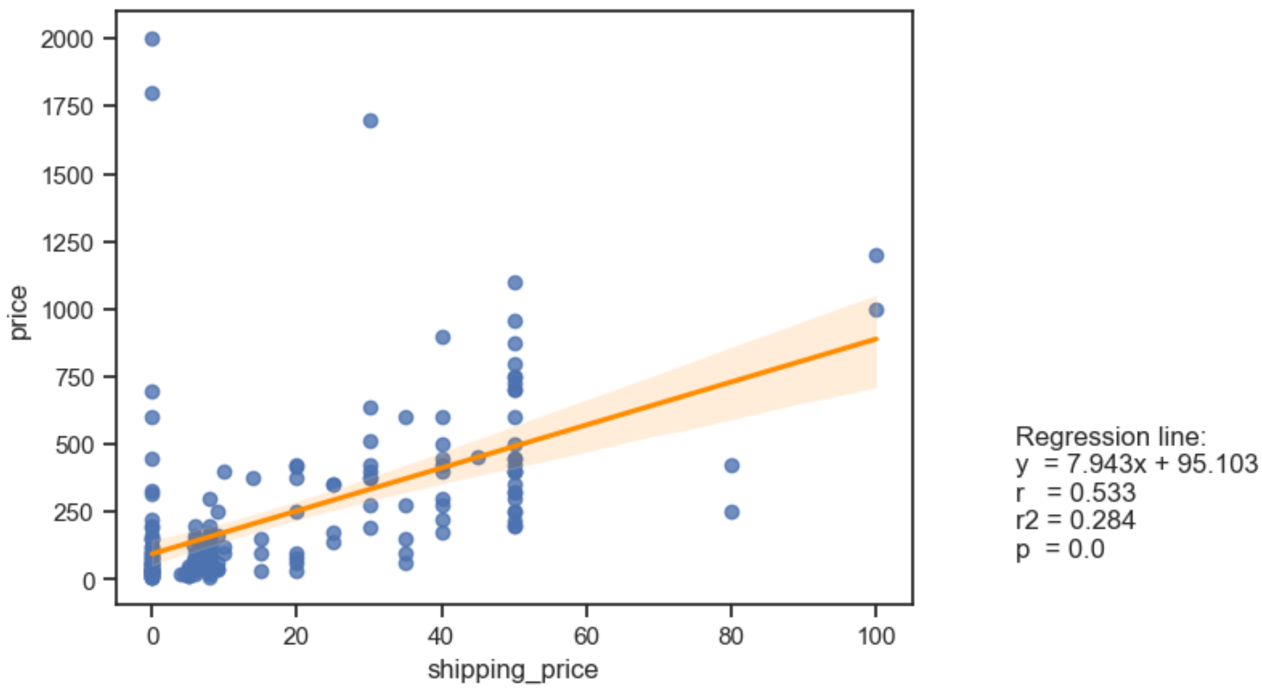
This report details the relationship between each potential feature with the label “price”

Summary Table

	missing	p	r	y = m(x) + b	F	X2
make	0.00%	0.0000	-	-	2.7631	-
product_type	0.00%	0.0000	-	-	23.041	-
condition	0.00%	0.0186	-	-	2.4752	-
offers_enabled	0.00%	0.0728	-	-	3.2533	-
shipping_price	0.00%	0.5328	0.0	y = 7.9426(x) + 95.1032	-	-
description	0.00%	NaN	-	-	NaN	-

The remainder of the report includes greater details on each relationship. I find that almost all of the features are worth including during the modeling phase, which includes the text analytics portion of the data mining process to find meaningful additional variables to contribute to the overall fit and predictive accuracy of the model.

