AI Project Documentation

Team: Tamashree Sarkar, Smaranika Mondal, Riddha Ghosh Dastidar, Prasmita Roy, Rohan Dey

Project 3: Sentiment Analysis on Flipkart E-commerce Products

Project Title: Sentiment Analysis using Python

Objective:

Develop a sentiment analysis tool to determine the sentiment (positive, negative, or neutral) of text data from Flipkart e-commerce product reviews.

Tools and Technologies:

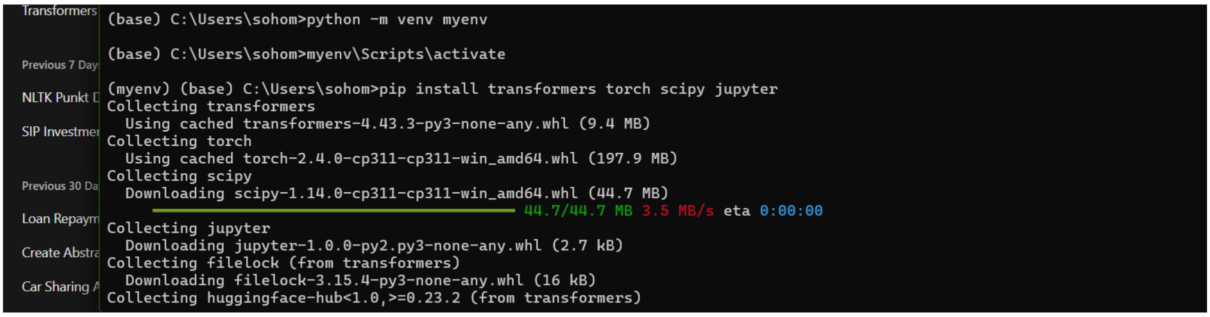
- **Python:** The primary programming language for the project.
- **Pandas:** A data manipulation and analysis library.
- **NumPy:** A library for numerical operations.
- **Scikit-learn:** A machine learning library for building and evaluating models.
- **Matplotlib:** A plotting library for creating static, animated, and interactive visualizations.
- **Seaborn:** A statistical data visualization library based on Matplotlib.

Project Description:

The project involves creating a sentiment analysis tool that can analyze text data from Flipkart product reviews and classify it into positive, negative, or neutral sentiment. The tool will also include data visualization to understand the sentiment distribution and key insights from the reviews.

Steps:

1. **Environment Setup:**
    - **Install Python and necessary libraries:**
        - **Ensure Python is installed on your system.**



```
(base) C:\Users\sohom>python -m venv myenv

(base) C:\Users\sohom>myenv\Scripts\activate

(myenv) (base) C:\Users\sohom>pip install transformers torch scipy jupyter
Collecting transformers
  Using cached transformers-4.43.3-py3-none-any.whl (9.4 MB)
Collecting torch
  Using cached torch-2.4.0-cp311-cp311-win_amd64.whl (197.9 MB)
Collecting scipy
  Downloading scipy-1.14.0-cp311-cp311-win_amd64.whl (44.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.7/44.7 MB 3.5 MB/s eta 0:00:00
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting filelock (from transformers)
  Downloading filelock-3.15.4-py3-none-any.whl (16 kB)
Collecting huggingface-hub<1.0,>=0.23.2 (from transformers)
```

Set up a virtual environment:

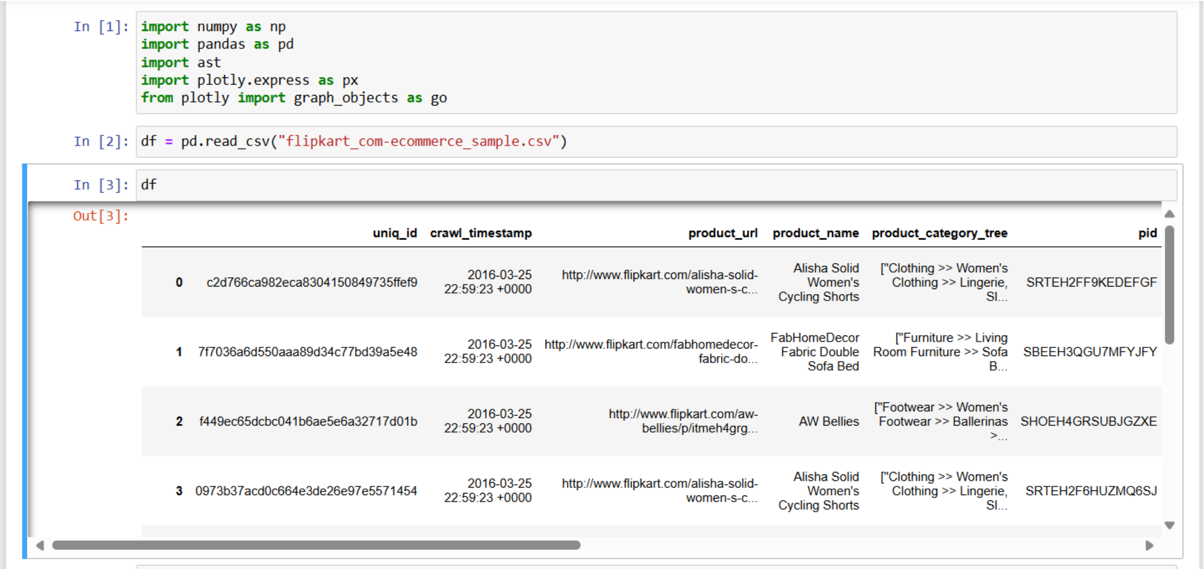
- Create and activate a virtual environment to manage project dependencies

2. **Data Collection:**

- **Collect text data from Flipkart product reviews:**
    - Scrape product review data from Flipkart using web scraping tools like Scrapy.

  - Alternatively, download datasets from platforms that provide e-commerce review data.

3. Data Preprocessing:
   - Load the data:
     - Use Pandas to load the database



```
In [1]: import numpy as np
        import pandas as pd
        import ast
        import plotly.express as px
        from plotly import graph_objects as go
```

```
In [2]: df = pd.read_csv("flipkart_com-ecommerce_sample.csv")
```

