stockX Off-White & Yeezy Sale Analysis | User Dashboards

In 2019, StockX once again held their data analysis contest offering up to $1000 in credit to the winner creating a stellar visualization using ~100K rows of Yeezy and Off-White sales data. In this project, although not eligible for the contest in 2020, I hoped to create a work-flow of dashboards for analysis inspired by real use-cases of someone attempting to enter the interesting and growing market of sneaker sales. *Original Contest Info: https://stockx.com/news/the-2019-data-contest/*

In this project I also strived to produce a model which once in production would help a user enter some details about their sneaker or shoe and understand what would be a potential re-sale value.

My idea was to answer the **When**, **What** and **How** of after-market sneaker sales.
When - Visually understand the months one should look out for sneakers being released and when to sell to get best return.
What - What sneakers would sell best and what sizes can one look to get to a return from. Also which sneakers have the highest selling volume and would offer the 'easiest' selling opportunity.
How - How to pick and plan the best opportunity for buying and selling.

adidas YEEZY

"OFF-WHITE"


adidas Yeezy Boost 350 V2 Butter
$220


adidas Yeezy Boost 350 V2 Beluga 2.0
$220


adidas Yeezy Boost 350 V2 Zebra
$220


adidas Yeezy Boost 350 V2 Blue-Tint
$220


adidas Yeezy Boost 350 V2 Cream White
$220


adidas Yeezy Boost 350 V2 Sesame
$220


adidas Yeezy Boost 350 V2 Static
$220


adidas Yeezy Boost 350 V2 Semi-Frozen Yellow
$220


Air Jordan 1 Retro High Off-White University Blue
$190


adidas Yeezy Boost 350 V2 Static Reflective
$220


Nike Air Presto Off-White Black 2018
$160


Nike Air Presto Off-White White 2018
$160


Nike Air VaporMax Off-White 2018
$250


Nike Blazer Mid Off-White All-Hallows Eve
$130


Nike Blazer Mid Off-White Grim Reaper
$130


Nike Zoom Fly Off-White Pink
$170


Nike VaporMax Off-White Black
$250

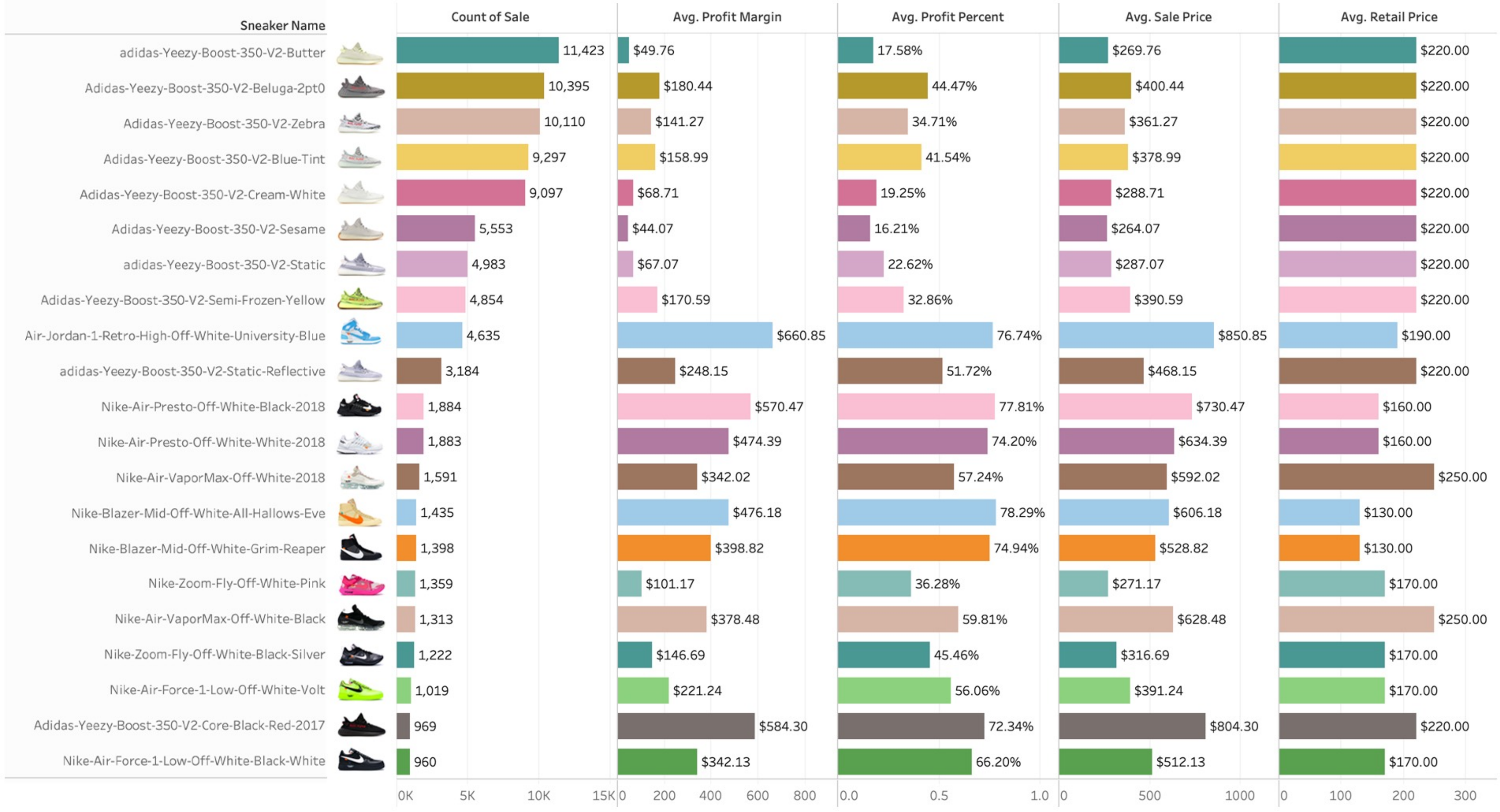

Nike Zoom Fly Off-White Black Silver
