SNEAKERS + MACHINE LEARNING

Kaili Hamilton

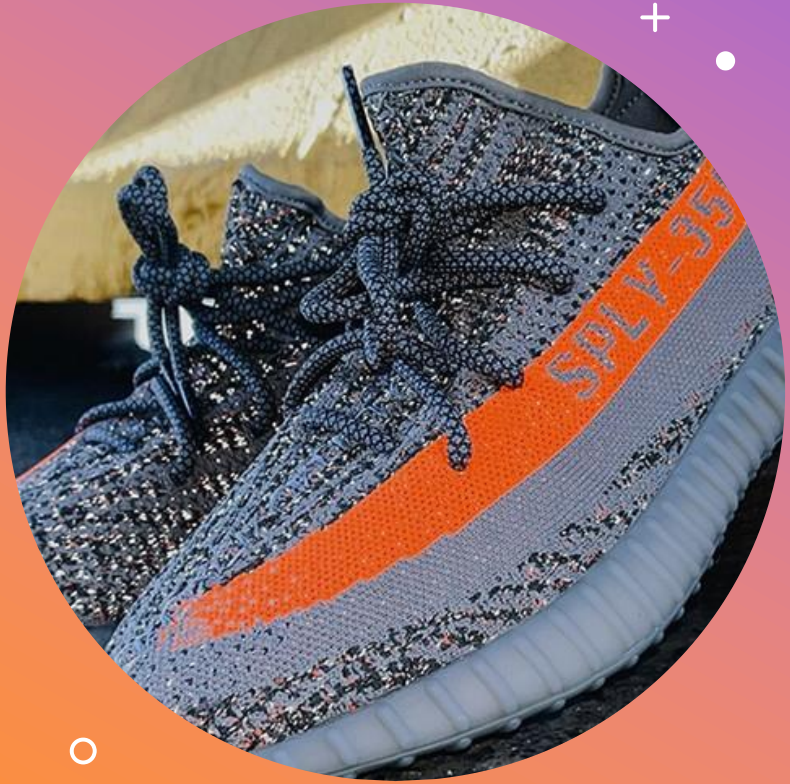
March 23, 2023

Capstone 2 Project

About the Data

- Source: StockX (marketplace to sell and buy sneakers and other cool stuff), Kaggle
- Time frame: 2015 – 2019
- Only two brands/collaborators/design labels:
  - Yeezy (Kanye “Ye” West)
  - Off-White x Nike (Virgil Abloh)
- 50 different kinds of sneakers
- 99,956 sneakers sold in dataset

SNEAKERS



MY QUESTIONS

Can I predict when the best time to resale a sneaker?

What factors are most influential in higher profits when reselling a sneaker?

Machine Learning Cycle

Gather & Explore Data

Found a dataset of interest on Kaggle.

Used StockX data

Pre-Process Data

Jupyter Lab, Pandas, Numpy, Matplotlib, Regex, Datetime

Exploratory Data Analysis

Changed data types

Weird stuff like a space prior to the text “Yeezy”

Extracted colorway and silhouette from one column

Created Profit Margin Column

Build Machine Learning Models & Train the Model

Tableau

Exploratory Data Analysis

Created many graphs to understand data

Evaluate the Model & Make Predictions

Looked for answers to my questions with the visuals and summary statistics.

Interpret the meaning of data

Conclusion

Tell a story of Yeezy and Off-White x Nike sneakers

Machine Learning Cycle

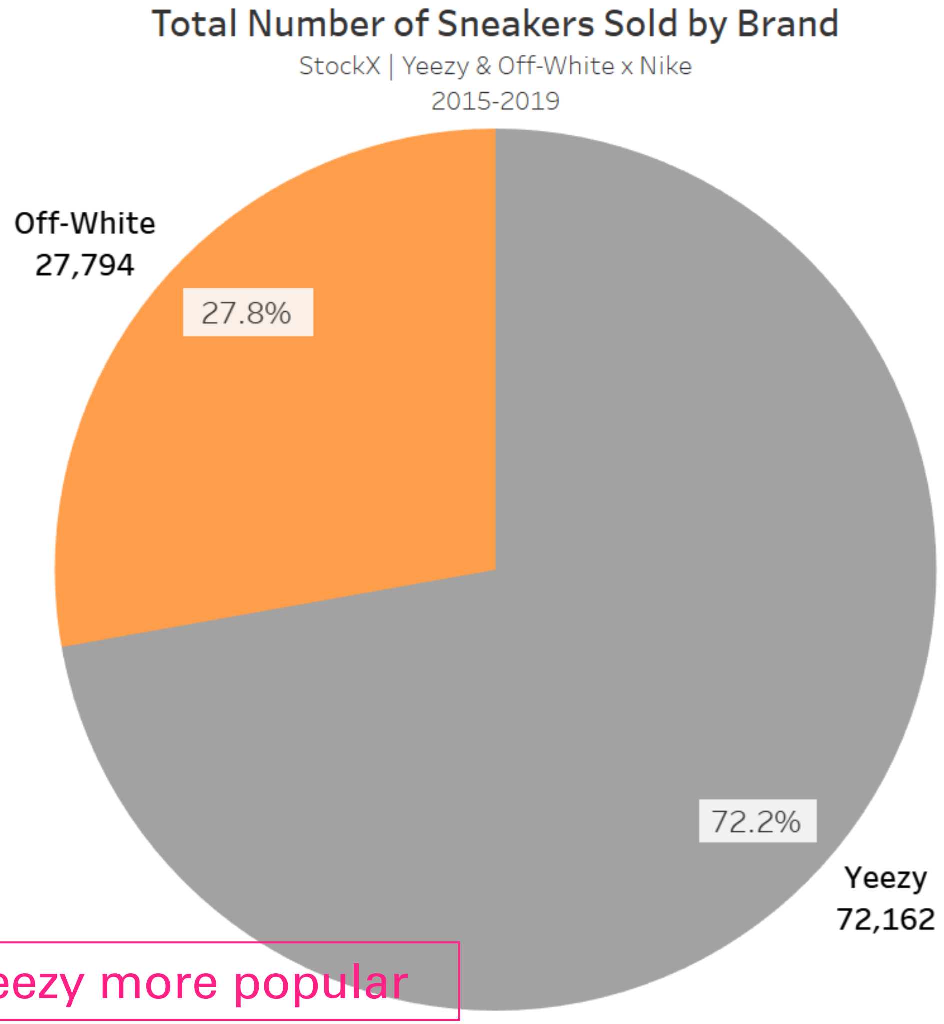
- Gather & Explore Data
- Pre-Process Data
- Build Machine Learning Models & Train the Model
- Evaluate the Model & Make Predictions
- Conclusion