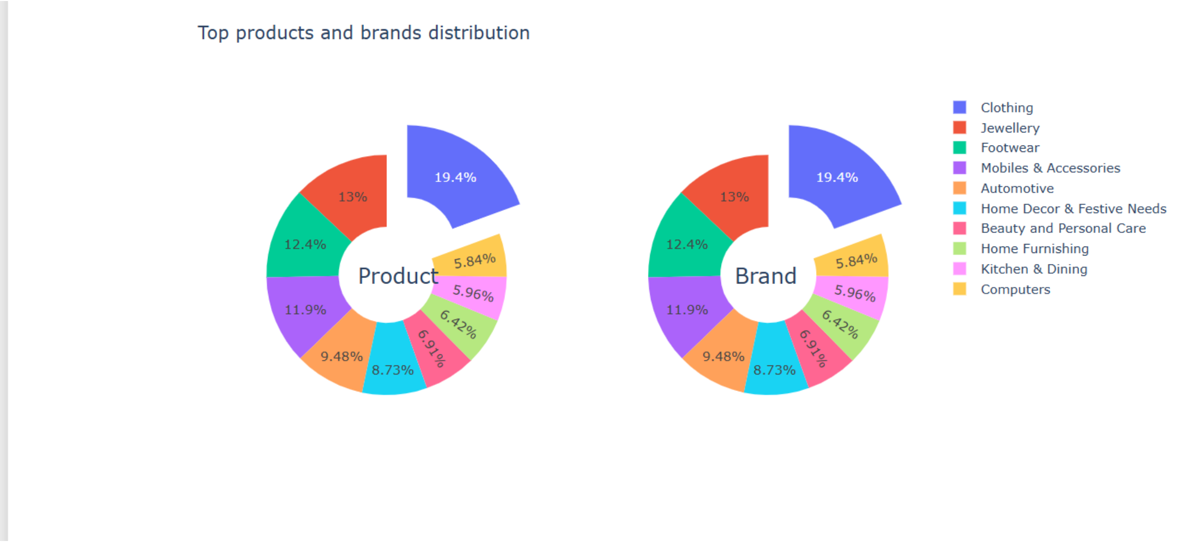
```
In [3]: df
```

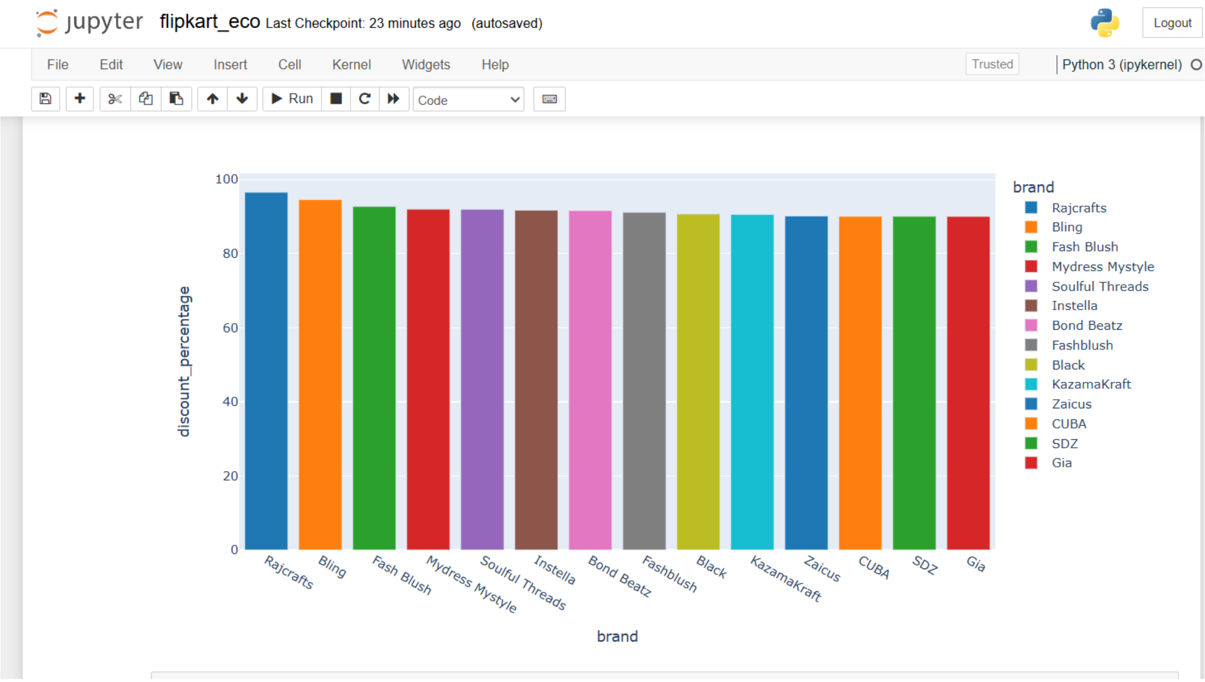
Out[3]:

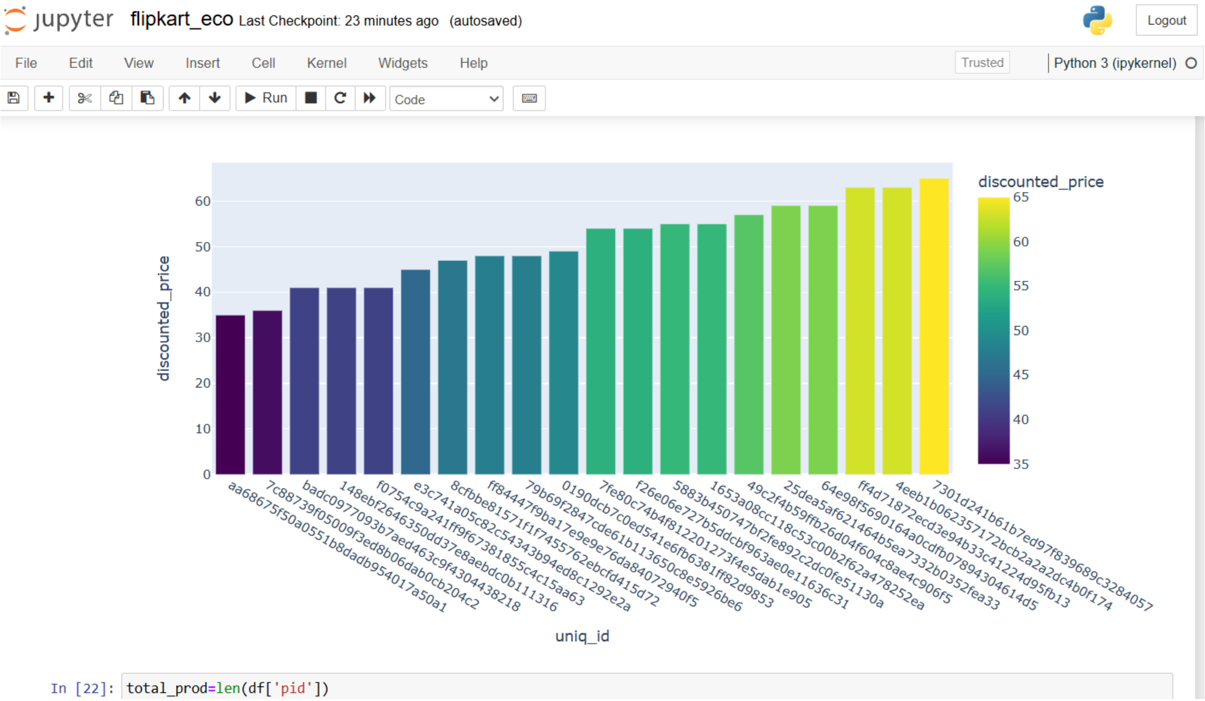
	uniq_id	crawl_timestamp	product_url	product_name	product_category_tree	pid
0	c2d766ca982eca8304150849735ffef9	2016-03-25 22:59:23 +0000	http://www.flipkart.com/alisha-solid-women-s-c...	Alisha Solid Women's Cycling Shorts	["Clothing >> Women's Clothing >> Lingerie, Sl...	SRTEH2FF9KEDEFGF
1	7f7036a6d550aaa89d34c77bd39a5e48	2016-03-25 22:59:23 +0000	http://www.flipkart.com/fabhomedecor-fabric-do...	FabHomeDecor Fabric Double Sofa Bed	["Furniture >> Living Room Furniture >> Sofa B...	SBEEH3QGU7MFYJFY
2	f449ec65dcbc041b6ae5e6a32717d01b	2016-03-25 22:59:23 +0000	http://www.flipkart.com/aw-bellies/p/itmeh4grg...	AW Bellies	["Footwear >> Women's Footwear >> Ballerinas >...	SHOEH4GRSUBJGZXE
3	0973b37acd0c664e3de26e97e5571454	2016-03-25 22:59:23 +0000	http://www.flipkart.com/alisha-solid-women-s-c...	Alisha Solid Women's Cycling Shorts	["Clothing >> Women's Clothing >> Lingerie, Sl...	SRTEH2F6HUZMQ6SJ

4. **Exploratory Data Analysis (EDA):**

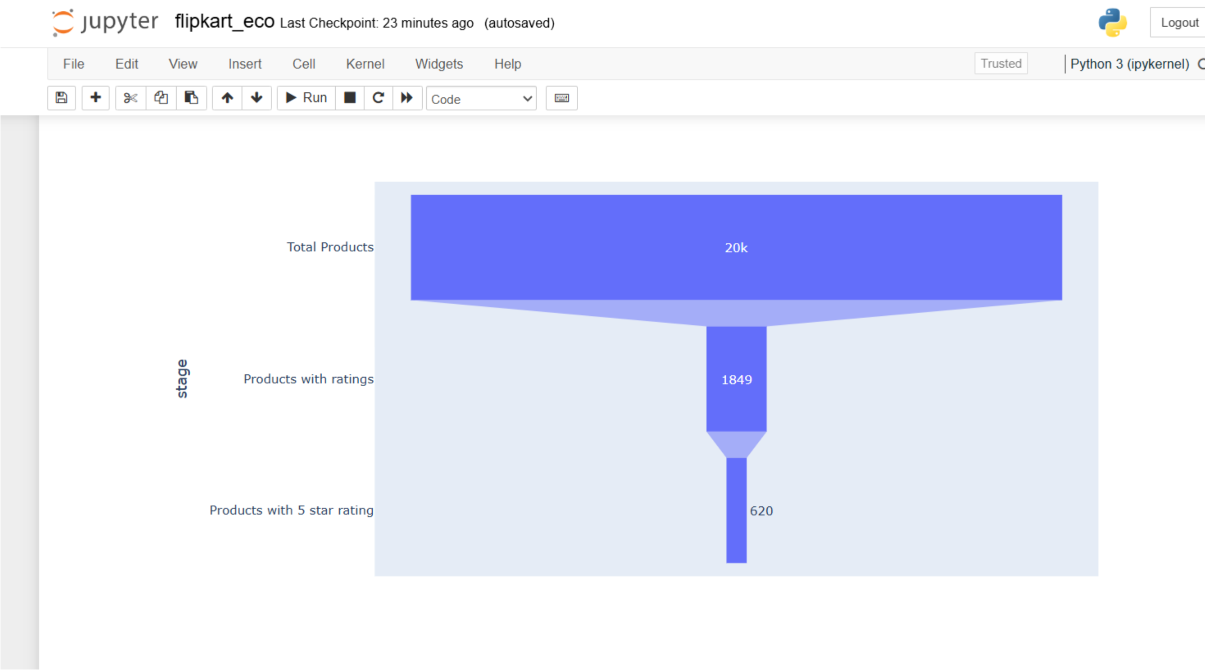
   - **Visualize the distribution of sentiments:**
     - Create bar plots or pie charts to show the proportion of positive, negative, and neutral reviews:
   - **DataVisualization:**