$170


Nike Air Force 1 Low Off-White Volt
$170


adidas Yeezy Boost 350 V2 Core Black Red 2017
$220


Nike Air Force 1 Low Off-White Black White
$170

Top 21 Sneakers Sorted by Volume (Count Sold)



In the chart above we can see various columns associated with each sneaker. A breakdown of this aggregated data is as follows:
Count of Sale: Represents the number of orders for this particular sneaker
Avg. Profit Margin: Subtracting the retail price from the StockX final sale price we obtain the profit margin. Taking the average profit margin for every order of this sneaker we obtain this column's value.
Avg. Profit Percent: Average of the percent representation of the number above. (Profit Margin / Sale Price)
Avg. Sale Price: The average of the final sale price for each sneaker.
Avg. Retail Price: The average retail price for this sneaker. This is a standard column thus there is no variance in the numbers.

The top selling sneakers are a mix of Off-White and Yeezy sneakers as we can see. **There is clear winner in terms of most common on the platform** however once the data is broken down **a different story is told in terms of return to the seller.**

Lets say a seller-user is interested in entering the 'shoe game' and wanted to have a decent shot at success.

How would they go about analyzing the current state of the Yeezy and Off-White market?

What if a seller-user wanted to know which items would be easier to flip and the associated margins related to these?

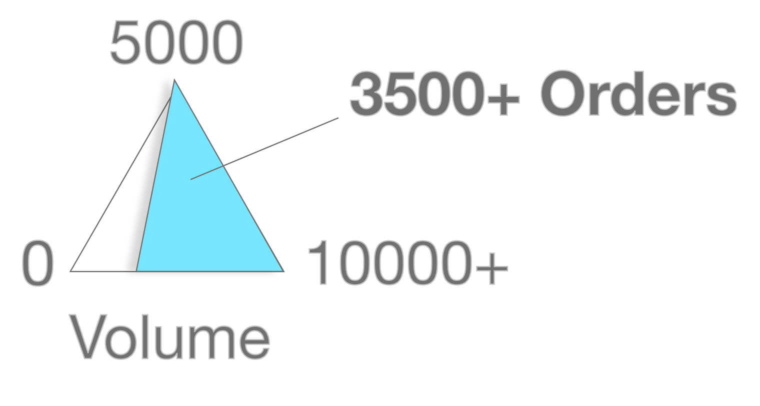
One would understand that the **items which are being sold with the most volume would be the easiest to sell.** Across the chart we see that those **items with more volume** being moved **usually show** a **lower profit margin and profit percent** than those with lower volumes.

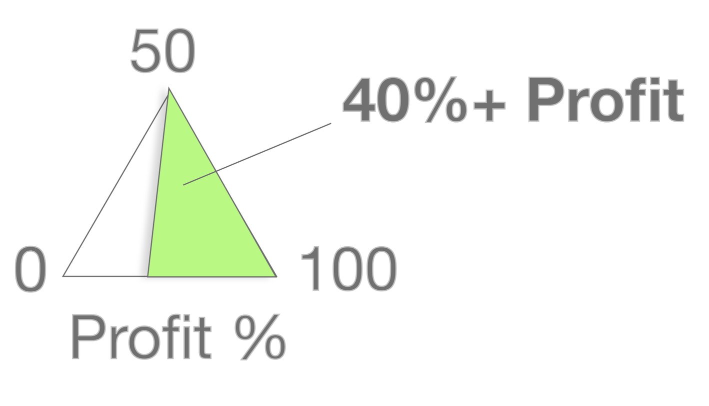
In particular it is interesting to see the **retail price** for the **Off-White** brand shoes being **lower than Yeezy's but** offering **higher profit margins and higher sale prices** alike.