GATHER & EXPLORE DATA

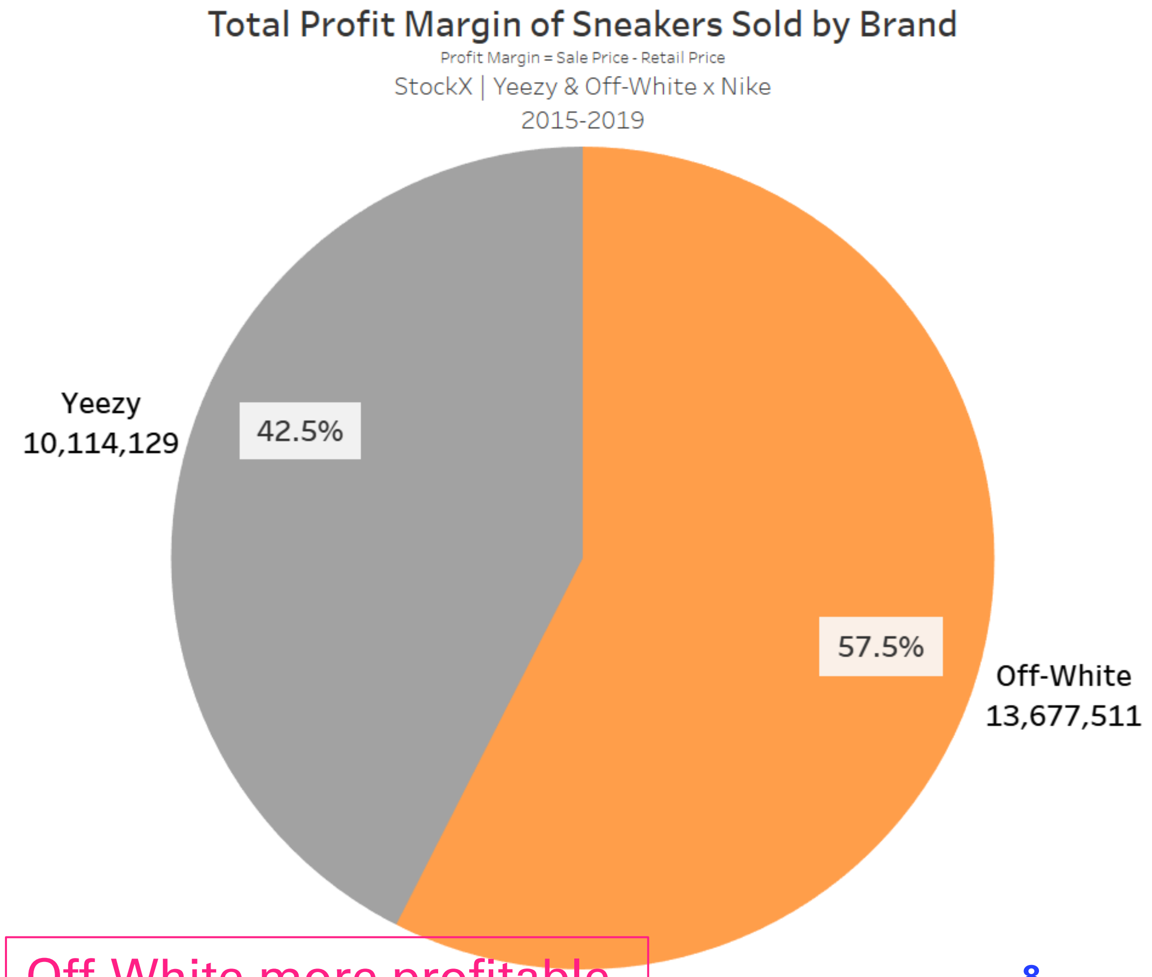
Summary Statistics

	Sale Price	Retail Price	Shoe Size	Profit Margin	Elapsed Time Days
count	99956.000000	99956.00000	99956.000000	99956.000000	99956.000000
mean	446.634719	208.61359	9.344181	238.021129	183.708722
std	255.982969	25.20001	2.329588	266.133179	232.354142
min	186.000000	130.00000	3.500000	-34.000000	-69.000000
25%	275.000000	220.00000	8.000000	58.000000	10.000000
50%	370.000000	220.00000	9.500000	154.000000	56.000000
75%	540.000000	220.00000	11.000000	342.000000	345.000000
max	4050.000000	250.00000	17.000000	3860.000000	1321.000000