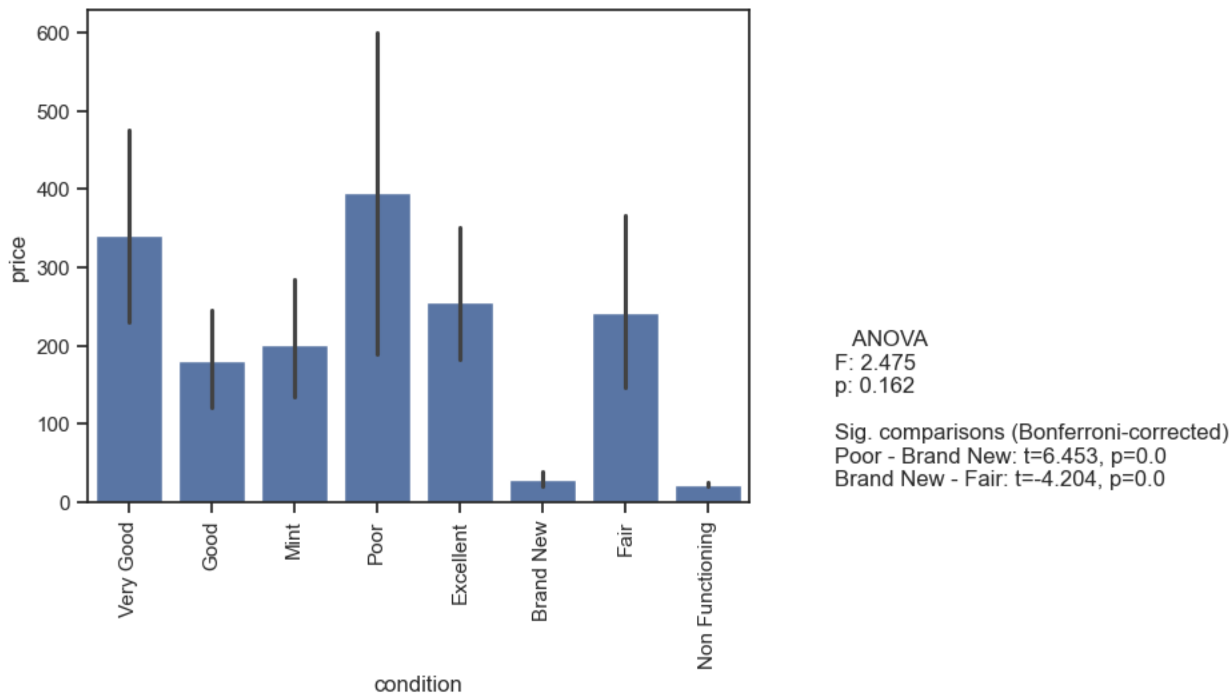
Shipping Price



Summary:

Shipping price appears to have a positive effect on price, although it is a small effect with a low R2 value. The shipping price was a variable that showed it needed transformation later on to be useful in the MLR model, so was transformed using a square root transformation to create a normal distribution. As shipping_price increased, on average, price also increased, with exceptions (outliers). The information gained from this visual is that as price increases the customer is also willing to pay more for shipping price, as it becomes a lower percentage of the overall cost of the item.