jupyter
flipkart_eco Last Checkpoint: 23 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
60
50
40
30
20
10
0
discounted_price
aa68675f50a0551b8dadb954017a50a1
7c88739f05009f3ed8b06dab0cb204c2
badc0977093b7aed463c9f4304438218
148ebf2646350dd37e8aebdc0b111316
f0754c9a241ff9f67381855c4c15aa63
e3c741a05c82c54343b94ed8c1292e2a
8cfbbe81571f1f7455762ebcfd415d72
ff84447f9ba17e9e9e76da84072940f5
79b69f2847cde61b113650c8e5926be6
0190dcb7c0ed541e6fb6381ff82d9853
7fe80c74b4f812201273f4e5dab1e905
f26e06e727b5ddcbf963ae0e11636c31
5883b450747bf2fe892c2dc0fe51130a
1653a08cc118c53c00b2f62a478252ea
49c2f4b59ffb26d04f604c8ae4c906f5
25dea5af621464b5ea7332b0352fea33
64e98f5690164a0cdfb07894304614d5
ff4d71872ecd3e94b33c41224d95fb13
4eeb1b062357172bcb2a2a2dc4b0f174
7301d241b61b7ed97f839689c3284057
uniq_id
discounted_price
65
60
55
50
45
40
35

```
In [22]: total_prod=len(df['pid'])
```



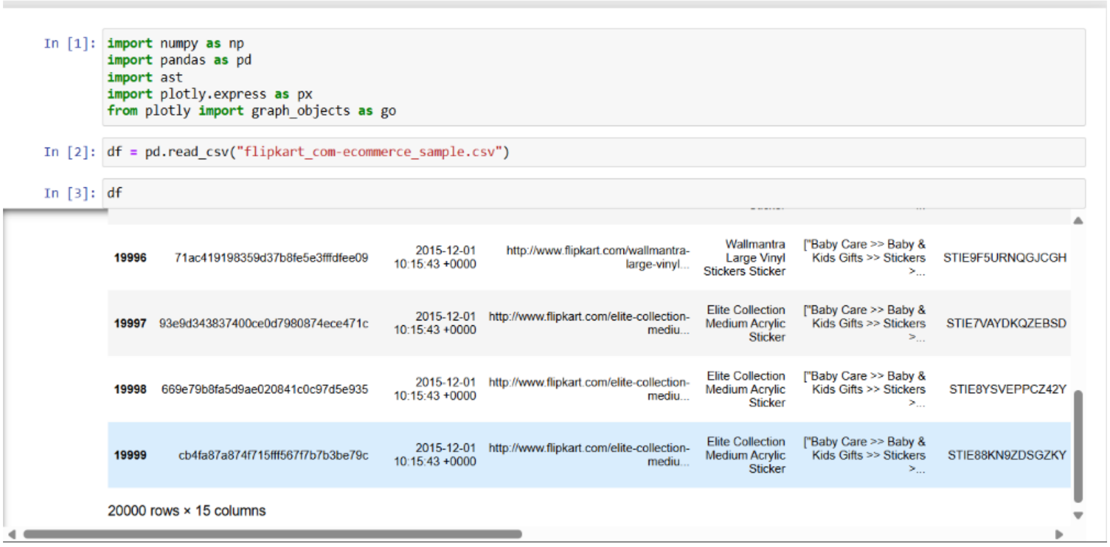
Model Training:

Use machine learning algorithms for sentiment classification:

- Apply algorithms like Naive Bayes or Logistic Regression to classify sentiment
- Calculate the accuracy of the model and generate a confusion matrix to understand its performance

Expected Outcome:

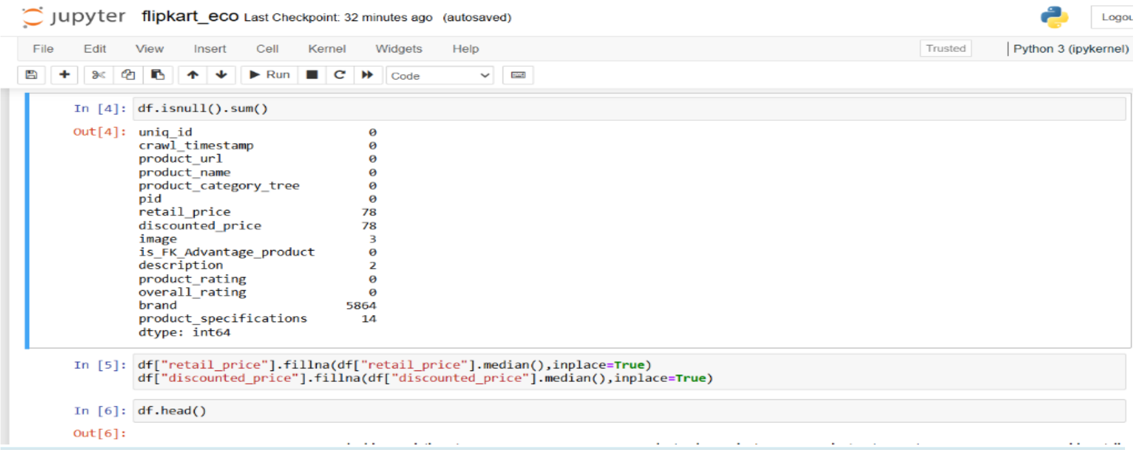
A functional sentiment analysis tool that can classify text data into positive, negative, or neutral sentiment. Users will be able to input text and receive the sentiment classification through a web interface.



```
In [1]: import numpy as np
        import pandas as pd
        import ast
        import plotly.express as px
        from plotly import graph_objects as go

In [2]: df = pd.read_csv("flipkart_com-ecommerce_sample.csv")

In [3]: df
```

19996	71ac419198359d37b8fe5e3fffdfee09	2015-12-01 10:15:43 +0000	http://www.flipkart.com/wallmantra-large-vinyl...	Wallmantra Large Vinyl Stickers Sticker	["Baby Care >> Baby & Kids Gifts >> Stickers >...	STIE9F5URNQGJCGH
19997	93e9d343837400ce0d7980874ece471c	2015-12-01 10:15:43 +0000	http://www.flipkart.com/elite-collection-mediu...	Elite Collection Medium Acrylic Sticker	["Baby Care >> Baby & Kids Gifts >> Stickers >...	STIE7VAYDKQZEBSD
19998	669e79b8fa5d9ae020841c0c97d5e935	2015-12-01 10:15:43 +0000	http://www.flipkart.com/elite-collection-mediu...	Elite Collection Medium Acrylic Sticker	["Baby Care >> Baby & Kids Gifts >> Stickers >...	STIE8YSVEPPCZ42Y
19999	cb4fa87a874f715fff567f7b7b3be79c	2015-12-01 10:15:43 +0000	http://www.flipkart.com/elite-collection-mediu...	Elite Collection Medium Acrylic Sticker	["Baby Care >> Baby & Kids Gifts >> Stickers >...	STIE88KN9ZDSGZKY

20000 rows × 15 columns


jupyter flipkart_eco Last Checkpoint: 32 minutes ago (autosaved)
Logou
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