Comparing Brands

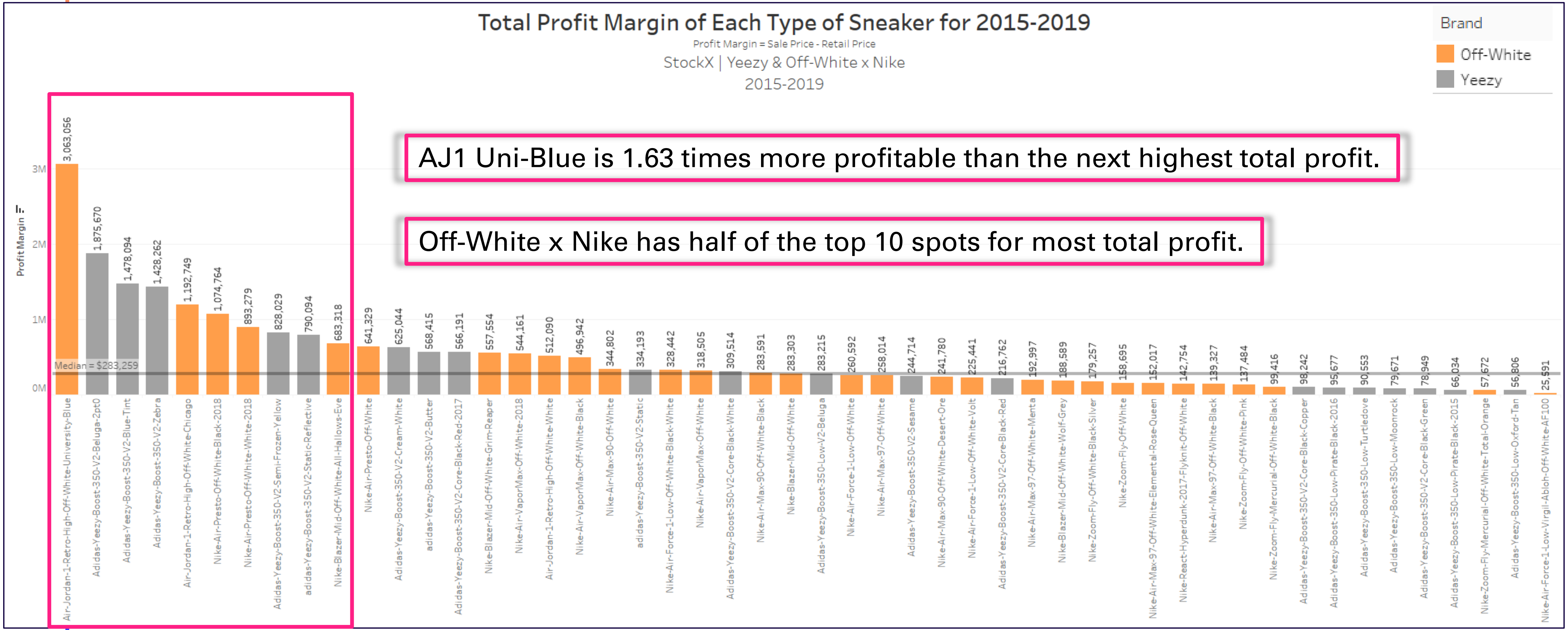


Yeezy more popular

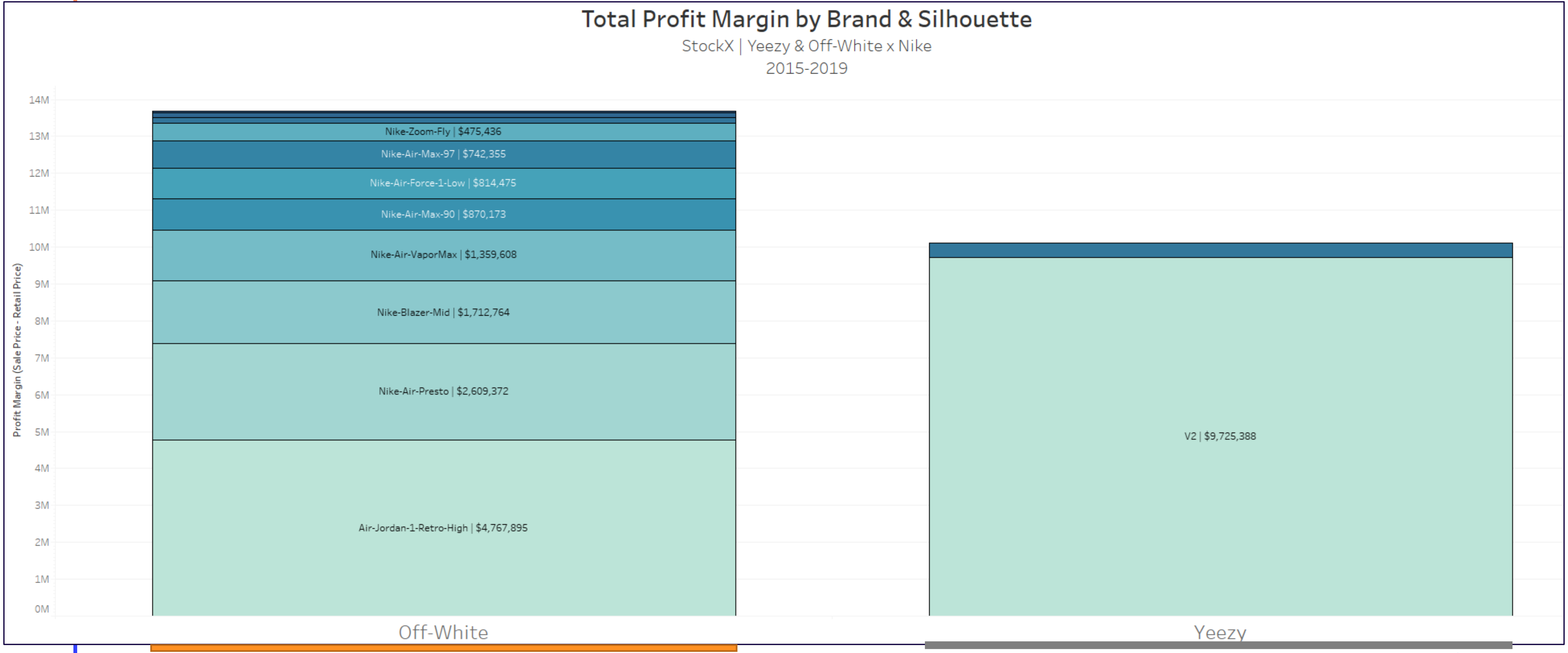


Off-White more profitable

Most Profitable Sneaker: Air Jordan 1 Retro High University Blue

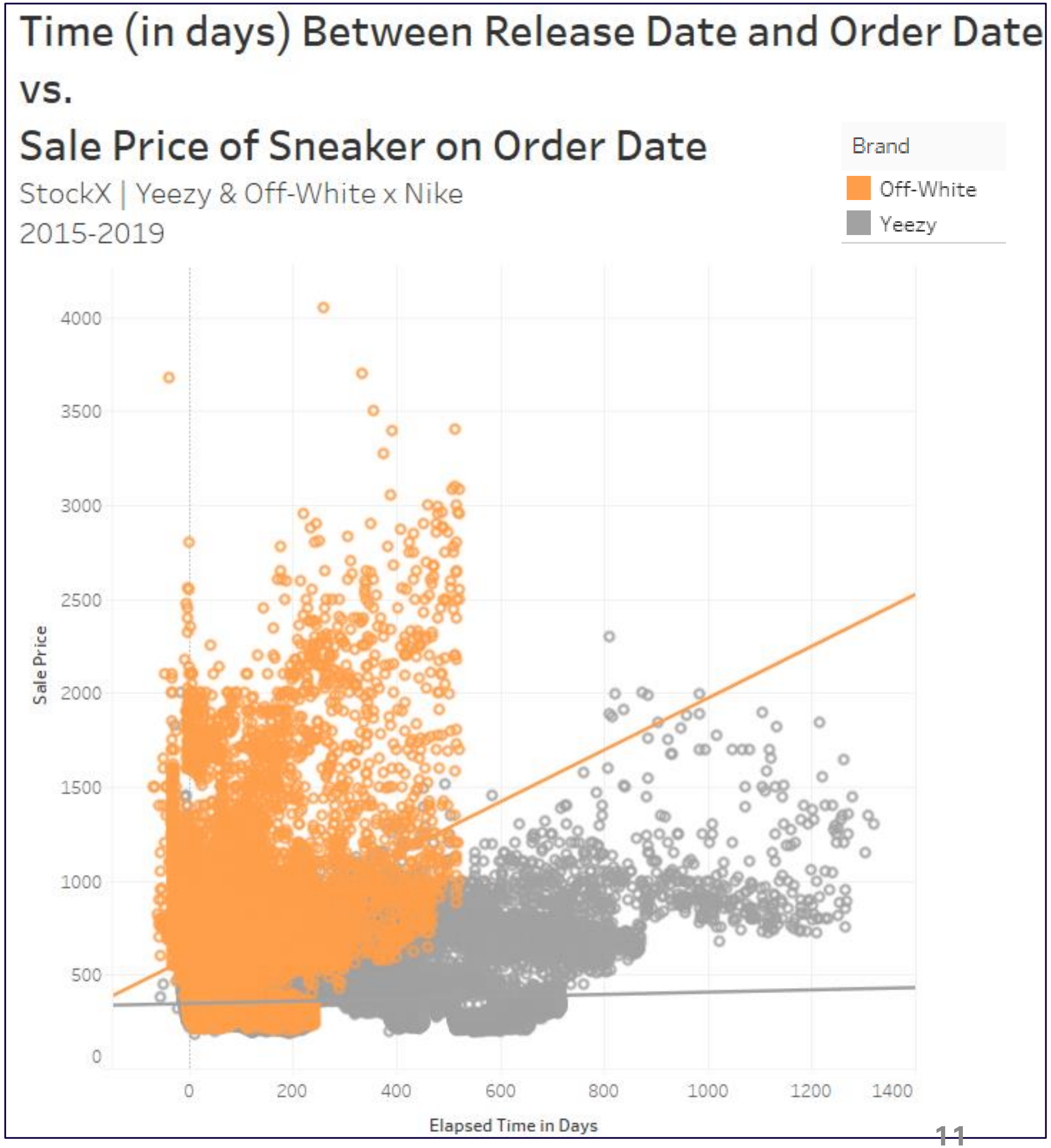


Most Profitable Silhouette: 350 v2