So what sneaker would be the sweet spot for a seller-user?

I have defined the 'sweet spot' as:
> Volume is moderate/high, meaning demand is present and consistent.

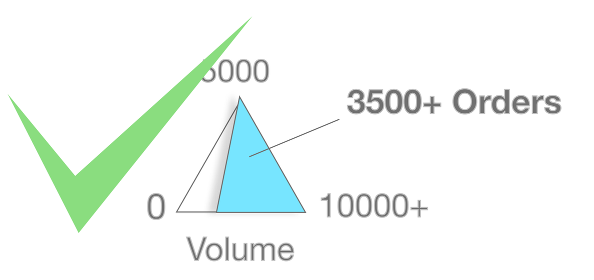
> Profit is more than 40%. If one were to look at those with an average profit margin of 40% it would show a 2x or higher conversion on the retail price for the item.

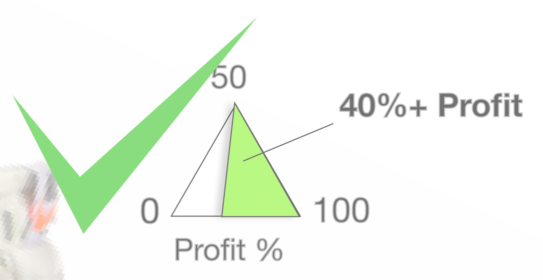




Air Jordan 1 Retro High Off-White University Blue

Count of Sales: 4,635 Orders
Avg. Profit Margin: $660
Avg. Profit Percent: 76.74%
Avg. Sale Price: $850.85
Avg. Retail Price: $190

