DS 223, Assignment #1

Bass Model

Shushan Gevorgyan

Libraries and Packages

In [1]:
```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
from scipy.optimize import least_squares
import plotly.express as px
import kaleido
from matplotlib import pyplot as plt
import matplotlib.image as mpimg
from IPython.display import Image, display
import numpy as np
```

Time Innovation: *ThredUp AI Search*

Similar Product: *Vinted*

In [7]:
```
display(Image(filename=r'img/header.png'))
```

A past innovation that resembles ThredUp AI Search is Vinted, the peer-to-peer online marketplace for second-hand clothing that has been popular in Europe since the late 2000s. Both platforms aim to simplify and accelerate the process of buying and selling pre-owned clothing online. Vinted allowed users to list items, browse through curated categories, and find products through text-based searches and filters. Functionally, Vinted pioneered the concept of accessible, large-scale second-hand shopping online, connecting buyers and sellers in a user-friendly interface and encouraging sustainable fashion practices.

ThredUp AI Search builds upon this concept with the use of modern artificial intelligence and computer vision. Unlike Vinted's primarily keyword- or category-based search, ThredUp lets users input ultra-specific phrases or upload images to find visually similar clothing items from millions of listings. This reduces guesswork and increases discoverability, enabling users who may be unfamiliar with brands or styles to shop sustainably with ease. While both innovations have expanded the second-hand fashion market, ThredUp's AI-driven approach represents a technological evolution, improving user experience and driving higher engagement, as seen in its reported 38% year-over-year increase in searches per session.

Data extraction

For this analysis, I sourced historical data on Vinted from Statista. The original data was provided in PPTX format as plots within a presentation, which required manual extraction. I was able to convert the visual data into Excel files for further processing. However, the Excel files I found only contained data up to 2021, which was insufficient for modeling the diffusion of the innovation. To address this, I combined the extracted historical data with more recent publicly available statistics to construct a complete time series suitable for Bass model estimation and forecasting. I was also able to find data showing downloads of Vinted in 2024 by countires, which supported my answer for question N *6*.

My main variable for the Bass model analysis is Gross Merchandise Volume (GMV) of Vinted worldwide from 2016 to 2024, measured in million USD. I also collected Revenue data, which served as an additional reference to validate the Bass model's predictive function. GMV was chosen as the primary variable because it reflects the total value of all transactions on the platform, capturing the overall scale, adoption, and market activity more directly than revenue alone. Revenue, while important for financial performance, depends on commission rates and business model specifics, which can fluctuate independently of user adoption. Therefore, GMV provides a better proxy for the diffusion and popularity of the platform across users, making it ideal for modeling adoption patterns using the Bass diffusion model.

Loading Data

In [8]:
```
gmv_path = 'data/GMV Vinted 2016-2024 .xlsx'
revenue_path = 'data/Revenue Vinted 2017-2024.xlsx'
```

```
downloads_by_country_path = 'data/Downloads by Country.xlsx'
gmv_df = pd.read_excel(gmv_path)
revenue_df = pd.read_excel(revenue_path)
downloads = pd.read_excel(downloads_by_country_path)
```

In [9]:
```
gmv_df.columns = gmv_df.columns.str.strip()
revenue_df.columns = revenue_df.columns.str.strip()

print(gmv_df)
print(revenue_df)
```

```
   Year      GMV
0  2016     29.5
1  2017    114.5
2  2018    506.6
3  2019   1154.3
4  2020   2424.3
5  2021   4829.5
6  2022   6487.2
7  2023  10720.0
8  2024  12564.9
   Year  Revenue
0  2017     10.0
1  2018     30.0
2  2019     84.0
3  2020    150.0
4  2021    245.3
5  2022    370.2
6  2023    596.3
7  2024    813.4
```

In [10]:
```
full_data = pd.merge(gmv_df, revenue_df, on='Year', how='inner')

print(full_data)
```

```
   Year      GMV  Revenue
0  2017    114.5     10.0
1  2018    506.6     30.0
2  2019   1154.3     84.0
3  2020   2424.3    150.0
4  2021   4829.5    245.3
5  2022   6487.2    370.2
6  2023  10720.0    596.3
7  2024  12564.9    813.4
```

Estimating the parameters for Bass Model

In [11]:
```
years = full_data['Year'].values
gmv = full_data['GMV'].values
rev = full_data['Revenue'].values
t = np.arange(len(years))
```