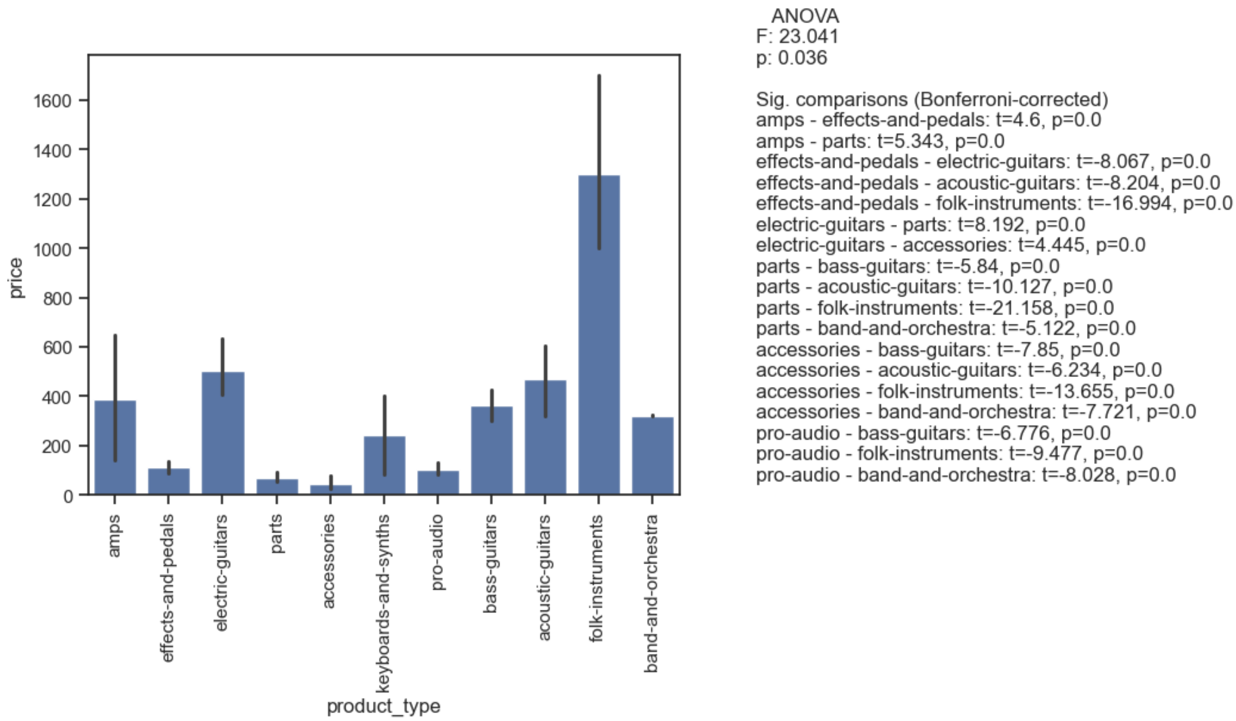
Condition



Summary

The condition of the item for "Very Good" and "Poor" items have the highest effect on price. This may be due to two factors: the first, that very good items are newer items that have been lightly played, that are most in demand, and second, that poor item condition means that there is an older item whose condition does not matter as much as the story, year, value or another factor contributing to its price. Additionally, this could mean that the Brand New items available in this dataset only represented items of lower value of different products, due to the expense necessary to gain a brand new item. More analysis would be necessary and likely future gathering of data to improve the effect of condition on price would be required.