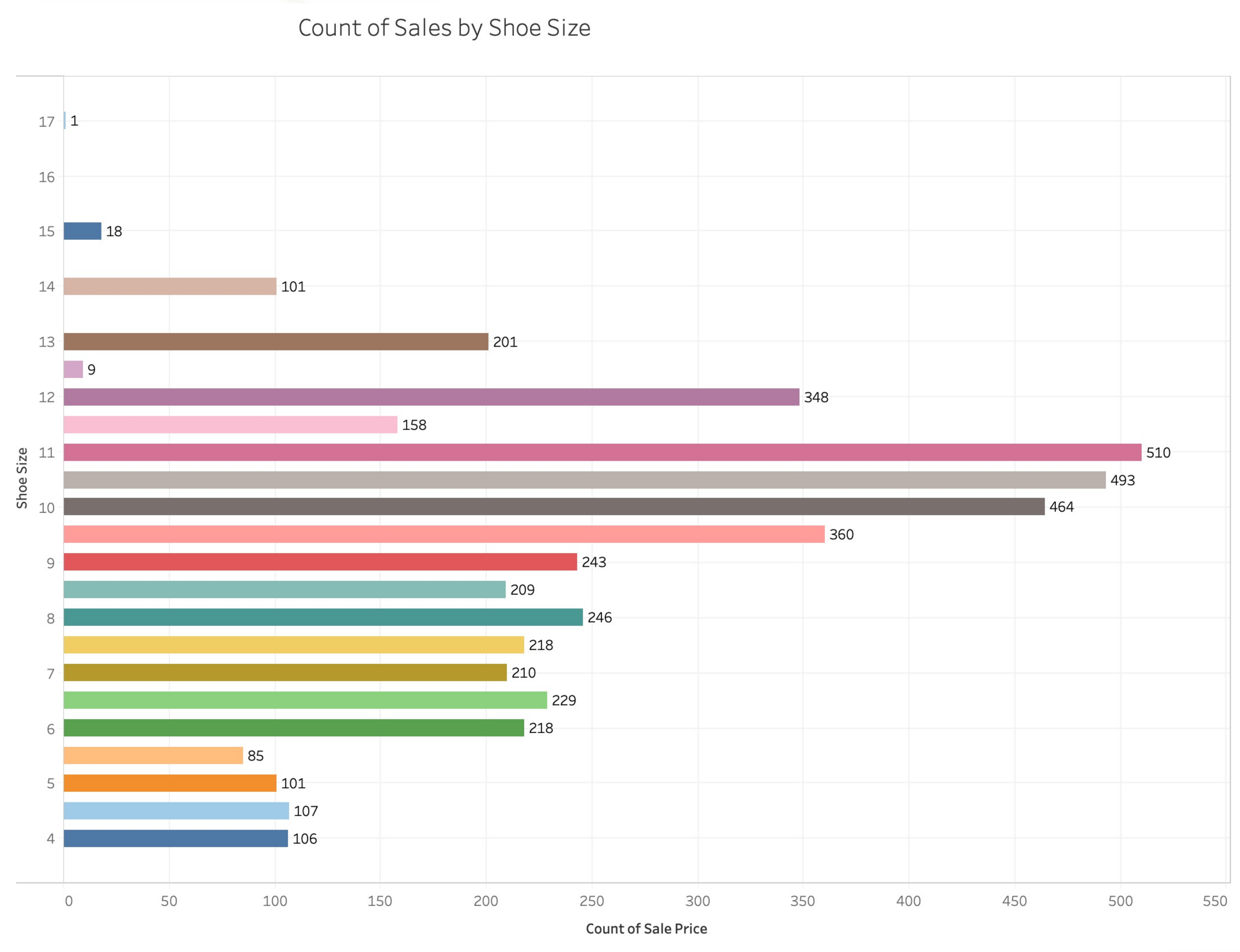




The sneaker above fits our 'sweet spot' in terms of picking an item to watch.

Now, we need to think about **what sizes would sell best if we were to find one of these Off-White University Blue** sneakers. **

** Because the project summary only allows for flat images and not our **Tableau** dashboards I have included a 'global' shoe size comparison chart and one specific for the Off-White University Blue sneaker.



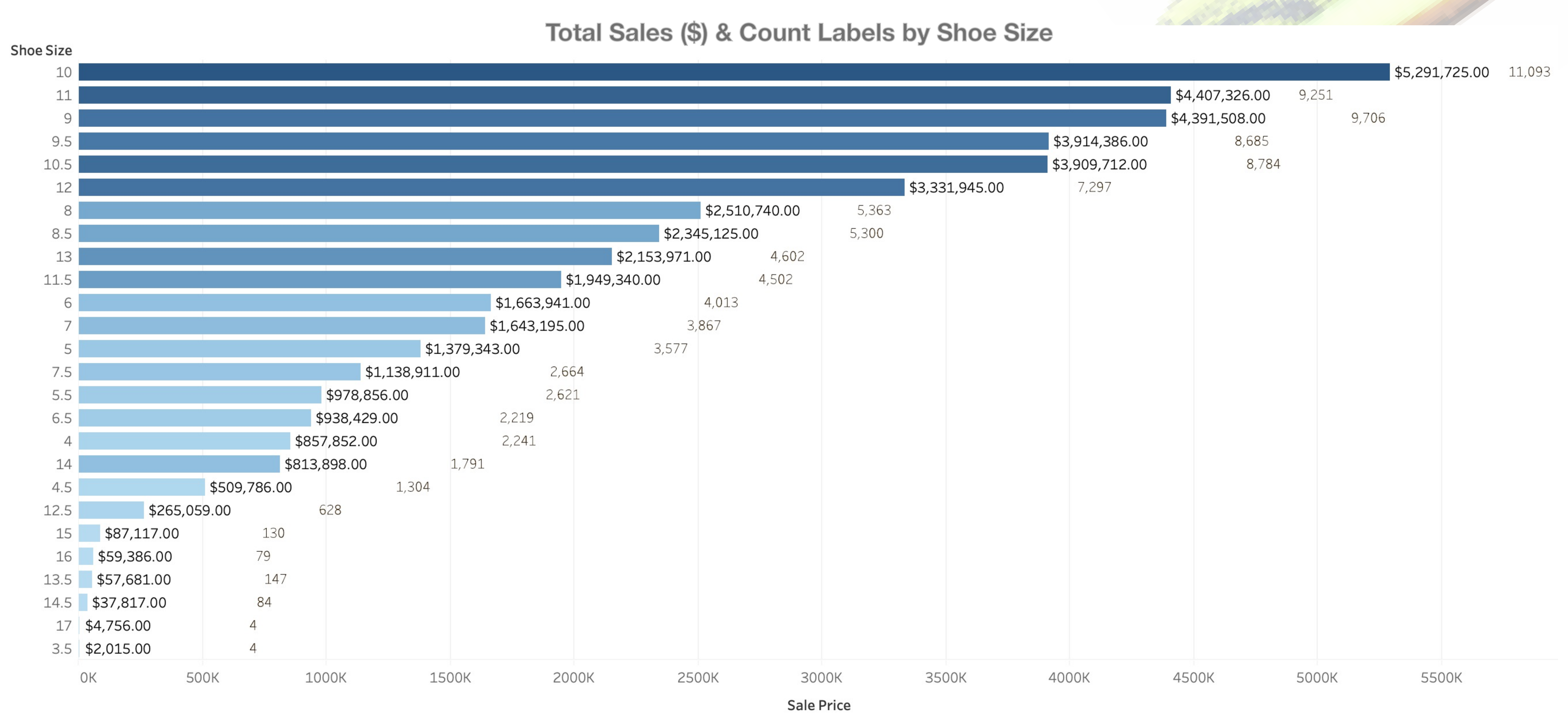
When referencing the **Air Jordan 1 Retro High Off-White University Blue:**

The seller-user would have the **best chance** selling sneakers with sizes **9.5 - 11** than any other as the volume of orders for these are almost double the rest.