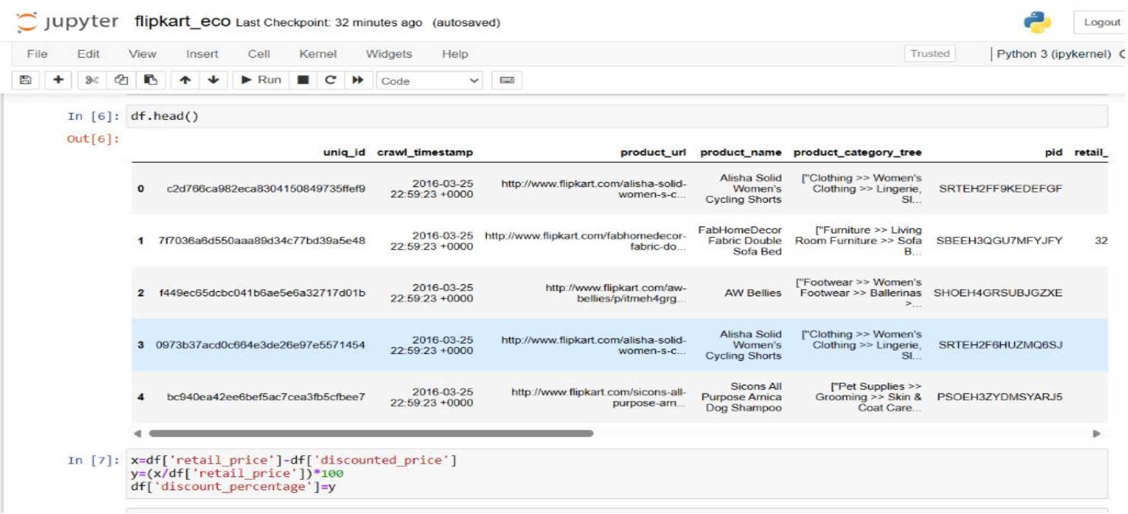
In [4]:
```
df.isnull().sum()
```

Out[4]:
```
uniq_id                       0
crawl_timestamp               0
product_url                   0
product_name                  0
product_category_tree         0
pid                           0
retail_price                 78
discounted_price             78
image                         3
is_FK_Advantage_product       0
description                   2
product_rating                0
overall_rating                0
brand                      5864
product_specifications       14
dtype: int64
```

In [5]:
```
df["retail_price"].fillna(df["retail_price"].median(),inplace=True)
df["discounted_price"].fillna(df["discounted_price"].median(),inplace=True)
```

In [6]:
```
df.head()
```

Out[6]:

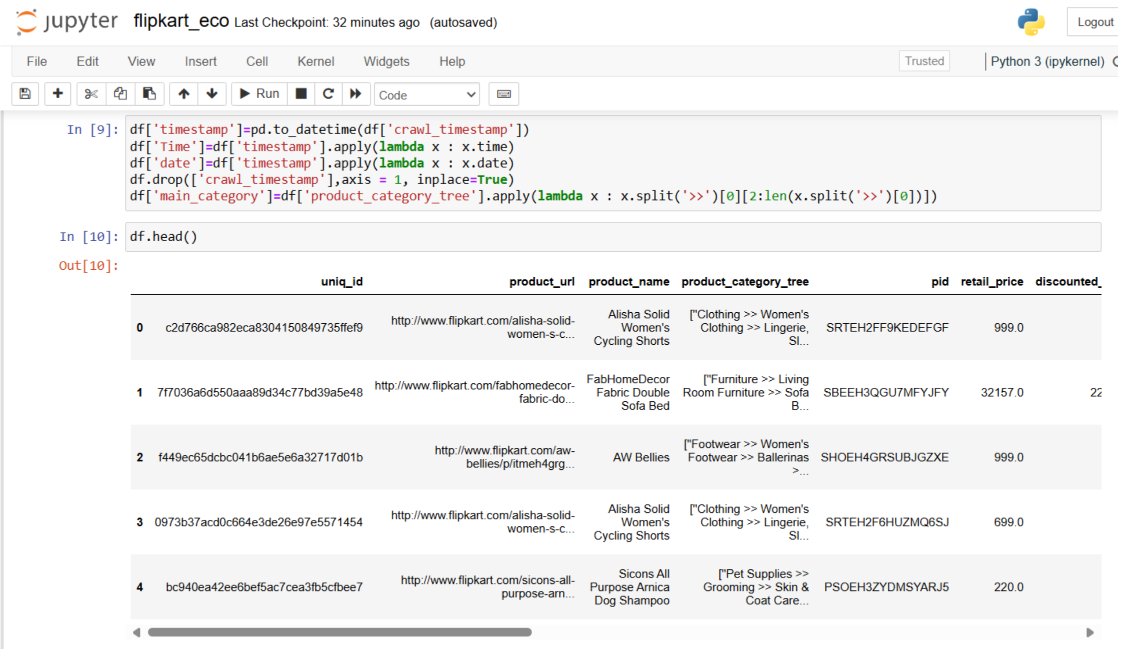

jupyter flipkart_eco Last Checkpoint: 32 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

In [6]:
```
df.head()
```

Out[6]:

	uniq_id	crawl_timestamp	product_url	product_name	product_category_tree	pid	retail_
0	c2d766ca982eca8304150849735ffef9	2016-03-25 22:59:23 +0000	http://www.flipkart.com/alisha-solid-women-s-c...	Alisha Solid Women's Cycling Shorts	["Clothing >> Women's Clothing >> Lingerie, Sl...	SRTEH2FF9KEDEFGF	
1	7f7036a6d550aaa89d34c77bd39a5e48	2016-03-25 22:59:23 +0000	http://www.flipkart.com/fabhomedecor-fabric-do...	FabHomeDecor Fabric Double Sofa Bed	["Furniture >> Living Room Furniture >> Sofa B...	SBEEH3QGU7MFYJFY	32
2	f449ec65dcbc041b6ae5e6a32717d01b	2016-03-25 22:59:23 +0000	http://www.flipkart.com/aw-bellies/p/itmeh4grg...	AW Bellies	["Footwear >> Women's Footwear >> Ballerinas >...	SHOEH4GRSUBJGZXE	
3	0973b37acd0c664e3de26e97e5571454	2016-03-25 22:59:23 +0000	http://www.flipkart.com/alisha-solid-women-s-c...	Alisha Solid Women's Cycling Shorts	["Clothing >> Women's Clothing >> Lingerie, Sl...	SRTEH2F6HUZMQ6SJ	
4	bc940ea42ee6bef5ac7cea3fb5cfbee7	2016-03-25 22:59:23 +0000	http://www.flipkart.com/sicons-all-purpose-arn...	Sicons All Purpose Arnica Dog Shampoo	["Pet Supplies >> Grooming >> Skin & Coat Care...	PSOEH3ZYDMSYARJ5	

In [7]:
```
x=df['retail_price']-df['discounted_price']
y=(x/df['retail_price'])*100
df['discount_percentage']=y
```

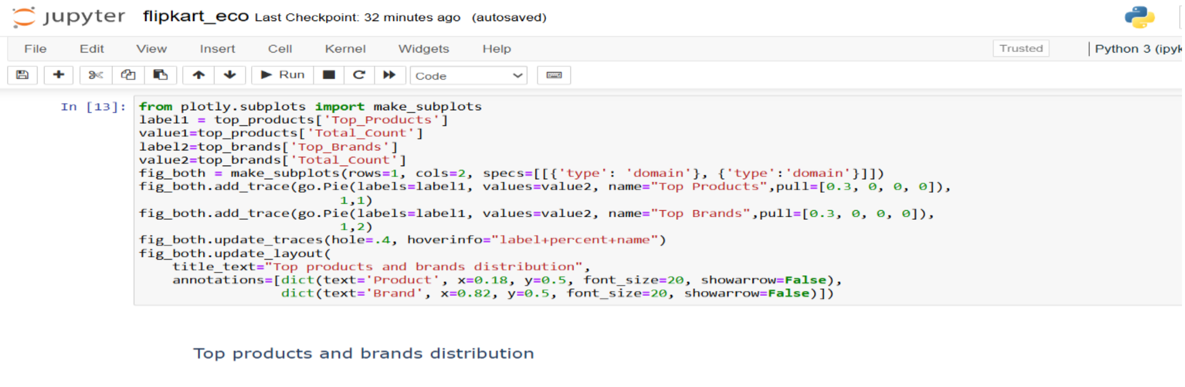

jupyter flipkart_eco Last Checkpoint: 32 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

In [9]:
```
df['timestamp']=pd.to_datetime(df['crawl_timestamp'])
df['Time']=df['timestamp'].apply(lambda x : x.time)
df['date']=df['timestamp'].apply(lambda x : x.date)
df.drop(['crawl_timestamp'],axis = 1, inplace=True)
df['main_category']=df['product_category_tree'].apply(lambda x : x.split('>>')[0][2:len(x.split('>>')[0])])
```