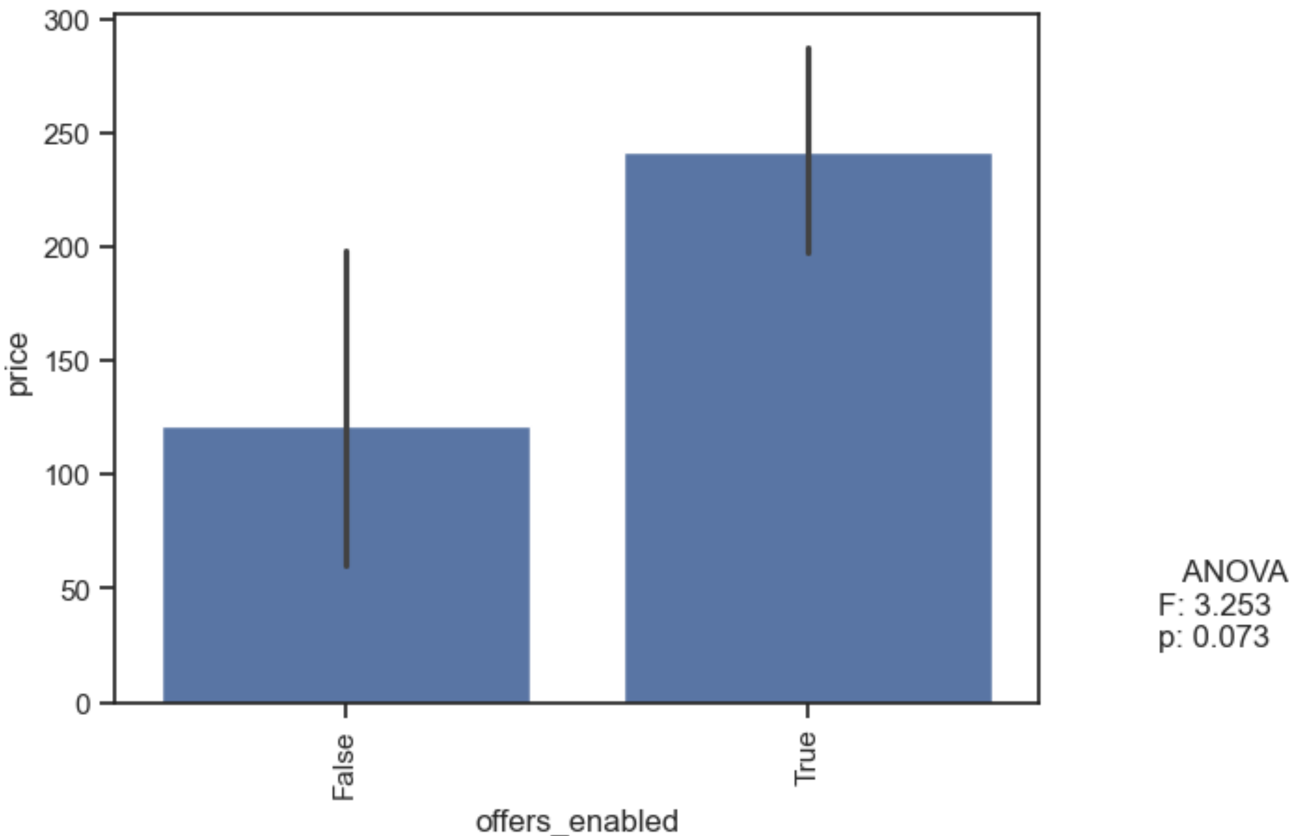
Product Type



Summary

Folk Instruments, electric guitars, and acoustic guitars seem to have a higher effect size on price than the other product categories, with the exception of amps being somewhat positively related alike to folk, electric, and acoustic instruments.