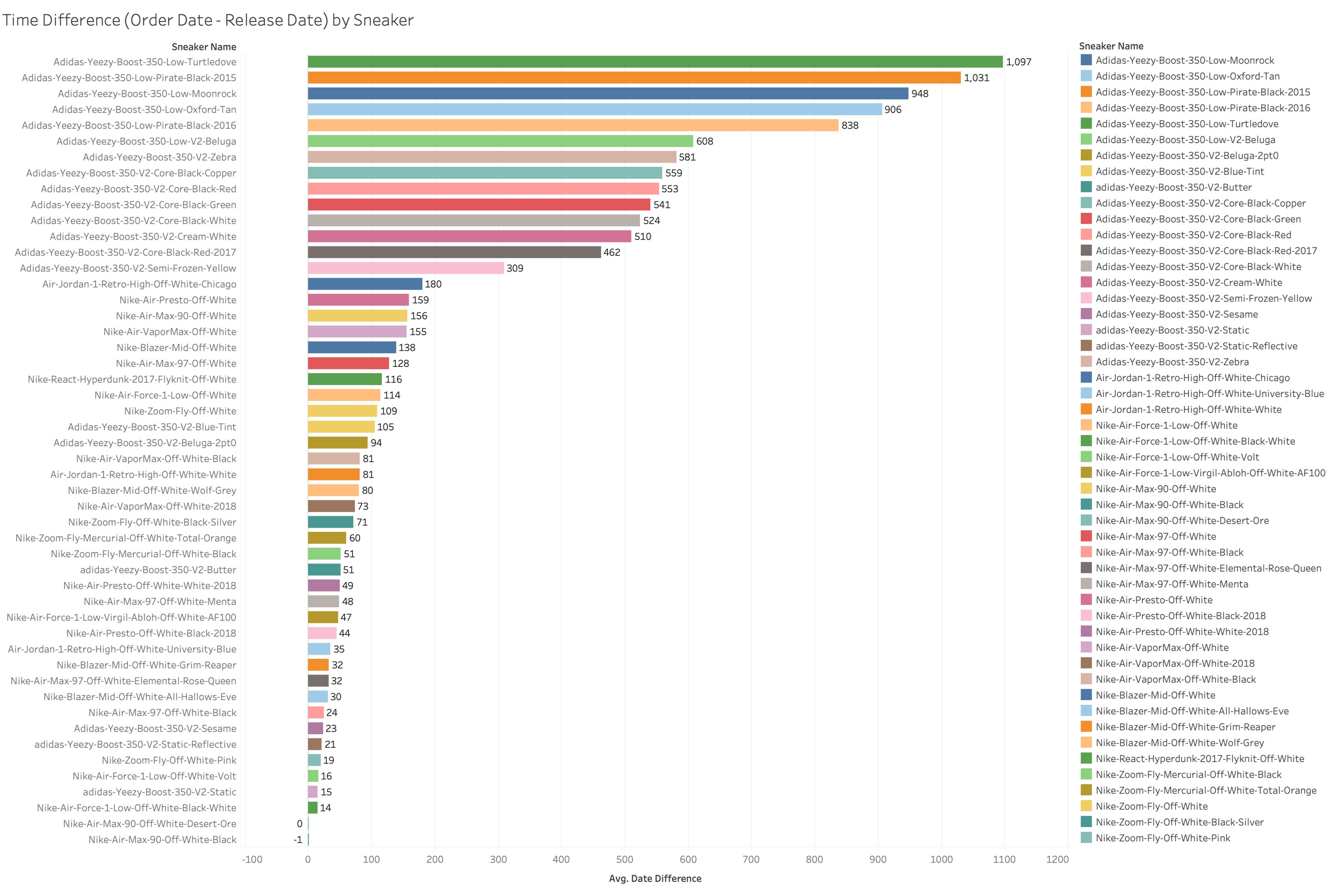
Anything in sizes **6 - 9** would offer a **moderate chance** of being sold.

The **extreme sizes** show a **lower chance** of being sold which would make sense considering what the most common shoe sizes are naturally.