In [10]:
```
df.head()
```

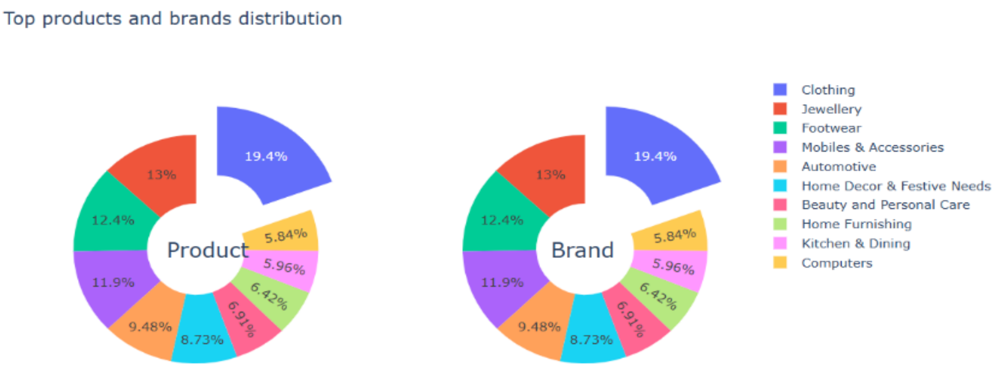
Out[10]:

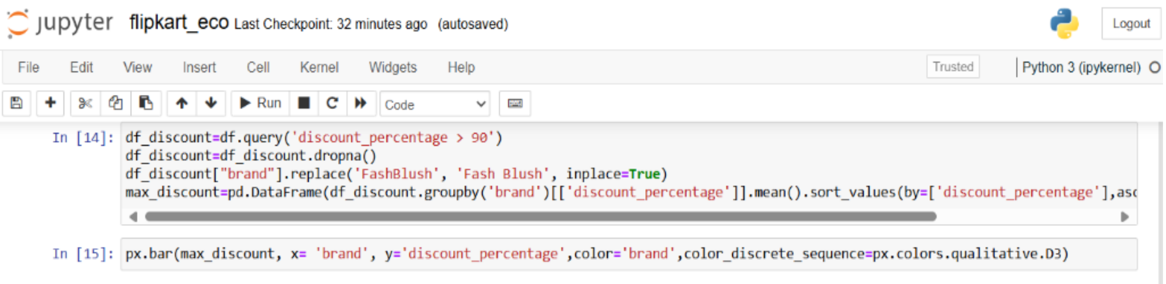
	uniq_id	product_url	product_name	product_category_tree	pid	retail_price	discounted_
0	c2d766ca982eca8304150849735ffef9	http://www.flipkart.com/alisha-solid-women-s-c...	Alisha Solid Women's Cycling Shorts	["Clothing >> Women's Clothing >> Lingerie, Sl...	SRTEH2FF9KEDEFGF	999.0	
1	7f7036a6d550aaa89d34c77bd39a5e48	http://www.flipkart.com/fabhomedecor-fabric-do...	FabHomeDecor Fabric Double Sofa Bed	["Furniture >> Living Room Furniture >> Sofa B...	SBEEH3QGU7MFYJFY	32157.0	22
2	f449ec65dcbc041b6ae5e6a32717d01b	http://www.flipkart.com/aw-bellies/p/itmeh4grg...	AW Bellies	["Footwear >> Women's Footwear >> Ballerinas >...	SHOEH4GRSUBJGZXE	999.0	
3	0973b37acd0c664e3de26e97e5571454	http://www.flipkart.com/alisha-solid-women-s-c...	Alisha Solid Women's Cycling Shorts	["Clothing >> Women's Clothing >> Lingerie, Sl...	SRTEH2F6HUZMQ6SJ	699.0	
4	bc940ea42ee6bef5ac7cea3fb5cfbee7	http://www.flipkart.com/sicons-all-purpose-arn...	Sicons All Purpose Arnica Dog Shampoo	["Pet Supplies >> Grooming >> Skin & Coat Care...	PSOEH3ZYDMSYARJ5	220.0	



```
In [13]: from plotly.subplots import make_subplots
         label1 = top_products['Top_Products']
         value1=top_products['Total_Count']
         label2=top_brands['Top_Brands']
         value2=top_brands['Total_Count']
         fig_both = make_subplots(rows=1, cols=2, specs=[[{'type': 'domain'}, {'type':'domain'}]])
         fig_both.add_trace(go.Pie(labels=label1, values=value2, name="Top Products",pull=[0.3, 0, 0, 0]),
                         1,1)
         fig_both.add_trace(go.Pie(labels=label1, values=value2, name="Top Brands",pull=[0.3, 0, 0, 0]),
                         1,2)
         fig_both.update_traces(hole=.4, hoverinfo="label+percent+name")
         fig_both.update_layout(
             title_text="Top products and brands distribution",
             annotations=[dict(text='Product', x=0.18, y=0.5, font_size=20, showarrow=False),
                          dict(text='Brand', x=0.82, y=0.5, font_size=20, showarrow=False)])
```

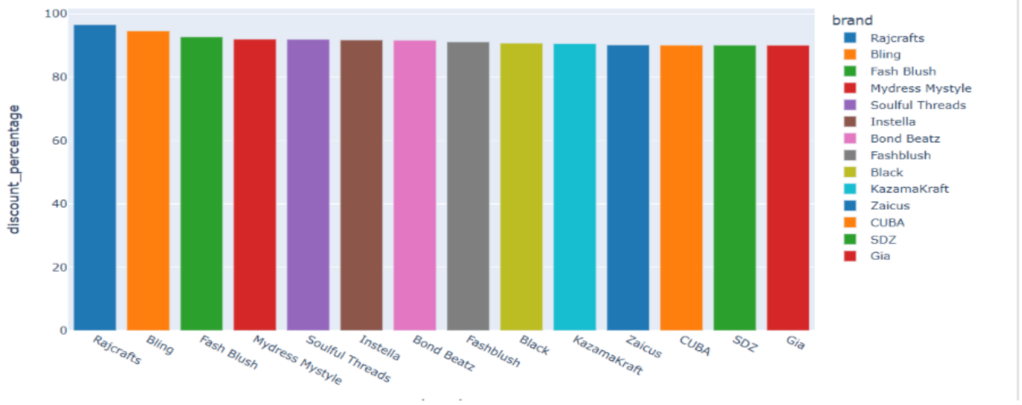
Top products and brands distribution

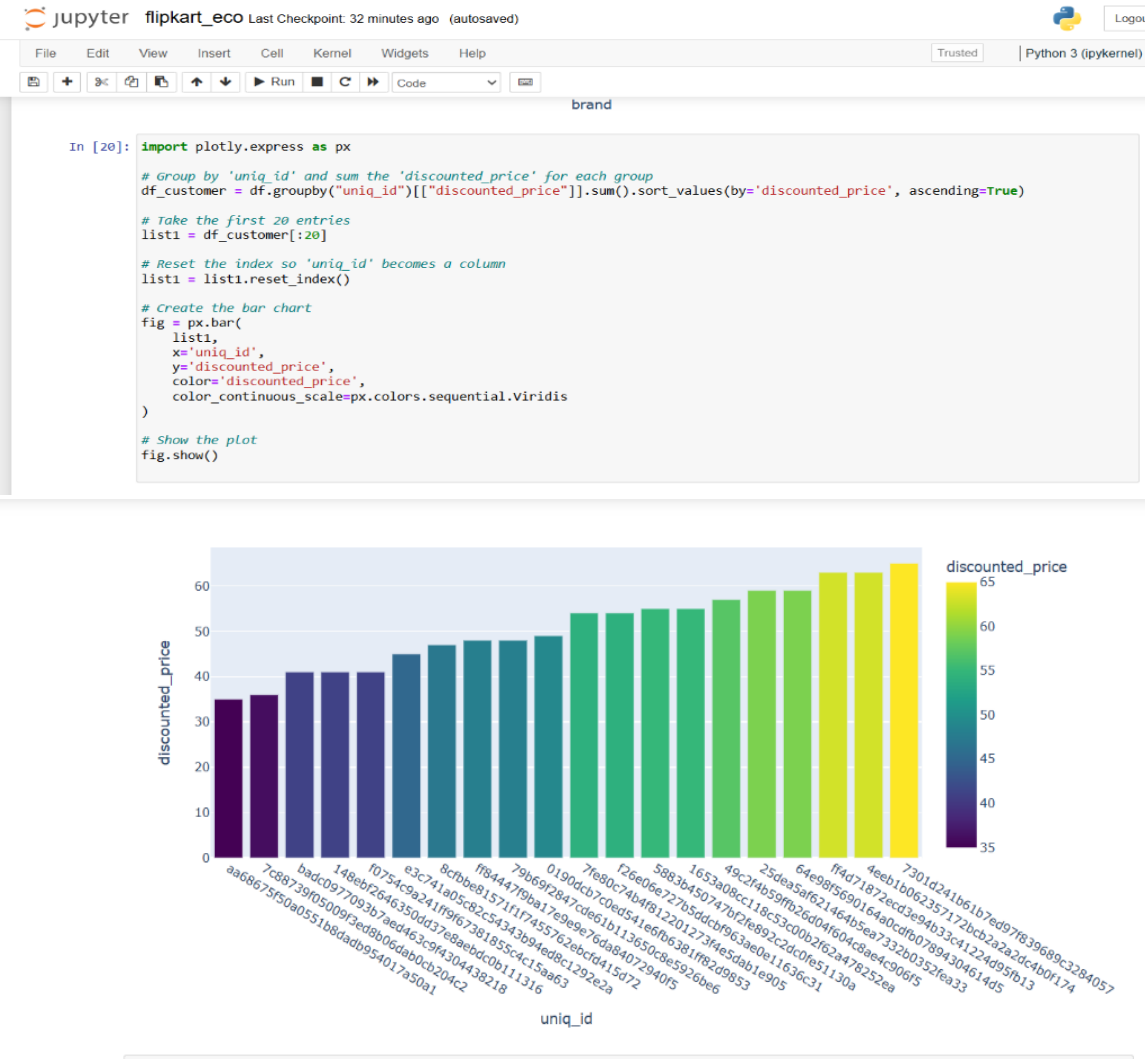




```
In [14]: df_discount=df.query('discount_percentage > 90')
         df_discount=df_discount.dropna()
         df_discount["brand"].replace('FashBlush', 'Fash Blush', inplace=True)
         max_discount=pd.DataFrame(df_discount.groupby('brand')[['discount_percentage']].mean().sort_values(by=['discount_percentage'],asc

In [15]: px.bar(max_discount, x= 'brand', y='discount_percentage',color='brand',color_discrete_sequence=px.colors.qualitative.D3)
```