In [12]:
```
def bass_model(params, t, actual):
    p, q, M = params
    Y = np.zeros(len(t))
```

```
    S = np.zeros(len(t))
    for i in range(len(t)):
        if i == 0:
            S[i] = min(actual[0], M)
            Y[i] = S[i]
        else:
            S[i] = (p + q * Y[i-1]/M) * (M - Y[i-1])
            Y[i] = Y[i-1] + S[i]
    return S
```

In [13]:
```
def residuals(params, t, actual):
    return actual - bass_model(params, t, actual)
```

In [14]:
```
initial_params = [0.03, 0.4, max(gmv)*2]
bounds = ([0, 0, max(gmv)], [1, 1, 1e6])
```

Bass Model Parameteres based on GMV

In [15]:
```
result_gmv = least_squares(residuals, initial_params, bounds=bounds
p_hat_gmv, q_hat_gmv, M_hat_gmv = result_gmv.x


print(f"Estimated parameters(GMV):\np = {p_hat_gmv:.4f}\nq = {q_hat
```

```
Estimated parameters(GMV):
p = 0.0109
q = 0.8901
M = 55352
```

Bass Model Parameteres based on Revenue

In [16]:
```
result_rev = least_squares(residuals, initial_params, bounds = boun
p_hat_rev, q_hat_rev, M_hat_rev = result_rev.x
print(f"Estimated parameters(Revenue):\np = {p_hat_rev:.4f}\nq = {q
```

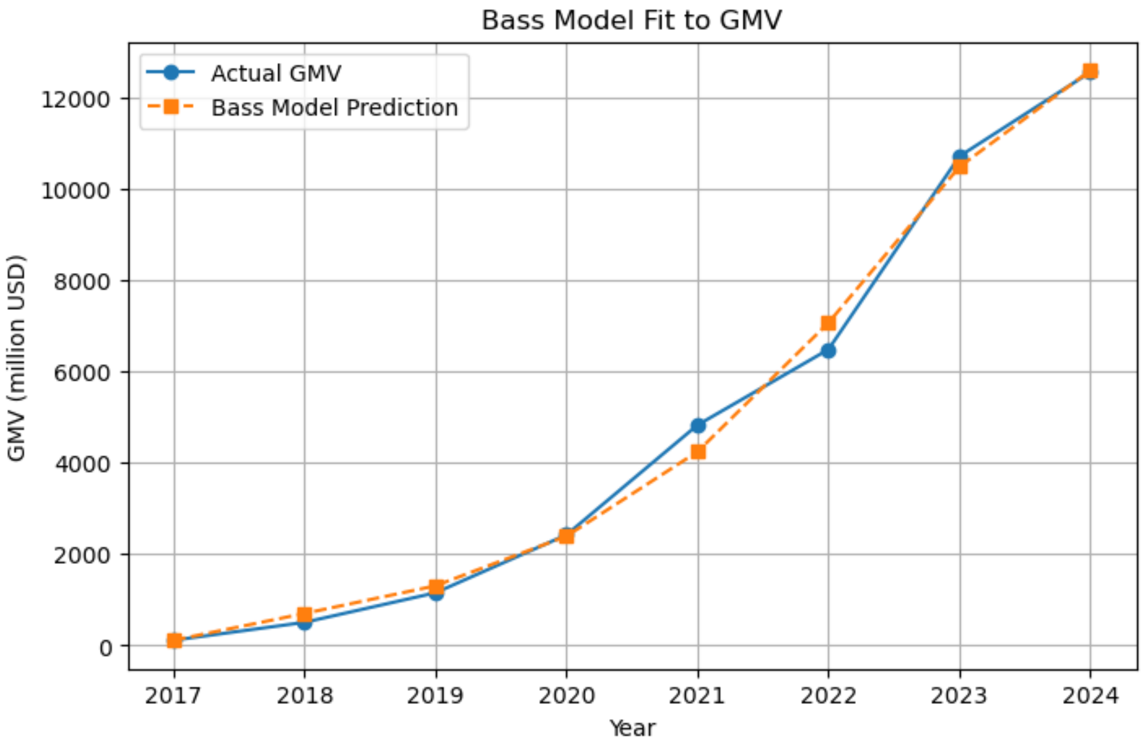
```
Estimated parameters(Revenue):
p = 0.0044
q = 0.5927
M = 12565
```