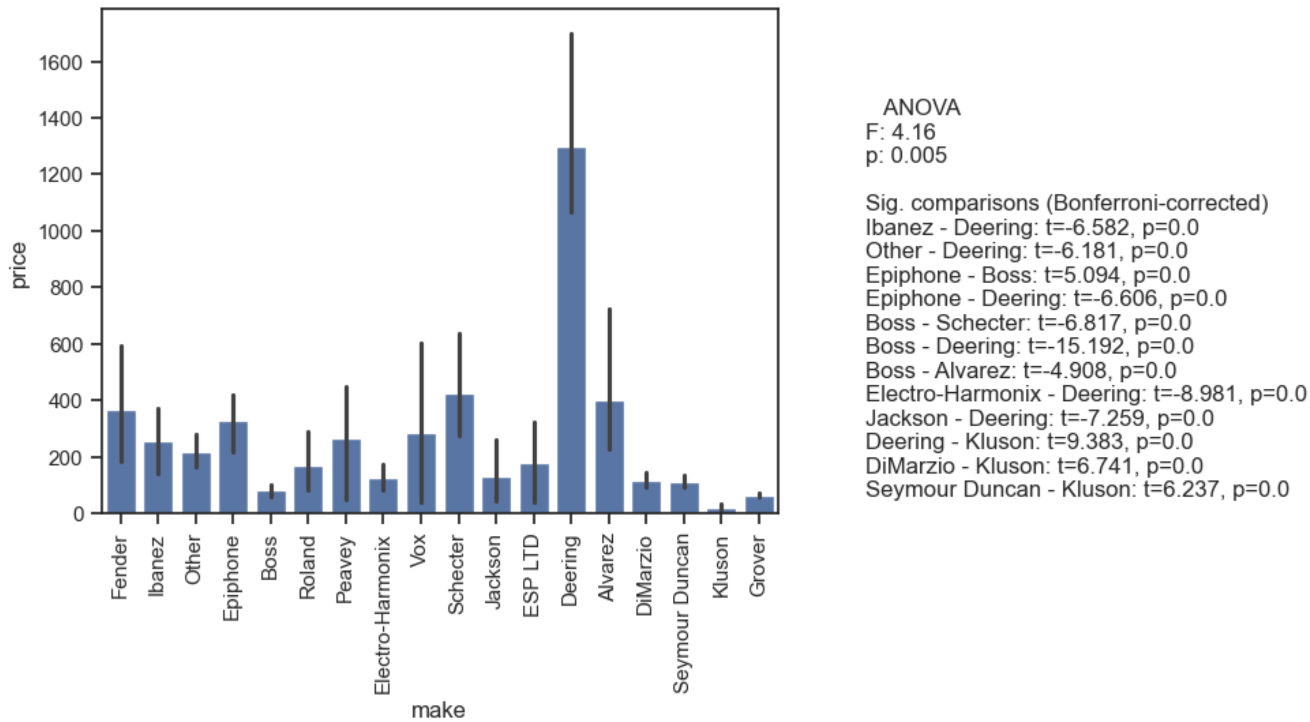
Offers Enabled



Summary

Offers enabled had a slightly high p-value to be considered statistically significant, but is still worth looking at due to the possibility that with collecting more data, or from other vendors, this may be a significant variable in an analysis. We can see that when offers are enabled, there is a higher effect on price, and that the effect is positive.

Make



Summary

The make of a product does have a significant p-value overall with 0.005, meaning that the make of a product will have a positive and large effect on price. Deering, Fender, Schecter, and Alvarez appear to have the highest specific price for their make.

Data Quality Report

Missing Data

There were a few pieces of missing data in just a few of the columns. The way the missing data were handled were the following:

- Data Cleaning Step 1 - all missing values have already been removed

Handling Missing Values

```python
# I will drop all missing values, since there are very few missing at random in only three columns.
df_small.dropna(inplace=True)
df_small.info()
```

[591] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
Index: 204 entries, 0 to 212
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   condition       204 non-null    object
 1   make            204 non-null    object
 2   shipping_price  204 non-null    float64
 3   product_type    204 non-null    object
 4   offers_enabled  204 non-null    object
 5   description     204 non-null    object
 6   price           204 non-null    float64
dtypes: float64(2), object(5)
memory usage: 12.8+ KB
```

- Data Cleaning Step 2 - previously, duplicate columns with the same exact description verbatim were removed, as these represented duplicate posts.

- Data Cleaning Step 3 - Binning values in the 'make' column to include any make that only had one listing or two listings as "Other"

```python
other_list = []

# create a threshold of having just 1 post only or 2 posts only so that any make that is 1 unique value or with 2 postings will be binned in "other"
for make in df_proportions.itertuples():
    if make.Count == 1:
        other_list.append(make[0])
    elif make.Count == 2:
        other_list.append(make[0])

for c in other_list:
    print(f'{c}, ', end='')
```

✓ 0.0s Python

```
Gibson, Mako, Rocktron, TC Electronic, Charvel, Neko, Line 6, Squier, Radial, Shubb, Paloma, Hosa, Mogami, Alesis, Savona, FGN Fujigen, Crafter, Tech 21, Voodoo Lab, Catalinbread, Morley, Bare
```

Now that each value that is going to be binned into the "other" make column is identified, which are just those that only have 1 post as a make or 2 posts as a make (so it's a completely unique brand/make or only has 2 postings with the same brand/make), it's time to replace that brand/make name with the term "Other" in the df_small DataFrame.

markdown

```python
for make in other_list:
    df_small['make'].replace(make, 'Other', inplace=True)

# preview only the 'make' column
df_small['make']
```

✓ 0.0s Python

- Data Cleaning Step 4 – dealing with outliers.
- Data Cleaning Step 5: Mathematical Transformations to Correct Skewness & Kurtosis in shipping_price.