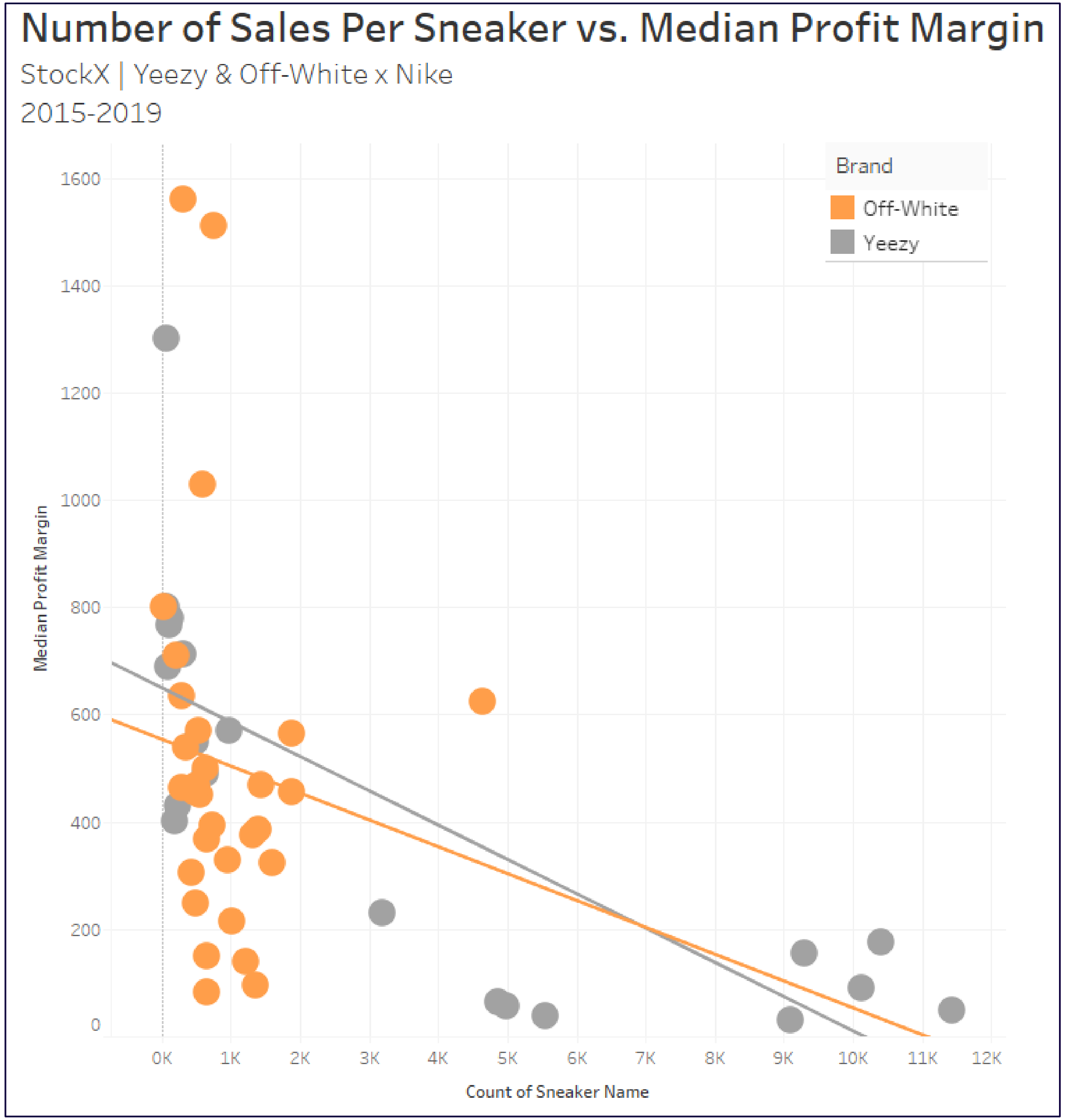
Time between release date and order date with sale price

- Two clear clusters
- Yeezys stayed on the market far longer than Off-White Nikes
- Cluster of Off-Whites that have a higher sale price than Yeezys

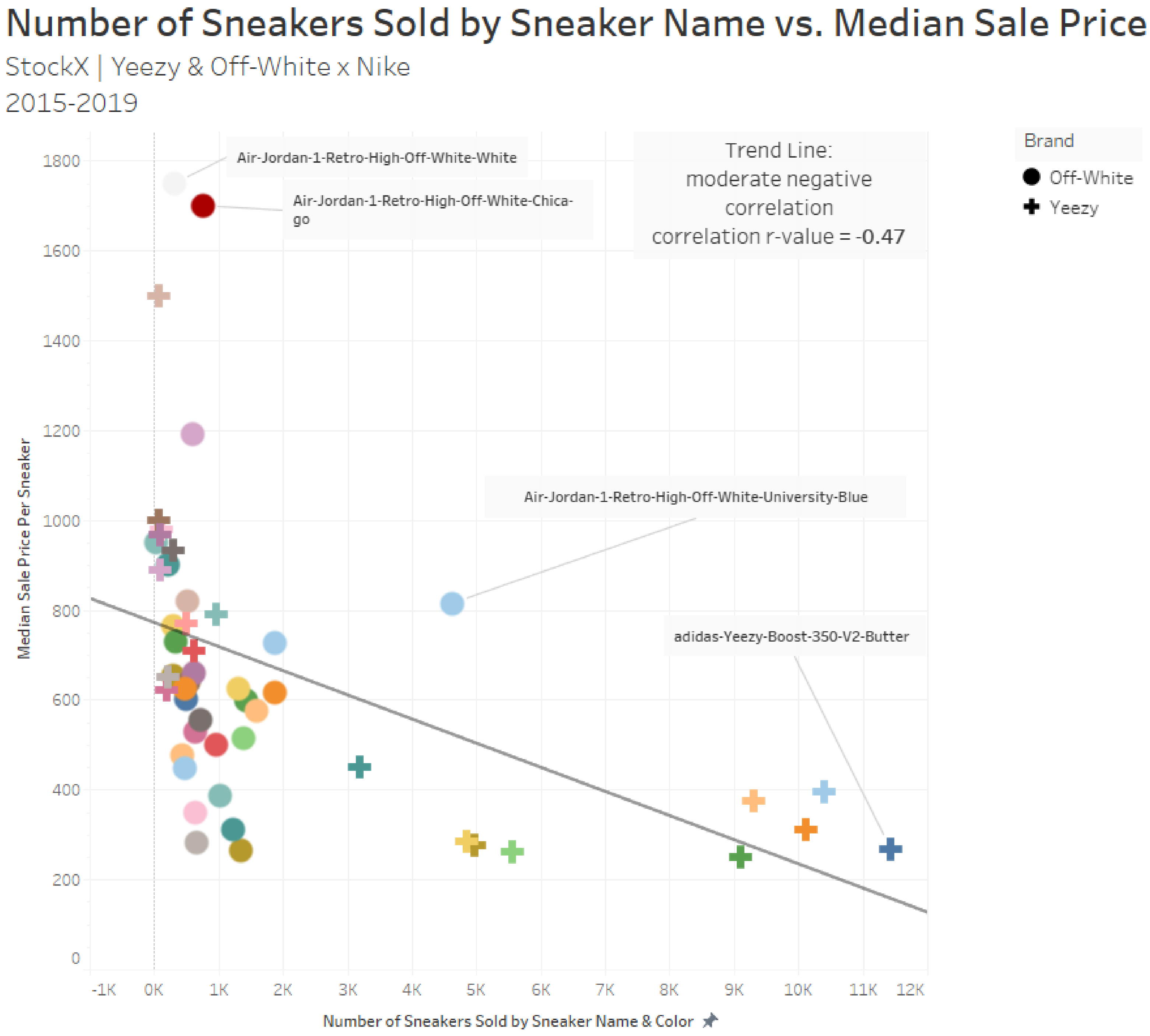


Sneakers and Sale price

- The more sneakers for sale, the smaller the profit margin, especially for the Yeezys
- 4 possible outliers for Off-White x Nike
- Overall, there are less Off-White x Nike sneakers made, which makes them more exclusive, hence the high median profit.