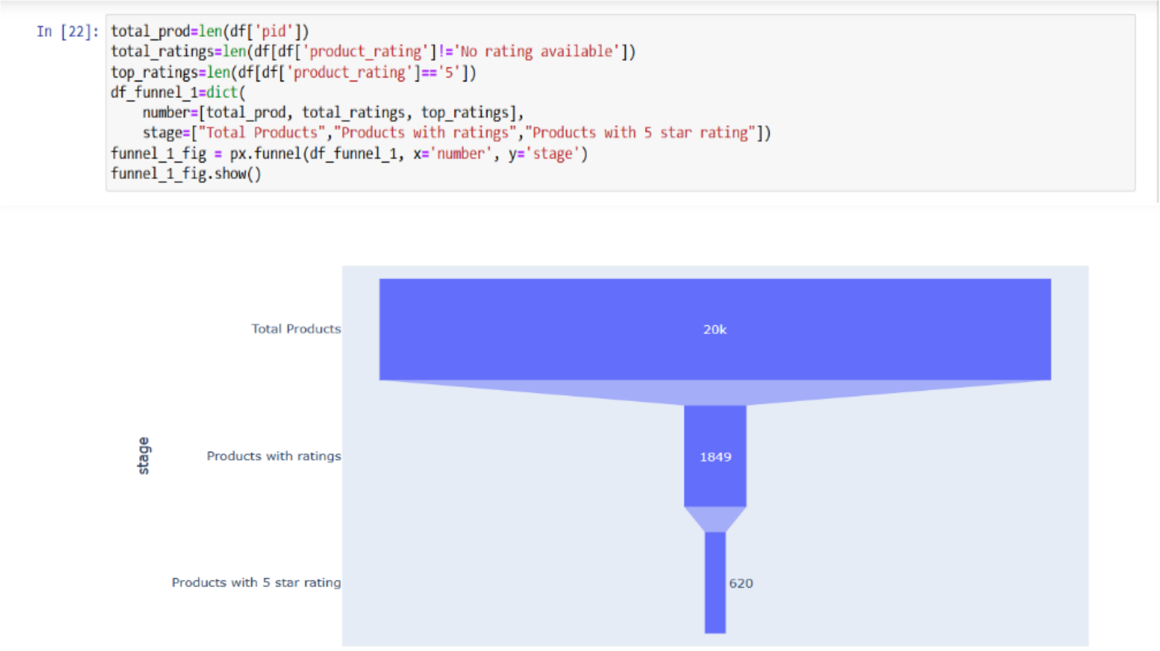
In [15]: px.bar(max_discount, x= 'brand', y='discount_percentage',color='brand',color_discrete_sequence=px.colors.qualitative.D3)





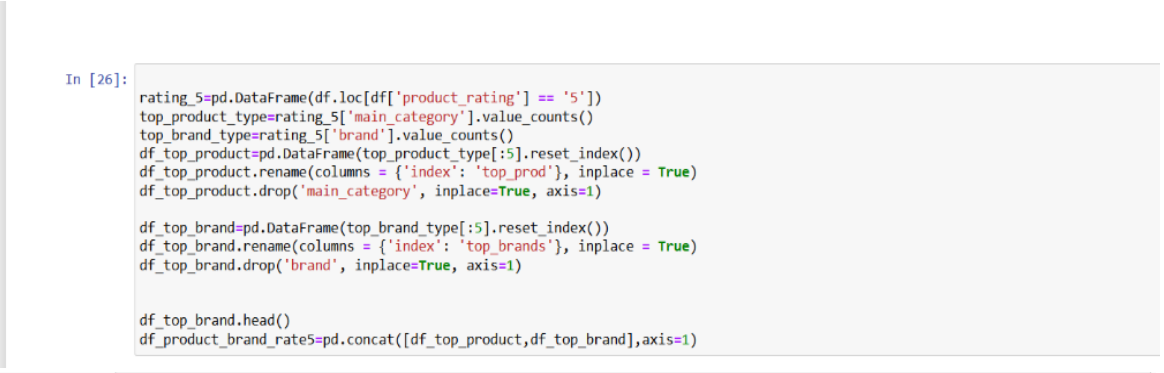
In [20]:
```
import plotly.express as px

# Group by 'uniq_id' and sum the 'discounted_price' for each group
df_customer = df.groupby("uniq_id")[["discounted_price"]].sum().sort_values(by='discounted_price', ascending=True)

# Take the first 20 entries
list1 = df_customer[:20]

# Reset the index so 'uniq_id' becomes a column
list1 = list1.reset_index()

# Create the bar chart
fig = px.bar(
    list1,
    x='uniq_id',
    y='discounted_price',
    color='discounted_price',
    color_continuous_scale=px.colors.sequential.Viridis
)

# Show the plot
fig.show()
```

discounted_price
60
50
40
30
20
10
0
aa68675f50a0551b8dadb954017a50a1
7c88739f05009f3ed8b06dab0cb204c2
badc0977093b7aed463c9f4304438218
148ebf2646350dd37e8aebdc0b111316
f0754c9a241ff9f67381855c4c15aa63
e3c741a05c82c54343b94ed8c1292e2a
8cfbbe81571f1f7455762ebcfd415d72
ff84447f9ba17e9e9e76da84072940f5
79b69f2847cde61b113650c8e5926be6
0190dcb7c0ed541e6fb6381ff82d9853
7fe80c74b4f812201273f4e5dab1e905
f26e06e727b5ddcbf963ae0e11636c31
5883b450747bf2fe892c2dc0fe51130a
1653a08cc118c53c00b2f62a478252ea
49c2f4b59ffb26d04f604c8ae4c906f5
25dea5af621464b5ea7332b0352fea33
64e98f5690164a0cdfb07894304614d5
ff4d71872ecd3e94b33c41224d95fb13
4eeb1b062357172bcb2a2a2dc4b0f174
7301d241b61b7ed97f839689c3284057
uniq_id
discounted_price
65
60
55
50
45
40
35