Forecast VS Real data

Plotting GMV model

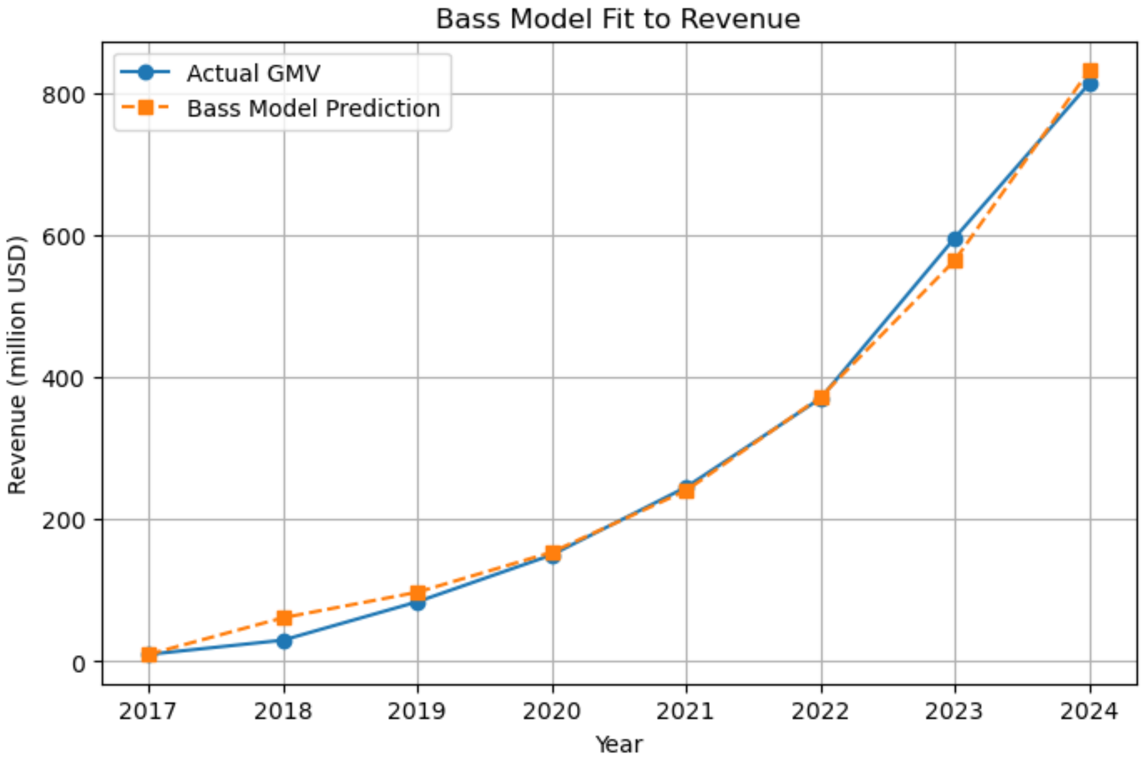
In [31]:
```
S_pred = bass_model([p_hat_gmv, q_hat_gmv, M_hat_gmv], t, gmv)
plt.figure(figsize=(8,5))
plt.plot(years, gmv, 'o-', label='Actual GMV')
plt.plot(years, S_pred, 's--', label='Bass Model Prediction')
plt.xlabel('Year')
plt.ylabel('GMV (million USD)')
```

```
plt.title('Bass Model Fit to GMV')
plt.legend()
plt.grid(True)
plt.savefig('img/bass_model_fit_gmv.png', dpi=300)
plt.show()
```



Plotting Revenue model

In [30]:
```
S_pred = bass_model([p_hat_rev, q_hat_rev, M_hat_rev], t, rev)
plt.figure(figsize=(8,5))
plt.plot(years, rev, 'o-', label='Actual GMV')
plt.plot(years, S_pred, 's--', label='Bass Model Prediction')
plt.xlabel('Year')
plt.ylabel('Revenue (million USD)')
plt.title('Bass Model Fit to Revenue')
plt.legend()
plt.grid(True)
plt.savefig('img/bass_model_fit_rev.png', dpi=300)
plt.show()
```