```
# Shipping price has a few large outliers. It also has skewness and kurtosis. Shipping Price is right-skewed. Addi

# Removing outliers
# the first option is the empirical rule, the 2nd is the Tukey BoxPlot, and the third option is using the boxplot.
# in the boxplot, there are two major outliers.
# The Z-score signifies how far a point is from the mean (measured in standard deviations). Points with Z-score ab
# Z-scores does not work well with non-normally distributed data or those with heavy tails (like 'shipping_price'
# Priya, B. (2025, January 9). Top 5 Statistical techniques to Detect and handle outliers in data. Statology. http

# Tukey Box-Plot Method
import numpy as np

column = 'shipping_price'

Q1 = filtered_df[column].quantile(0.25)
Q3 = filtered_df[column].quantile(0.75)
IQR = Q3 - Q1

# Define bounds
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filter out outliers
df_filtered = filtered_df[(filtered_df[column] >= lower_bound) & (filtered_df[column] <= upper_bound)]
print(f'Shape of filtered_df with outliers: {filtered_df.shape}')
print(f'Shape of filtered_df without outliers: {df_filtered.shape}')
```

[609] ✓ 0.0s

```
Shape of filtered_df with outliers: (199, 7)
Shape of filtered_df without outliers: (195, 7)
```

Summary

The code snippet about shows how the IQR Tukey-Boxplot method was utilized to create a mathematical transformation on shipping_price to create 4 outliers that were removed from the dataset, in order to prepare the data for modeling.

- Data Cleaning Step 6: Correcting Skewness in price

```
# since price has a larger postive right skew, I will use a cube root transformation
df_filtered['price'] = np.cbrt(df_filtered['price'])
unistats(df_filtered)
```

✓ 0.0s

	Count	Unique	Type	Min	Max	25%	50%	75%	Mean	Median	Mode	Std	Skew	Kurt
condition	195	8	object	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
make	195	18	object	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
shipping_price	195	24	float64	0.0	7.07	0.0	2.83	5.48	3.03	2.83	0.0	2.51	0.27	-1.18
product_type	195	11	object	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
offers_enabled	195	2	object	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
description	195	195	object	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
price	195	91	float64	1.94	12.6	3.68	4.64	6.77	5.25	4.64	3.42	2.11	0.88	0.56

- Data Cleaning Step 7: Natural Language Processing - Dealing with the text data to make it useable.

```
import spacy # import the required modules/libraries for Natural Language Processing
import pandas as pd

nlp = spacy.load("en_core_web_sm")

# create the nlp processing function
nlp = spacy.load("en_core_web_sm")

# apply nlp to the description column to tokenize each document (row of the description column) and create a new tokenized column.
df_filtered['Tokenized'] = df_filtered['description'].apply(lambda x: nlp(x))
```

✓ 5.4s

\+ Code \+ Markdown

```
df_filtered.head(5)
```

✓ 0.0s Open 'df_filtered' in Data Wrangler

	condition	make	shipping_price	product_type	offers_enabled	description	price	Tokenized
0	Very Good	Fender	6.708204	amps	False	This is the Tweed version of the Fender ...	7.691372	(&, nbsp;This, is, the, Tweed, version, of, th...
1	Good	Ibanez	2.549510	effects-and-pedals	True	Great condition, with one small ding on the si...	3.684031	(Great, condition, ,, with, one, small, ding, ...
2	Very Good	Ibanez	2.828427	effects-and-pedals	False	Ibanez Prime Dual Chorus, model PC-10. Not a c...	4.121285	(Ibanez, Prime, Dual, Chorus, ,, model, PC-10,...
3	Mint	Fender	5.477226	electric-guitars	False	Fender Standard Tele. Its a 2014-2015 Model. L...	7.368063	(Fender, Standard, Tele, ., Its, a, 2014, -, 2...
4	Good	Fender	3.872983	parts	True	1990s Japanese Fender Strat body. It is matte ...	4.641589	(1990s, Japanese, Fender, Strat, body, ., It, ...

Summary

The code snippet shows data cleaning step 7, which includes the tokenization of the text for natural language processing and feature engineering.

Predictive Analytics: MLR

Model Used: Ordinary Least Squares MLR with Min/Max Normalization

Key features: Dummy Coded Categorical features and VIF addressed with 47 Degrees of Freedom.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.692
Model:                            OLS   Adj. R-squared:                  0.594
Method:                 Least Squares   F-statistic:                     7.036
Date:                Wed, 05 Feb 2025   Prob (F-statistic):           3.41e-20
Time:                        19:34:16   Log-Likelihood:                 154.56
No. Observations:                 195   AIC:                            -213.1
Df Residuals:                     147   BIC:                            -56.01
Df Model:                          47
Covariance Type:            nonrobust
------------------------------------------------------------------------------
```