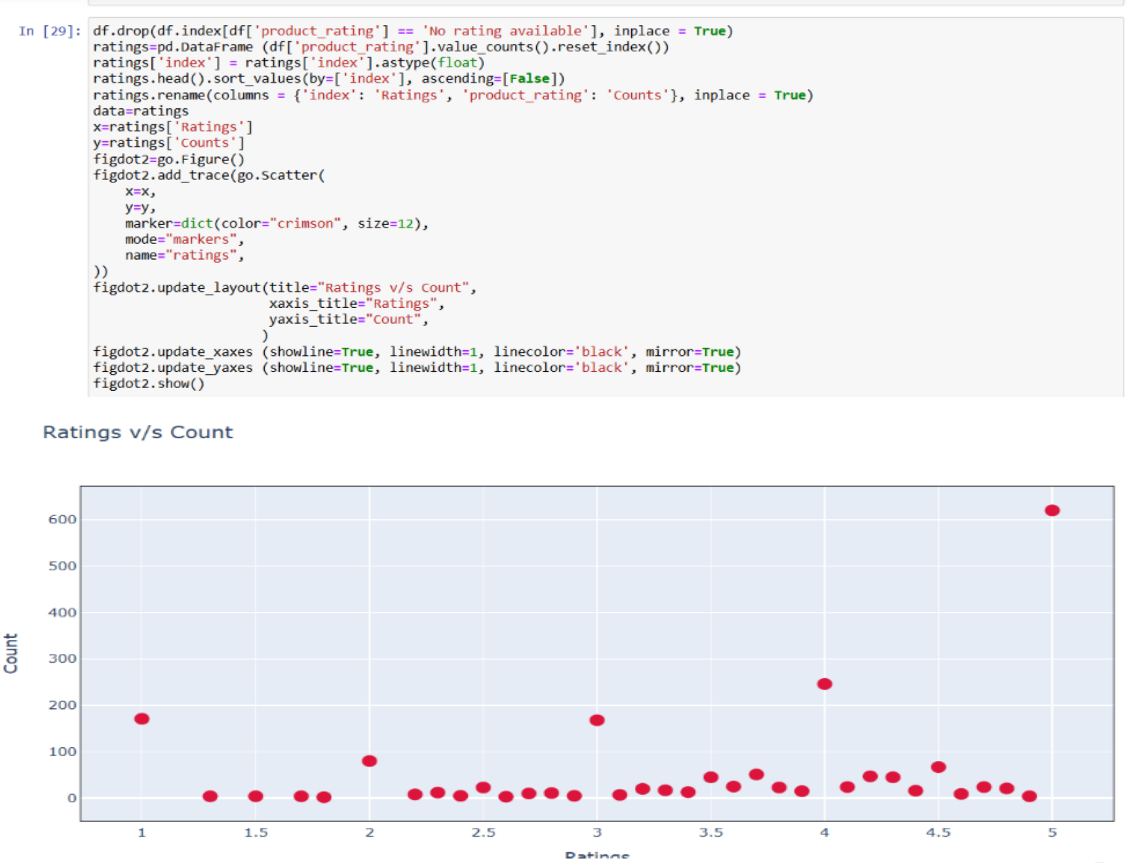


In [22]:
```
total_prod=len(df['pid'])
total_ratings=len(df[df['product_rating']!='No rating available'])
top_ratings=len(df[df['product_rating']=='5'])
df_funnel_1=dict(
    number=[total_prod, total_ratings, top_ratings],
    stage=["Total Products","Products with ratings","Products with 5 star rating"])
funnel_1_fig = px.funnel(df_funnel_1, x='number', y='stage')
funnel_1_fig.show()
```

stage
Total Products
Products with ratings
Products with 5 star rating
20k
1849
620

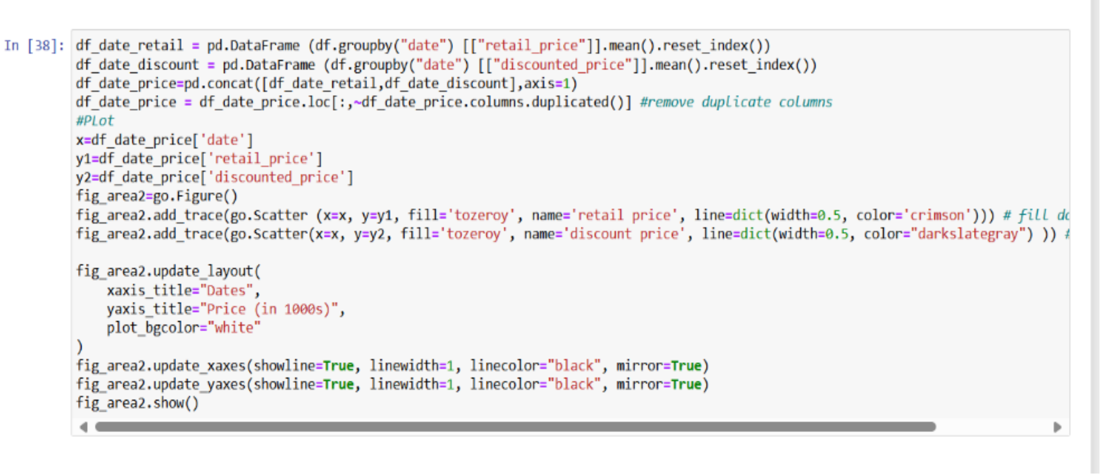


```
In [26]:
rating_5=pd.DataFrame(df.loc[df['product_rating'] == '5'])
top_product_type=rating_5['main_category'].value_counts()
top_brand_type=rating_5['brand'].value_counts()
df_top_product=pd.DataFrame(top_product_type[:5].reset_index())
df_top_product.rename(columns = {'index': 'top_prod'}, inplace = True)
df_top_product.drop('main_category', inplace=True, axis=1)

df_top_brand=pd.DataFrame(top_brand_type[:5].reset_index())
df_top_brand.rename(columns = {'index': 'top_brands'}, inplace = True)
df_top_brand.drop('brand', inplace=True, axis=1)


df_top_brand.head()
df_product_brand_rate5=pd.concat([df_top_product,df_top_brand],axis=1)
```

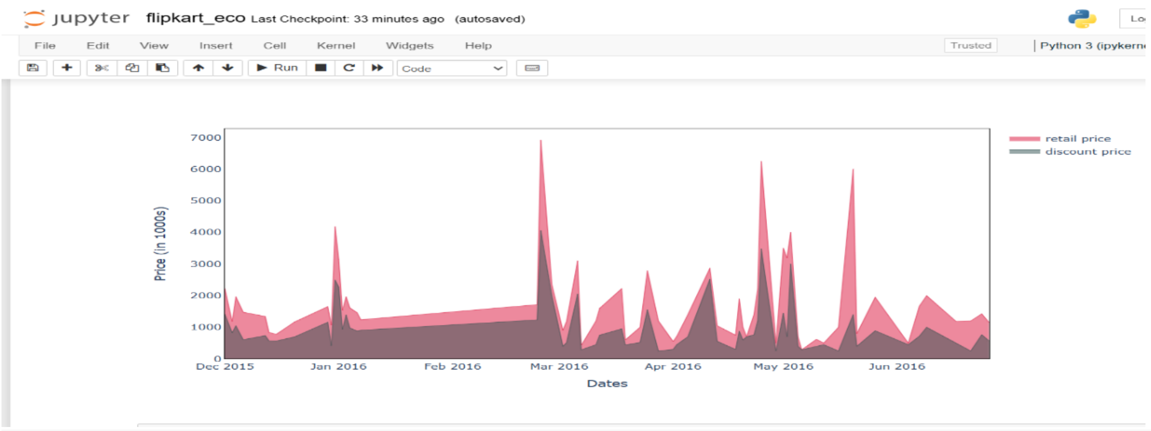


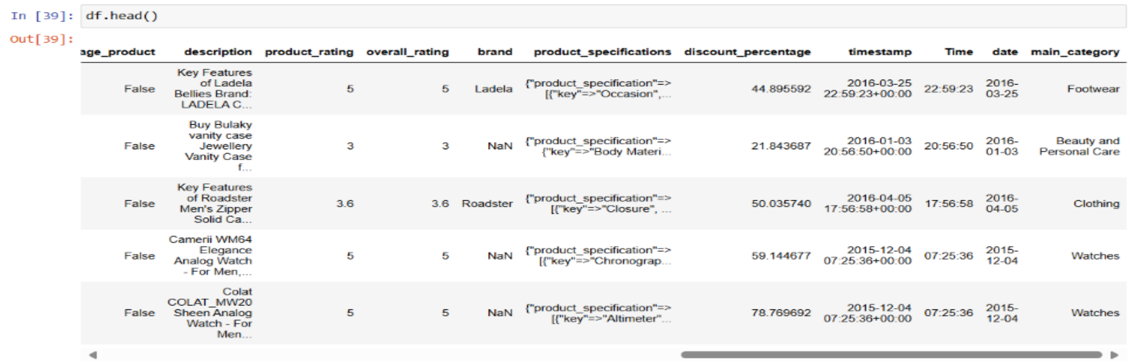
```
In [29]: df.drop(df.index[df['product_rating'] == 'No rating available'], inplace = True)
ratings=pd.DataFrame (df['product_rating'].value_counts().reset_index())
ratings['index'] = ratings['index'].astype(float)
ratings.head().sort_values(by=['index'], ascending=[False])
ratings.rename(columns = {'index': 'Ratings', 'product_rating': 'Counts'}, inplace = True)
data=ratings
x=ratings['Ratings']
y=ratings['Counts']
figdot2=go.Figure()
figdot2.add_trace(go.Scatter(
    x=x,
    y=y,
    marker=dict(color="crimson", size=12),
    mode="markers",
    name="ratings",
))
figdot2.update_layout(title="Ratings v/s Count",
                      xaxis_title="Ratings",
                      yaxis_title="Count",
                     )
figdot2.update_xaxes (showline=True, linewidth=1, linecolor='black', mirror=True)
figdot2.update_yaxes (showline=True, linewidth=1, linecolor='black', mirror=True)
figdot2.show()
```