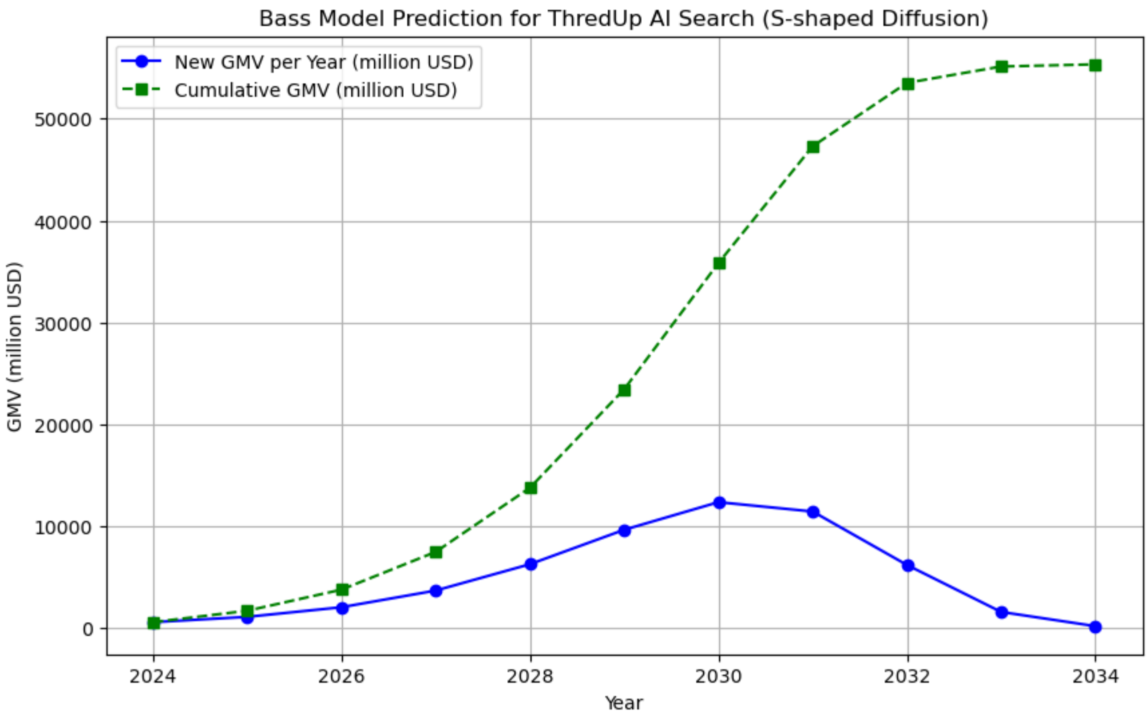
Future Forecasting for ThredUp AI Search

In [19]:
```
future_years = np.arange(2024, 2035)
T = len(future_years)

S_future = np.zeros(T)
Y_future = np.zeros(T)

for i in range(T):
    if i == 0:
        S_future[i] = p_hat_gmv * M_hat_gmv
        Y_future[i] = S_future[i]
    else:
        S_future[i] = (p_hat_gmv + q_hat_gmv * (Y_future[i-1] / M_h
        Y_future[i] = Y_future[i-1] + S_future[i]

# Create DataFrame for predictions
df_pred = pd.DataFrame({
    'Year': future_years,
    'New_GMV_million_USD': S_future.round(1),
    'Cumulative_GMV_million_USD': Y_future.round(1),
    'Cumulative_pct_of_M': (Y_future / M_hat_gmv * 100).round(1)
})

print(df_pred)
```

```
    Year  New_GMV_million_USD  Cumulative_GMV_million_USD  Cumulativ
e_pct_of_M
0   2024                600.9                       600.9
1.1
1   2025               1123.4                      1724.2
3.1
2   2026               2069.1                      3793.3
6.9
3   2027               3704.7                      7497.9
13.5
4   2028               6289.3                     13787.3
24.9
5   2029               9666.4                     23453.7
42.4
6   2030              12376.7                     35830.3
64.7
7   2031              11459.7                     47290.0
85.4
8   2032               6218.1                     53508.1
96.7
9   2033               1606.3                     55114.4
99.6
10  2034                212.9                     55327.3
100.0
```

In [28]:

```
plt.figure(figsize=(10,6))

plt.plot(df_pred['Year'], df_pred['New_GMV_million_USD'], marker='o

plt.plot(df_pred['Year'], df_pred['Cumulative_GMV_million_USD'], ma

plt.xlabel('Year')
plt.ylabel('GMV (million USD)')
plt.title('Bass Model Prediction for ThredUp AI Search (S-shaped Di
plt.legend()
plt.grid(True)
plt.savefig('img/thredup_difussion.png', dpi=300)
plt.show()
```