Generally speaking one could see the trend in shoe sizes for the Air Jordan 1 Retro High Off-White University Blue is very comparable to that of almost every sneaker in the dataset. The chart below shows a similar trend and is broken down into Total Sales per Size as well as the order count labeled next to the dollar amount.



Date After Release - The next item to consider is how long after the release date were orders created and how that affects the final sale price. For our sample item, the Off-White University Blue it was just 35 days. However for some items the average orders were almost 1100 days later.

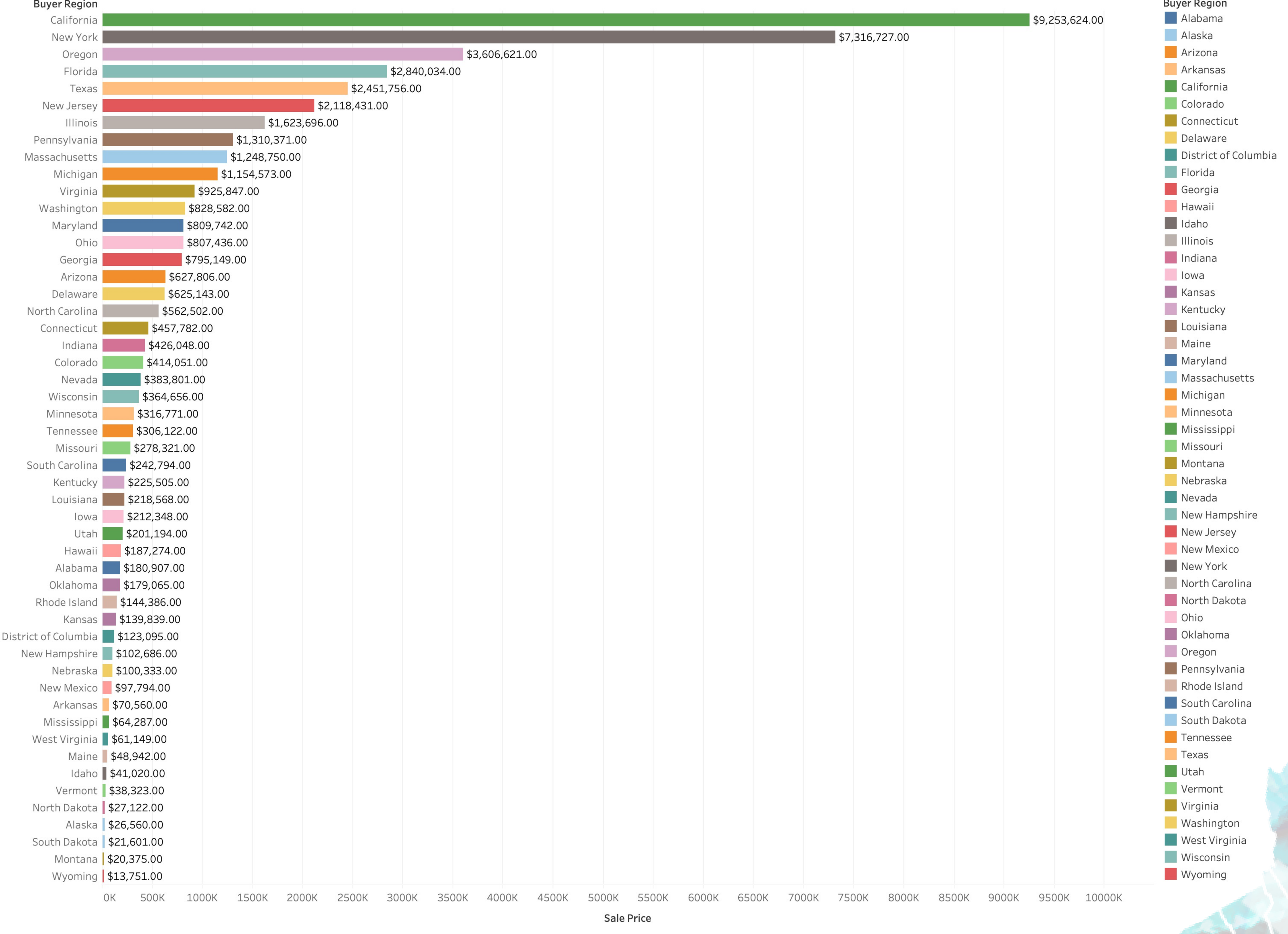


Average of Date Difference for each Sneaker Name. Color shows details about Sneaker Name. The marks are labeled by average of Date Difference.