Ratings v/s Count



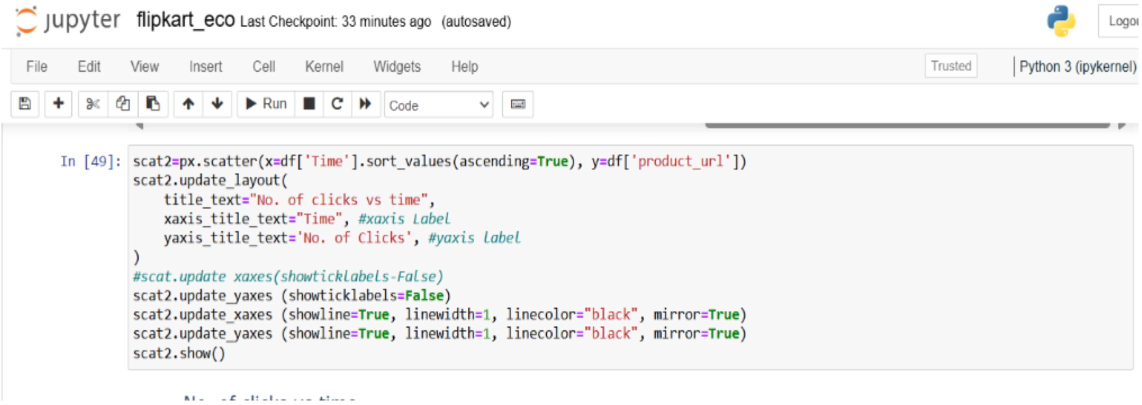
```
In [38]: df_date_retail = pd.DataFrame (df.groupby("date") [["retail_price"]].mean().reset_index())
df_date_discount = pd.DataFrame (df.groupby("date") [["discounted_price"]].mean().reset_index())
df_date_price=pd.concat([df_date_retail,df_date_discount],axis=1)
df_date_price = df_date_price.loc[:,~df_date_price.columns.duplicated()] #remove duplicate columns
#PLot
x=df_date_price['date']
y1=df_date_price['retail_price']
y2=df_date_price['discounted_price']
fig_area2=go.Figure()
fig_area2.add_trace(go.Scatter (x=x, y=y1, fill='tozeroy', name='retail price', line=dict(width=0.5, color='crimson'))) # fill dc
fig_area2.add_trace(go.Scatter(x=x, y=y2, fill='tozeroy', name='discount price', line=dict(width=0.5, color="darkslategray") )) #

fig_area2.update_layout(
    xaxis_title="Dates",
    yaxis_title="Price (in 1000s)",
    plot_bgcolor="white"
)
fig_area2.update_xaxes(showline=True, linewidth=1, linecolor="black", mirror=True)
fig_area2.update_yaxes(showline=True, linewidth=1, linecolor="black", mirror=True)
fig_area2.show()
```



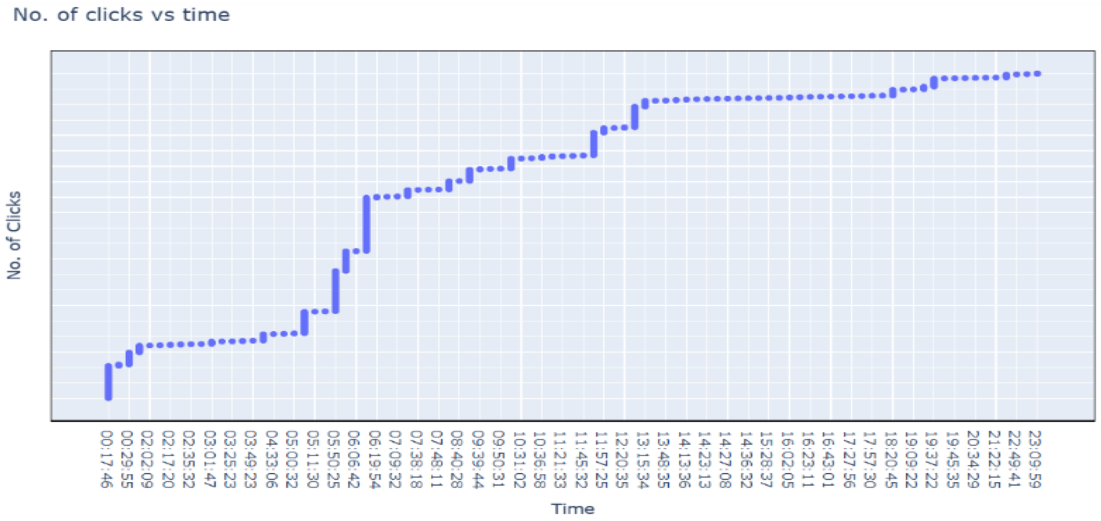


```
In [39]: df.head()
Out[39]:
```

age_product	description	product_rating	overall_rating	brand	product_specifications	discount_percentage	timestamp	Time	date	main_category
False	Key Features of Ladela Bellies Brand: LADELA C...	5	5	Ladela	{"product_specification"=> [{"key"=>"Occasion",...	44.895592	2016-03-25 22:59:23+00:00	22:59:23	2016-03-25	Footwear
False	Buy Bulaky vanity case Jewellery Vanity Case f...	3	3	NaN	{"product_specification"=> {"key"=>"Body Materi...	21.843687	2016-01-03 20:56:50+00:00	20:56:50	2016-01-03	Beauty and Personal Care
False	Key Features of Roadster Men's Zipper Solid Ca...	3.6	3.6	Roadster	{"product_specification"=> [{"key"=>"Closure", ...	50.035740	2016-04-05 17:56:58+00:00	17:56:58	2016-04-05	Clothing
False	Camerii WM64 Elegance Analog Watch - For Men,...	5	5	NaN	{"product_specification"=> [{"key"=>"Chronograp...	59.144677	2015-12-04 07:25:36+00:00	07:25:36	2015-12-04	Watches
False	Colat COLAT_MW20 Sheen Analog Watch - For Men...	5	5	NaN	{"product_specification"=> [{"key"=>"Altimeter"...	78.769692	2015-12-04 07:25:36+00:00	07:25:36	2015-12-04	Watches



```
In [49]: scat2=px.scatter(x=df['Time'].sort_values(ascending=True), y=df['product_url'])
         scat2.update_layout(
             title_text="No. of clicks vs time",
             xaxis_title_text="Time", #xaxis label
             yaxis_title_text='No. of Clicks', #yaxis label
         )
         #scat.update_xaxes(showticklabels=False)
         scat2.update_yaxes (showticklabels=False)
         scat2.update_xaxes (showline=True, linewidth=1, linecolor="black", mirror=True)
         scat2.update_yaxes (showline=True, linewidth=1, linecolor="black", mirror=True)
         scat2.show()
```



Key Takeaways

- **Customer Feedback:** The analysis highlighted common sentiments in Flipkart reviews, providing insights into customer satisfaction and areas needing improvement.
- **Data-Driven Decisions:** The sentiment analysis tool can help Flipkart make data-driven decisions to enhance customer experience, product offerings, and overall service quality.
- **Scalability:** The project demonstrated the scalability of sentiment analysis, showing potential for integration with real-time systems to monitor and respond to customer feedback dynamically.

Future Enhancements

- **Advanced Models:** Implementing deep learning models could further improve accuracy and provide more nuanced sentiment detection.
- **Real-Time Analysis:** Integrating the tool with Flipkart's review system for real-time sentiment analysis could offer immediate insights and prompt actions.
- **Broader Applications:** Expanding the tool to analyze reviews from other eCommerce platforms could provide a comparative analysis of customer satisfaction across different marketplaces.

In conclusion, this sentiment analysis project not only showcases the power of data science in understanding customer sentiment but also sets the stage for continuous improvement and innovation in the eCommerce domain. By harnessing the insights gained, Flipkart can enhance its services, address customer concerns more effectively, and ultimately foster a more positive shopping experience for its users.