Sneakers and Sale price



PRE-PROCESS DATA

Pre-Process the Data

	Order Date	Brand	Sneaker Name	Sale Price	Retail Price	Release Date	Shoe Size	Buyer Region	Profit Margin	Colorway	Silhouette	Elapsed Time Days	Elapsed Time Weeks	Elapsed Time Years	Release Date Year	Release Date Month	Release Date Day	Order Date Year	Order Date Month	Order Date Day
0	2017-09-01	Yeezy	Adidas-Yeezy-Boost-350-Low-V2-Beluga	1097	220	2016-09-24	11.0	California	877	Beluga	V2	342	48.7	0.94	2016	9	24	2017	9	1
1	2017-09-01	Yeezy	Adidas-Yeezy-Boost-350-V2-Core-Black-	685	220	2016-11-23	11.0	California	465	Core-Black-Copper	V2	282	40.2	0.77	2016	11	23	2017	9	1

[9]:
```python
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Brand'] = le.fit_transform(df['Brand'])
df.head()
```

[9]:

	Brand	Sale Price	Retail Price	Shoe Size	Profit Margin	Silhouette	Elapsed Time Days
0	0	1097	220	11.0	877	V2	342
1	0	685	220	11.0	465	V2	282
2	0	690	220	11.0	470	V2	282
3	0	1075	220	11.5	855	V2	282
4	0	828	220	11.0	608	V2	202

[10]:
```python
print(le.classes_, le.transform(le.classes_))
```

```
[' Yeezy' 'Off-White'] [0 1]
```

- Dropped columns containing strings and datetime objects – machine learning can only use numerical data types
- Use LabelEncoder to change Brand to 0 or 1
- Used pd.get_dummies to factor in the silhouette
- Added a column “Price Ratio” (percent of retail price that the shoe resold for)

Fix Silhouette

[12]:
```python
df = pd.concat([df, pd.get_dummies(df['Silhouette'])], axis=1).drop('Silhouette', axis=1)
```

[13]:
```python
df.head()
```

[13]:

	Brand	Sale Price	Retail Price	Shoe Size	Profit Margin	Elapsed Time Days	Price Ratio	Air-Jordan-1-Retro-High	Nike-Air-Force-1-Low	Nike-Air-Force-1-Low-Virgil-Abloh	Nike-Air-Max-90	Nike-Air-Max-97	Nike-Air-Presto	Nike-Air-VaporMax	Nike-Blazer-Mid	Nike-React-Hyperdunk-2017-Flyknit	Nike-Zoom-Fly	Nike-Zoom-Fly-Mercurial	V1	V2
0	0	1097	220	11.0	877	342	4.986364	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	685	220	11.0	465	282	3.113636	0	0	0	0	0	0	0	0	0	0	0	0	1
2	0	690	220	11.0	470	282	3.136364	0	0	0	0	0	0	0	0	0	0	0	0	1
3	0	1075	220	11.5	855	282	4.886364	0	0	0	0	0	0	0	0	0	0	0	0	1
4	0	828	220	11.0	608	202	3.763636	0	0	0	0	0	0	0	0	0	0	0	0	1

BUILD MACHINE LEARNING MODELS & TRAIN THE MODEL

Training & Testing Sets

- Split data set into a training and testing set
- I did 80% training, 20% testing

```
X = df[['Brand', 'Shoe Size', 'const']]
y = df['Price Ratio']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2)
```

Build ML Model

- Ordinary Least Square Regression (OLS)
- Add a constant so regression line doesn’t have to pass through origin

```
df = sm.add_constant(df)
```

```
lr = sm.OLS(y_train, X_train).fit()
lr.summary()
```

EVALUATE THE MODEL
&
MAKE PREDICTIONS

X (dropped Price Ratio) vs y (Price Ratio)

Overfitted?

OLS Regression Results

Dep. Variable:	Price Ratio	**R-squared:**	0.994
Model:	OLS	**Adj. R-squared:**	0.994
Method:	Least Squares	**F-statistic:**	8.049e+05
Date:	Thu, 23 Mar 2023	**Prob (F-statistic):**	0.00
Time:	15:15:24	**Log-Likelihood:**	57391.
No. Observations:	79964	**AIC:**	-1.147e+05
Df Residuals:	79947	**BIC:**	-1.146e+05
Df Model:	16		
Covariance Type:	nonrobust		

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(predicted, y_test)
print(mse)
```