Trends in Release and Ordering - The matrix below offers a view into patterns associated with the month items are released and when items are ordered most. For the most part, orders are usually occurring in the latter part of the year, close to the year-end Holiday season. There are some orders also in the summer. The release date columns offer a general breakdown of when items are released and later sold. The majority of items released and then sold in these two periods will be released in the months of February, June, September, November and December. It would be wise for a potential seller-user to stay up on release trends during these months and then sell in the subsequent months where the most orders are created by buyer-users.

Sum of Sale Price for each Release Date Day broken down by Order Date Month vs. Release Date Month. Color shows details about Order Date Month. The marks are labeled by sum of Sale Price.

Buyer-User Regions- The chart below shows the distribution of sales per state as the region the order occurred in. The seller-user does not have control over this so it will not be included in the predictive model to be introduced later in the project summary. It is important however for understanding dynamics of sales and if any potential sales could happen via off-platform websites or groups (Facebook groups, forums, etc.)



Sum of Sale Price for each Buyer Region. Color shows details about Buyer Region. The marks are labeled by sum of Sale Price.

On the next page we will cover predictive modeling using the stockx.com dataset provided.

Sale Price Prediction Tool - Say a seller-user has acquired one of the shoes in our dataset, it is in an excellent, or like-new selling condition as well as knowing the size. Would we be able to estimate a potential selling price?

Using predictive models in Python's scikit-learn library we are able to select features for a prediction engine. I've also designed a lightweight UX/UI of what potentially could be done to facilitate this type of workflow.

Model Features / Dataset Columns

Retail Price - Original selling price of the sneaker
Date Difference (days) - Calculated as Order Date-Release Date
Sneaker Name
Shoe Size

scikit-learn Models Used

Linear Regression
Ridge
Lasso
Random Forest Regressor
Decision Tree Regressor

Target Column / Dependent Variable

Sale Price - final price item sold for on stockx.com

These models were chosen because of their ability to process the data and produce a **single numerical value**. In this case it **will predict a potential final sale price** for the seller-user's items.

Model Performance & Metrics

Metric	Decision Tree Regression	Random Forest Regression	Lasso Regression	Linear Regression	Ridge Regression
Train Score	99.866%	99.629%	83.778%	83.778%	83.778%
Test Score	97.501%	98.079%	83.708%	83.708%	83.708%
Run Time(secs)	0.768	25.069	512.449	0.205	0.114
RMSE	$40.45	$35.46	$103.27	$103.27	$103.27
MSE	1636.143	1257.314	10664.320	10664.319	10664.319
MAE	17.007	15.163	60.432	60.432	60.432

Train Score
Describes how well the model fits on training data which can then take in test data to estimate how well our model performs.

Test Score
The test score will show you how well the trained model generalizes for the new 'test' data. In other words, how well it predicts the test data's target value.

Run Time(secs)
The time it took for the computer to run this algorithm with the data pulled in.

RMSE
Looks at the residuals, prediction errors, which are the distance of a point from the regression line. This tells us how well the points are concentrated around the regression line and how well our model is generalizing.

MSE
This Mean Squared Error allows us to penalize for any outliers and handles any negative values as well as MAE. The MSE calculates the average square distance between predicted values and original values.

MAE
The average magnitude of error or Mean Absolute Average is the distance between the average of predictions from the actual values. The units here are $.

Per the results above the most viable model for use in production would be a Decision Tree Regression model. This model would allow the fastest computing time while still boasting a fairly high accuracy in predicting prices on the Training and Test sets. Looking at the RMSE we would be within ~10% of the true price, or better, in relation to the mean of the Sales Price column in the dataset.

Below are 10 samples of the model predicting through the trained Decision Tree Regressor. The incorrect predictions are marked in red.

Predictions:	1097.0	685.0	690.0	1075.0	828.0	798.0	784.0	**466.5**	465.0	465.0
Actual:	1097	685	690	1075	828	798	784	**460**	465	465

Summary & Conclusion Seller-users would be able to rely on this tool, potentially built into the stockx.com website, to predict how well the sneaker will do based on two parameters.

The data inside of the model is setup by assigning each Sneaker Type a 0 or 1 value and allowing it its own column. This is done, as opposed to assigning a unique numerical ID because our model would have taken the large number and used it to calculate a prediction. Assigning these binary classifications as separate columns allowed for a more sensitive model.

The model was trained using Shoe Sizes as well and these sizes were managed in a similar way, providing individual columns with binary, 0 and 1, numerical labels. This method was appropriate since the sizes of the shoes does matter in the sale price per our tables in the previous pages of the document.

Lastly it was seen as relevant to include the differences in time between an order happening and the release date of the sneaker. Sneakers in great condition will increase in price as time goes on. This University Blue Sneaker is trading at around ~$1900 in August of 2020.

The target column used to train the model was the Sales Price, which is the final price the item sold to it's Buyer for.