Global or Country-specific

In [21]:
```
downloads
```

Out[21]:

	Country	Downloads
0	NaN	NaN
1	United Kingdom	6.37
2	France	3.10
3	Italy	3.09
4	Germany	2.18
5	Poland	2.14
6	Spain	2.11
7	Romania	1.26
8	Sweden	1.19
9	Netherlands	0.97
10	Greece	0.97

In [22]:
```
downloads = downloads.dropna()
downloads.columns = downloads.columns.str.strip()
downloads
```

Out[22]:

	Country	Downloads
1	United Kingdom	6.37
2	France	3.10
3	Italy	3.09
4	Germany	2.18
5	Poland	2.14
6	Spain	2.11
7	Romania	1.26
8	Sweden	1.19
9	Netherlands	0.97
10	Greece	0.97

In [27]:

```
fig = px.choropleth(
    downloads,
    locations="Country",
    locationmode="country names",
    color="Downloads",
    color_continuous_scale="Reds",
    range_color=[0, downloads["Downloads"].max()],
    title="Vinted App Downloads in 2024 by Country (millions)"
)


fig.update_geos(
    scope="europe",
    fitbounds="locations",
    visible=False
)

fig.write_html('img/vinted_map.html')

fig.show()
```

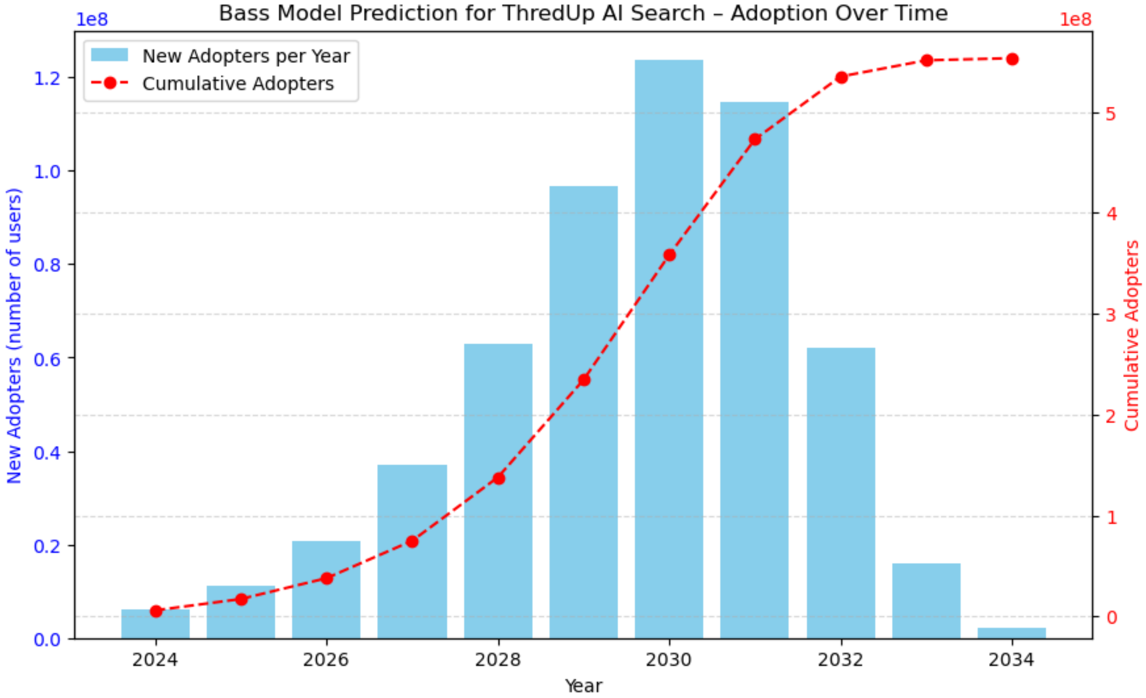
In 2024, Vinted app downloads were substantial across multiple countries, with the United Kingdom leading at 6.37 million downloads, followed by France (3.10 million), Italy (3.09 million), and Germany (2.18 million). Other countries such as Poland, Spain, Romania, Sweden, the Netherlands, and Greece collectively contributed millions more, reflecting strong international adoption. This distribution demonstrates that the secondhand marketplace is not confined to a single country but has significant usage across Europe. Consequently, analyzing the diffusion of innovations like ThredUp AI Search on a global scale is appropriate, as it captures the broad market potential and network effects evident in similar international platforms.