```
0.014100233924813736
```

	coef	std err	t	P>|t|	[0.025	0.975]
const	-4.018e+04	1.33e+05	-0.303	0.762	-3e+05	2.2e+05
Brand	-6897.7516	2.28e+04	-0.303	0.762	-5.15e+04	3.77e+04
Sale Price	1.197e+07	3.95e+07	0.303	0.762	-6.55e+07	8.94e+07
Retail Price	-1.197e+07	3.95e+07	-0.303	0.762	-8.94e+07	6.55e+07
Shoe Size	-0.0009	0.000	-4.759	0.000	-0.001	-0.001
Profit Margin	-1.197e+07	3.95e+07	-0.303	0.762	-8.94e+07	6.55e+07
Elapsed Time Days	2.387e-05	1.97e-06	12.145	0.000	2e-05	2.77e-05
Air-Jordan-1-Retro-High	-3497.8364	1.15e+04	-0.303	0.762	-2.61e+04	1.91e+04
Nike-Air-Force-1-Low	1826.6201	6027.311	0.303	0.762	-9986.872	1.36e+04
Nike-Air-Force-1-Low-Virgil-Abloh	7151.9998	2.36e+04	0.303	0.762	-3.91e+04	5.34e+04
Nike-Air-Max-90	4489.0224	1.48e+04	0.303	0.762	-2.45e+04	3.35e+04
Nike-Air-Max-97	-3497.8833	1.15e+04	-0.303	0.762	-2.61e+04	1.91e+04
Nike-Air-Presto	4489.2064	1.48e+04	0.303	0.762	-2.45e+04	3.35e+04
Nike-Air-VaporMax	-1.947e+04	6.42e+04	-0.303	0.762	-1.45e+05	1.06e+05
Nike-Blazer-Mid	1.248e+04	4.12e+04	0.303	0.762	-6.82e+04	9.32e+04
Nike-React-Hyperdunk-2017-Flyknit	-6160.1643	2.03e+04	-0.303	0.762	-4.6e+04	3.37e+04
Nike-Zoom-Fly	1826.4833	6027.311	0.303	0.762	-9987.008	1.36e+04
Nike-Zoom-Fly-Mercurial	-6160.1446	2.03e+04	-0.303	0.762	-4.6e+04	3.37e+04
V1	-1.306e+04	4.31e+04	-0.303	0.762	-9.75e+04	7.14e+04
V2	-1.838e+04	6.07e+04	-0.303	0.762	-1.37e+05	1.01e+05

X (dropped Price Ratio, Sale Price, & Retail Price, Profit Margin) **vs y** (Price Ratio)

Better. But is this too specific? What about other silhouettes?

OLS Regression Results

Dep. Variable:	Price Ratio	**R-squared:**	0.655
Model:	OLS	**Adj. R-squared:**	0.655
Method:	Least Squares	**F-statistic:**	1.086e+04
Date:	Thu, 23 Mar 2023	**Prob (F-statistic):**	0.00
Time:	15:19:11	**Log-Likelihood:**	-1.0346e+05
No. Observations:	79964	**AIC:**	2.069e+05
Df Residuals:	79949	**BIC:**	2.071e+05
Df Model:	14		
Covariance Type:	nonrobust		

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(predicted, y_test)
print(mse)
```

```
0.8165182917756894
```

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.3460	0.016	142.994	0.000	2.314	2.378
Brand	0.9904	0.017	58.584	0.000	0.957	1.024
Shoe Size	0.0234	0.001	17.358	0.000	0.021	0.026
Elapsed Time Days	0.0004	1.46e-05	24.917	0.000	0.000	0.000
Air-Jordan-1-Retro-High	1.7885	0.020	89.567	0.000	1.749	1.828
Nike-Air-Force-1-Low	-0.6039	0.024	-24.963	0.000	-0.651	-0.556
Nike-Air-Force-1-Low-Virgil-Abloh	3.0189	0.158	19.051	0.000	2.708	3.329
Nike-Air-Max-90	0.1303	0.026	5.062	0.000	0.080	0.181
Nike-Air-Max-97	0.2139	0.029	7.369	0.000	0.157	0.271
Nike-Air-Presto	1.1430	0.021	54.238	0.000	1.102	1.184
Nike-Air-VaporMax	-0.9984	0.022	-44.945	0.000	-1.042	-0.955
Nike-Blazer-Mid	1.0573	0.022	48.291	0.000	1.014	1.100
Nike-React-Hyperdunk-2017-Flyknit	-1.1306	0.043	-26.195	0.000	-1.215	-1.046
Nike-Zoom-Fly	-1.6443	0.023	-71.627	0.000	-1.689	-1.599
Nike-Zoom-Fly-Mercurial	-1.9843	0.030	-66.887	0.000	-2.042	-1.926
V1	2.3839	0.036	65.839	0.000	2.313	2.455
V2	-1.0284	0.013	-77.986	0.000	-1.054	-1.003

Brand & Shoe Size vs. Price Ratio

OLS Regression Results

Dep. Variable:	Price Ratio	R-squared:	0.426
Model:	OLS	Adj. R-squared:	0.426
Method:	Least Squares	F-statistic:	2.972e+04
Date:	Thu, 23 Mar 2023	Prob (F-statistic):	0.00
Time:	09:14:22	Log-Likelihood:	-1.2382e+05
No. Observations:	79964	AIC:	2.476e+05
Df Residuals:	79961	BIC:	2.477e+05
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Brand	2.1776	0.009	241.960	0.000	2.160	2.195
Shoe Size	0.0217	0.002	12.518	0.000	0.018	0.025
const	1.4387	0.017	86.008	0.000	1.406	1.472

```
'PriceRatio = 2.18*Brand + 0.02*ShoeSize + 1.44'
```

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(predicted, y_test)
print(mse)
```

```
1.3723087600999193
```

Square root of MSE = 1.17

- $R^2 = 0.426$ | 42.6% of the variance can be explained with the model
- $R = 0.65$ | moderate, positive linear correlation
- $\sqrt{MSE} = 1.17$ | the predicted price ratio is over or under predicting by 1.17 on average compared to the actual price ratio.

CONCLUSIONS

Conclusions

- Knowing the silhouette improves the model.
- But, if you wanted to predict about all Yeezys or all Off-Whites, we'd want a more general model

OLS Regression Results

Dep. Variable:	Price Ratio	**R-squared:**	0.655
Model:	OLS	**Adj. R-squared:**	0.655
Method:	Least Squares	**F-statistic:**	1.086e+04
Date:	Thu, 23 Mar 2023	**Prob (F-statistic):**	0.00
Time:	15:19:11	**Log-Likelihood:**	-1.0346e+05
No. Observations:	79964	**AIC:**	2.069e+05
Df Residuals:	79949	**BIC:**	2.071e+05
Df Model:	14		
Covariance Type:	nonrobust		

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(predicted, y_test)
print(mse)
```

```
0.8165182917756894
```