Per the results above we see our model have above a 90% accuracy, more specifically ~97% for the Decision Tree Regressor model with a 0.768 second computation time on 99K records.

It is important to note although the Random Forest model is slower, it can be improved with hyper parameter tuning and has a better chance of not overfitting data in comparison to Decision Trees.

Below is a simple to use UX/UI for how this model could be applied into a website or similar web-app.

Data & User Input Requirements

Requires User Input

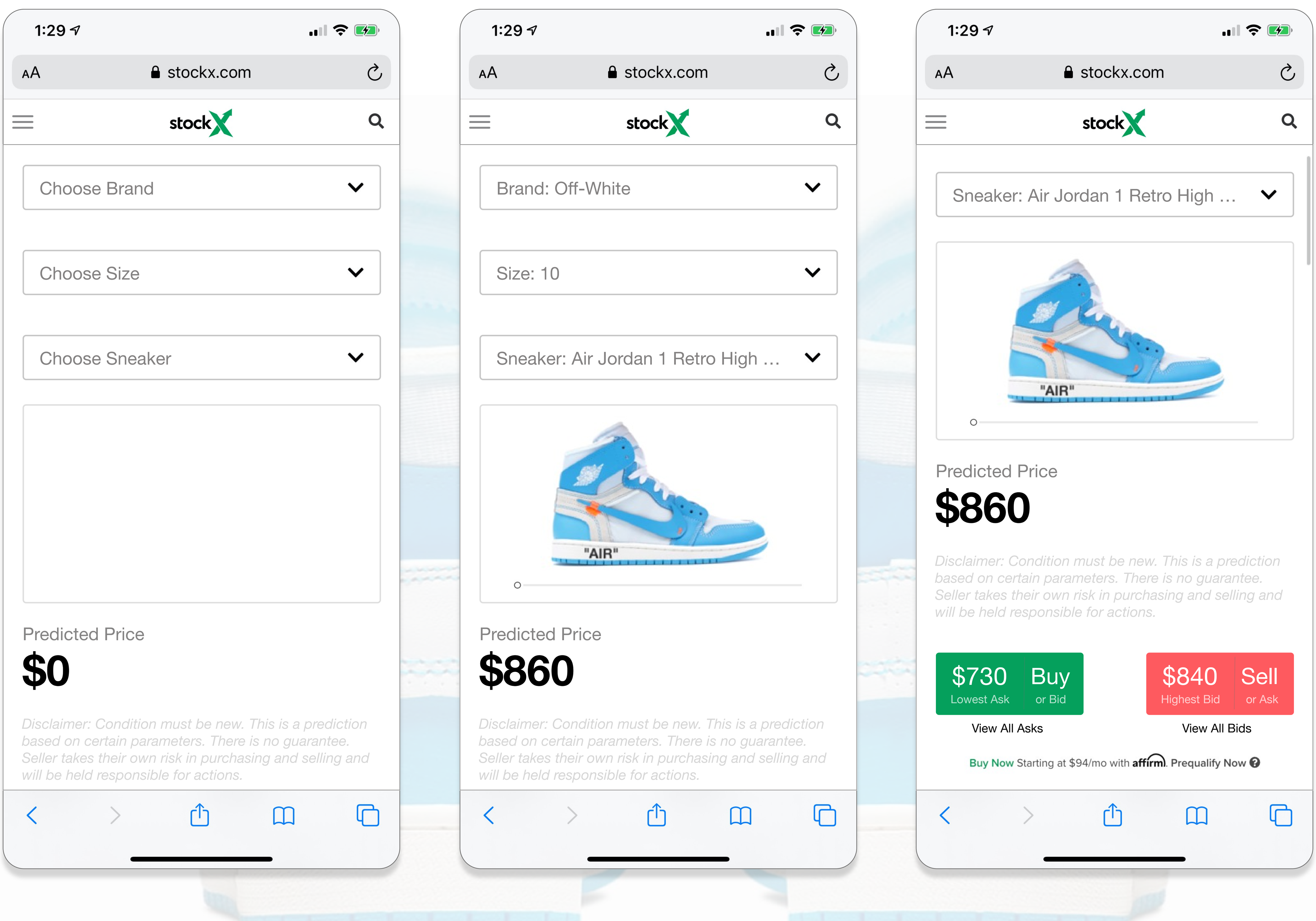
Brand
Sneaker Name
Shoe Size

Database Stored

Retail Price - Original selling price of the sneaker
Date Difference (days) - Calculated as Current Date - Release Date of Sneaker (stored in DB and found in dataset)
Item Image

Predicted Value

Predicted Price - final price item potentially could sell for on stockx.com



Connect with me at
https://www.linkedin.com/in/ginop8/

Gino Parages August 2020 eugeniosp3@gmail.com