GMV to estimated adopters

In [24]:
```
avg_gmv_per_user = 0.0001  #as gmv is in million USD

df_pred['New_Adopters'] = (df_pred['New_GMV_million_USD'] / avg_gmv
df_pred['Cumulative_Adopters'] = (df_pred['Cumulative_GMV_million_U

print(df_pred[['Year','New_Adopters','Cumulative_Adopters']])
```

```
    Year  New_Adopters  Cumulative_Adopters
0   2024     6009000.0            6009000.0
1   2025    11234000.0           17242000.0
2   2026    20691000.0           37933000.0
3   2027    37047000.0           74979000.0
4   2028    62893000.0          137873000.0
5   2029    96664000.0          234537000.0
6   2030   123767000.0          358303000.0
7   2031   114597000.0          472900000.0
8   2032    62181000.0          535081000.0
9   2033    16063000.0          551144000.0
10  2034     2129000.0          553273000.0
```

In [26]:
```
fig, ax1 = plt.subplots(figsize=(10,6))

ax1.bar(df_pred['Year'], df_pred['New_Adopters'], color='skyblue',
ax1.set_xlabel('Year')
ax1.set_ylabel('New Adopters (number of users)', color='blue')
ax1.tick_params(axis='y', labelcolor='blue')

ax2 = ax1.twinx()
ax2.plot(df_pred['Year'], df_pred['Cumulative_Adopters'], marker='o
ax2.set_ylabel('Cumulative Adopters', color='red')
ax2.tick_params(axis='y', labelcolor='red')


lines_1, labels_1 = ax1.get_legend_handles_labels()
lines_2, labels_2 = ax2.get_legend_handles_labels()
ax1.legend(lines_1 + lines_2, labels_1 + labels_2, loc='upper left'

plt.title('Bass Model Prediction for ThredUp AI Search – Adoption O
plt.grid(True, which='both', linestyle='--', alpha=0.5)
plt.savefig('img/thredup_gmv_forecast.png', dpi=300)

plt.show()
```



Summary

Using Vinted as a look-alike innovation, we estimated the Bass diffusion model parameters for the marketplace GMV: $p = 0.0109$ (coefficient of innovation), $q = 0.8901$ (coefficient of imitation), and $M = 55{,}352$ million USD (market potential). These parameters indicate that adoption is heavily driven by social contagion and imitation, consistent with peer-to-peer marketplaces where word-of-mouth and network effects play a major role.

Applying these parameters to ThredUp AI Search, we forecast the GMV growth over the next decade. The model predicts a gradual start in 2024 with 600.9 million USD in new GMV, accelerating rapidly as the technology spreads: by 2027, cumulative GMV reaches 7,498 million USD (13.5% of market potential), and by 2030, adoption crosses 64.7% of M. Peak adoption occurs around 2032–2033, after which growth slows, approaching saturation at the total market potential of 55,352 million USD by 2034.

This diffusion path reflects a typical S-shaped adoption curve: slow initial uptake due to early adopters experimenting with AI-based second-hand shopping, followed by rapid growth as the tool gains awareness, and eventual plateau as most of the target market has adopted the innovation. The model highlights the potential for ThredUp AI Search to significantly accelerate second-hand fashion adoption, leveraging the same network-driven dynamics that fueled Vinted's success.

References

1. **Statista.** (2024). *Vinted: Study Overview*. Retrieved from https://www.statista.com/study/172216/vinted/

2. **Statista.** (2024). *Vinted App Downloads by Country*. Retrieved from https://www.statista.com/statistics/1447603/vinted-app-downloads-by-country/

3. **Time.** (2024). *Easier Secondhand Shopping: ThredUp AI Search*. Retrieved from https://time.com/7094866/thredup-ai-search/

4. **Vinted.** (2024). *How It Works*. Retrieved from https://www.vinted.com/how_it_works

5. **Course Slides.** (2024). *DS-223: Bass Model*. [PDF file]

6. **GeeksforGeeks.** (2025). *Bass Diffusion Model*. Retrieved from https://www.geeksforgeeks.org/machine-learning/bass-diffusion-model/