Conclusions

$$r^2 = 0.426$$
$$r = 0.65$$

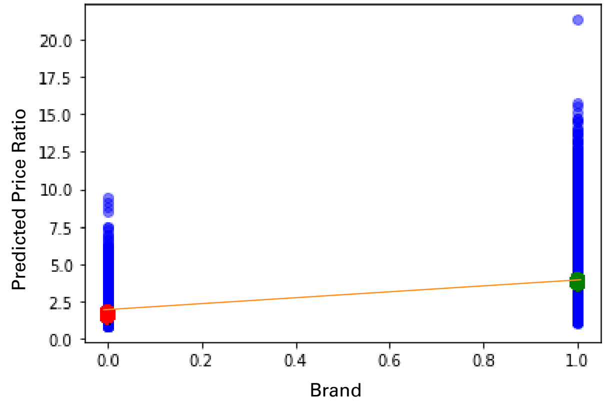
$$\sqrt{MSE} = 1.17$$

- Not confident about predicting profitability given brand and shoe size.
  - The r^2 and the MSE are the best fit I could find for the model.
  - The error is too high. If a shoe retails for $100, and the sneaker actually sales for $200 (price ratio of 2), then my model would, on average, have an error for price ratio of $\pm$1.17 above or below that actual value of 2. So price ratio of 3.17 or 0.83. It could have predicted the sale price to be $317 or $83. This is a big interval for sneaker prices.

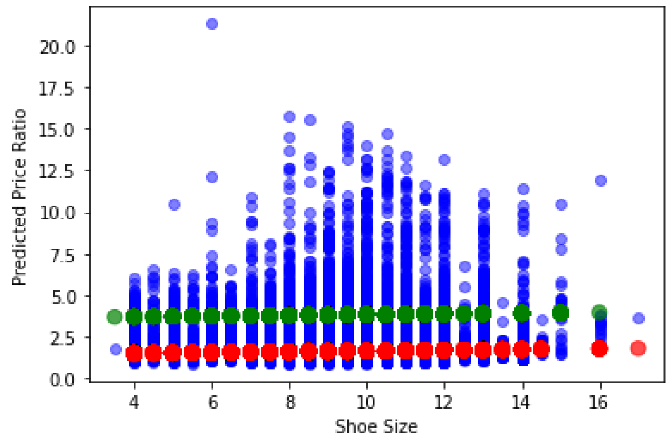
Conclusions

```
'PriceRatio = 2.18*Brand + 0.02*ShoeSize + 1.44'
```

- Influence of brand is very high in predicting price ratio.
  - When shoe is a 1 (Off-White Nike), the predicted price ratio goes up 218%
  - Off-White Nikes are generally more profitable than Yeezys.



- Shoe size not as much of an influence as expected, but it did improve the model.



Brand Comparison

Yeezy

```python
from sklearn.metrics import mean_squared_error
mean_squared_error(predicted, y_test)
```

```
0.3664584439006497
```

OLS Regression Results

Dep. Variable:	Price Ratio	**R-squared:**	0.188
Model:	OLS	**Adj. R-squared:**	0.188
Method:	Least Squares	**F-statistic:**	4465.
Date:	Thu, 23 Mar 2023	**Prob (F-statistic):**	0.00
Time:	07:12:01	**Log-Likelihood:**	-52717.
No. Observations:	57729	**AIC:**	1.054e+05
Df Residuals:	57725	**BIC:**	1.055e+05
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.2005	0.013	174.792	0.000	2.176	2.225
V1	2.9250	0.022	135.792	0.000	2.883	2.967
V2	-0.7245	0.011	-65.087	0.000	-0.746	-0.703
Shoe Size	0.0152	0.001	14.450	0.000	0.013	0.017

Off-White

OLS Regression Results

Dep. Variable:	Price Ratio	**R-squared:**	0.585
Model:	OLS	**Adj. R-squared:**	0.585
Method:	Least Squares	**F-statistic:**	2613.
Date:	Thu, 23 Mar 2023	**Prob (F-statistic):**	0.00
Time:	07:14:51	**Log-Likelihood:**	-35859.
No. Observations:	22235	**AIC:**	7.174e+04
	2222	**BIC:**	7.185e+04
	12		
	bust		

```python
from sklearn.metrics import mean_squared_error
mean_squared_error(predicted, y_test)
```

```
1.5100287620976034
```

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.3253	0.022	61.546	0.000	1.283	1.367
Brand	1.3253	0.022	61.546	0.000	1.283	1.367
Shoe Size	0.0429	0.004	11.341	0.000	0.036	0.050
Elapsed Time Days	0.0077	9.67e-05	79.261	0.000	0.007	0.008
Air-Jordan-1-Retro-High	1.9117	0.029	65.353	0.000	1.854	1.969
Nike-Air-Force-1-Low	-0.3720	0.035	-10.657	0.000	-0.440	-0.304
Nike-Air-Force-1-Low-Virgil-Abloh	3.0963	0.237	13.090	0.000	2.633	3.560
Nike-Air-Max-90	0.2711	0.037	7.401	0.000	0.199	0.343
Nike-Air-Max-97	0.1818	0.041	4.420	0.000	0.101	0.262
Nike-Air-Presto	1.2044	0.031	39.238	0.000	1.144	1.265
Nike-Air-VaporMax	-1.1603	0.032	-36.041	0.000	-1.223	-1.097
Nike-Blazer-Mid	1.1894	0.032	37.500	0.000	1.127	1.252
Nike-React-Hyperdunk-2017-Flyknit	-1.5243	0.062	-24.555	0.000	-1.646	-1.403
Nike-Zoom-Fly	-1.5503	0.033	-46.966	0.000	-1.615	-1.486
Nike-Zoom-Fly-Mercurial	-1.9225	0.042	-45.557	0.000	-2.005	-1.840

Any questions?

My questions:

- Can the model be improved to predict price ratio?
- Other types of linear regression besides Ordinary Least Squares?
- What if the data isn't linear? How do I deal with that?
- I need a review of statistics concepts like p-value, confidence interval, linear regression, mean-square error.

SNEAKERS

THANK YOU

Kaili