R2:	0.6923
R2-adj:	0.5939
MAE:	0.078
RMSE:	0.1095

Predictive Analytics: Decision Tree

Model Used: Basic Decision Tree

Train/Test Split: 90/10

R squared:	0.3148761046582699
MAE:	1.1774777986722555
RMSE:	1.5062632170938346

Recommended Model: MLR

Because Decision Tree is prone to overfitting and is better suited for classification tasks, and the data was able to be normalized into normal distributions, the MLR model performed best with an R2 value of 0.69 vs. the Decision Tree R2 value of 0.31. Additionally, MLR had much lower MAE and RMSE at 0.078 and 0.1095 indicating a much better fit with lower amounts of residuals. In this case, MLR has much better predictive power on this dataset than decision tree.

Prescriptive Analytics: Linear Optimization

Business Problem: Maximize Price

Since the goal here is maximize the price of each sale, the idea is to look at the model to determine within the constraints, how many of each type of listing to create. To create a simple start in optimization, I will look at the binary variable to optimize: looking at only two product types that had high coefficients and p-values at 0.000. This is a hypothetical problem but could actually produce a realistic result for a guitar store just looking at two basic types of guitars to sell which are acoustic or electric.

Problem Definition

- Product Type: Acoustic Guitars or Electric Guitars are the two types of product produced.
- Variables: production time, cost to produce, storage space
- Resource Information
  - Acoustic Guitar
    - Produces $500 in price.
    - Requires 0.5 Hours of packing time.
    - Costs $350 to produce.
    - Uses 2 units of storage space.
  - Electric Guitar
    - Produces $300 in price.
    - Requires 0.15 hours of packing time.
    - Costs $150 to produce.
    - Uses 1 unit of storage space
- Constraints:
  - 1000 hours of packing time
  - $60,000 budget
  - 125 units of storage space
  - Minimum Demand total Acoustics: 40
  - Minimum Demand Total Electrics: 75

Optimal solution found with pulp:

Acoustics Listed: 55
Electrics Listed: 390
Total Price: $144500

Discussion, Calibration, and Verification

Model results indicate that interesting text features known as “tokens” were utilized to define named entities, which specifically were able to increase predictive power for the MLR method. When choosing just a few features to begin with, conducting NLP on the description data, each document was able to yield features up to 55 columns from just 6 (5 features, 1 label). According to Reverb.com (2023), an extensive price guide has been created for customer to understand how to price their individual instruments. This price guide is based on real descriptive statistics based on selling prices for similar items (association analytics). Reverb.com on their price guide shows times of year (“Date” was a feature that was statistically significant in my model using named entity recognition) and the condition of the item. Pricing estimates are based on the same item at the same condition. According to the models determined, there are other important factors, with the “very good” condition contributing the most statistically significant amount to overall price.

Business Decisions

- Product Categories of Acoustic Guitars and Electric Guitars should be more heavily focused on if the goal is to maximize the price of each individual item.
- Items in “Brand New” Condition should be further analyzed to see if there is room for growth in price by diagnosing what specific items, brands, makes, and other features are in “Brand New Condition.”
- Since People, Organizations, & Dates were all named entities that contributed statistical significance to the model, these specific types of descriptors should be included in each ad’s description to increase the price of the item.
- The Make “Deering” should be further analyzed, explored, and utilized since the price of the make Deering was much higher than any other specific Make in relation to price.

Recommendations for Practice, Future Research, and Conclusions

In future analysis, deeper cleaning and processing of the description data will be utilized to discover if there are more valuable features contributing to overall price. Additionally, since there is a large amount of data and API integration available from Reverb.com, larger and further analysis can be conducted for all shops across Reverb.com rather than only the Tetelestai Guitars shop. Additionally variables will also be analyzed such as product year, country of manufacture, and region sold.

References

How to use the Reverb Price Guide. (10 March, 2023). Reverb.com.

https://reverb.com/news/how-to-use-the-reverb-price-guide