Team 8 - CMS Open Payments

Objective:

To analyze the CMS Open Payments data using various tools and build an analytical dashboard as a Proof-of-concept to illustrate the value of data driven marketing analytics . Also, to familiarize the significance of each tool.

What is CMS Open Payments data:

Open Payments is a national disclosure program that promotes a more transparent and accountable health care system by making the financial relationships between applicable manufacturers and group purchasing organizations (GPOs) and health care providers (physicians and teaching hospitals) available to the public.

Datasets:

1. General - This dataset contains General Payments reported for the 2017-2019 program years. General Payments are defined as payments or other transfers of value made to a covered recipient (physician or teaching hospital) that are not made in connection with a research agreement or research protocol.

2. Research - This dataset contains Research Payments reported for the 2017-2019 program years. Research Payments are defined as payments or other transfers of value made in connection with a research agreement or research protocol.

3. Ownership -This dataset contains Ownership and Investment Interest Information reported for the 2017-2019 program years. Ownership and Investment Interest Information is defined as information on the value of ownership or investment interests that a physician or an immediate family member of a physician held in an applicable manufacturer or applicable group purchasing organization (GPO).

Data Wrangling

Trifacta:

With its UI design and the transformations its a very easy tool to clean and, sort and merge dataset.
In our case it had a problem of reading the number of rows properly so we couldnt use the tool for data wrangling.

Pandas:

WIth pandas and numpy libraries it was very easy to manipulate the data.

```
In [7]: total = df.isnull().sum()[df.isnull().sum() != 0].sort_values(ascending = False)
        percent = pd.Series(round(total/len(df)*100,2))
        pd.concat([total, percent], axis=1, keys=['total_missing', 'percent'])
```

Out[7]:

	total_missing	percent
Recipient_Province	149999	100.00
Recipient_Postal_Code	149999	100.00
Physician_License_State_code5	149974	99.98
Associated_Drug_or_Biological_NDC_5	149909	99.94
Physician_License_State_code4	149836	99.89
Associated_Drug_or_Biological_NDC_4	149628	99.75
Teaching_Hospital_ID	149363	99.58
Teaching_Hospital_Name	149363	99.58
Teaching_Hospital_CCN	149363	99.58
Associated_Drug_or_Biological_NDC_3	149210	99.47
Product_Category_or_Therapeutic_Area_5	149137	99.42
Indicate_Drug_or_Biological_or_Device_or_Medical_Supply_5	149137	99.42
Covered_or_Noncovered_Indicator_5	149137	99.42
Name_of_Drug_or_Biological_or_Device_or_Medical_Supply_5	149137	99.42
Physician_License_State_code3	148992	99.33
Associated_Drug_or_Biological_NDC_2	148280	98.85
Name_of_Third_Party_Entity_Receiving_Payment_or_Transfer_of_Value	147862	98.57
Third_Party_Equals_Covered_Recipient_Indicator	146682	97.79
Covered_or_Noncovered_Indicator_4	145828	97.22
Name_of_Drug_or_Biological_or_Device_or_Medical_Supply_4	145828	97.22
Product_Category_or_Therapeutic_Area_4	145828	97.22

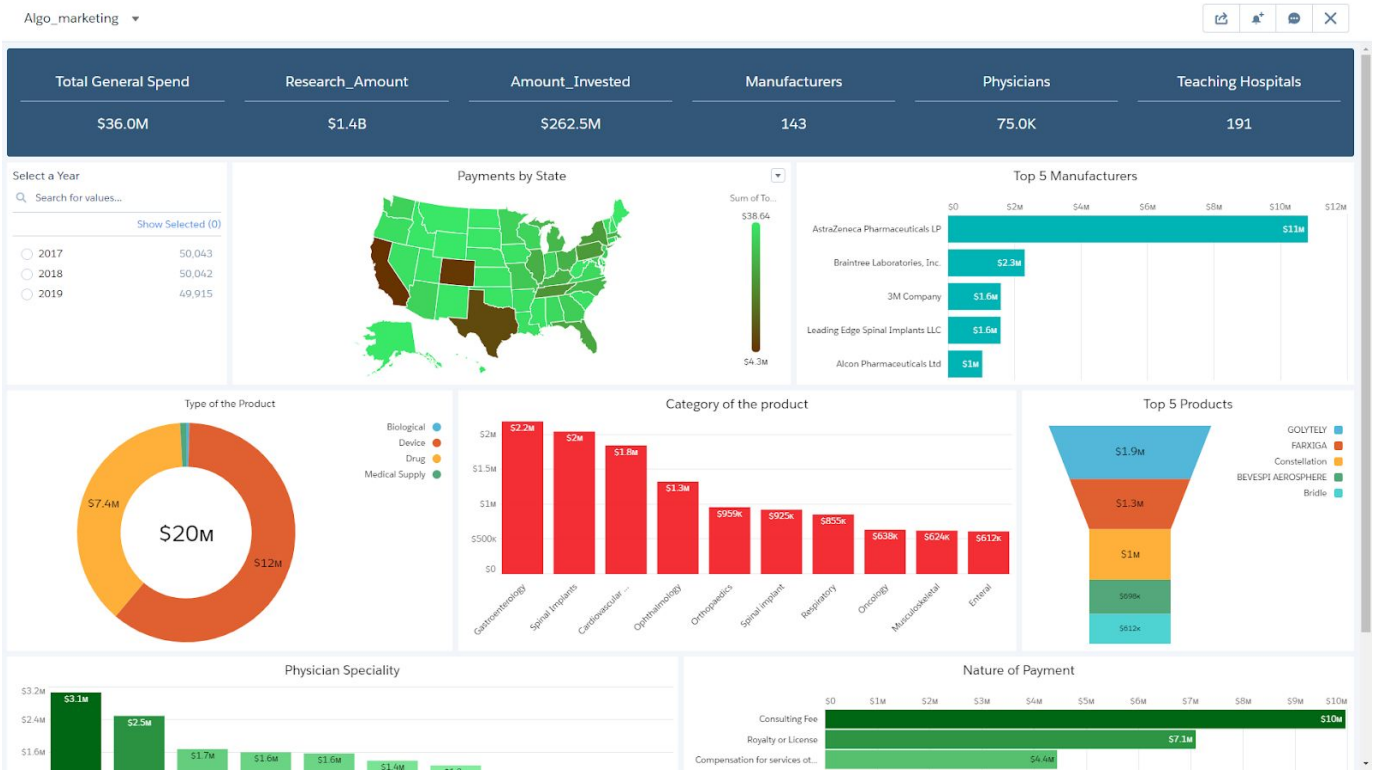
Using formulas we found the percentage of the null values in the dataset. This helped us delete the columns which had null values greater then 80%.

XSV:

XSV is a very fast command line tool with easy syntax which can be used for data wrangling. We used XSV to slice and sample the data. We also used XSV to merge the data files.

Analysis:

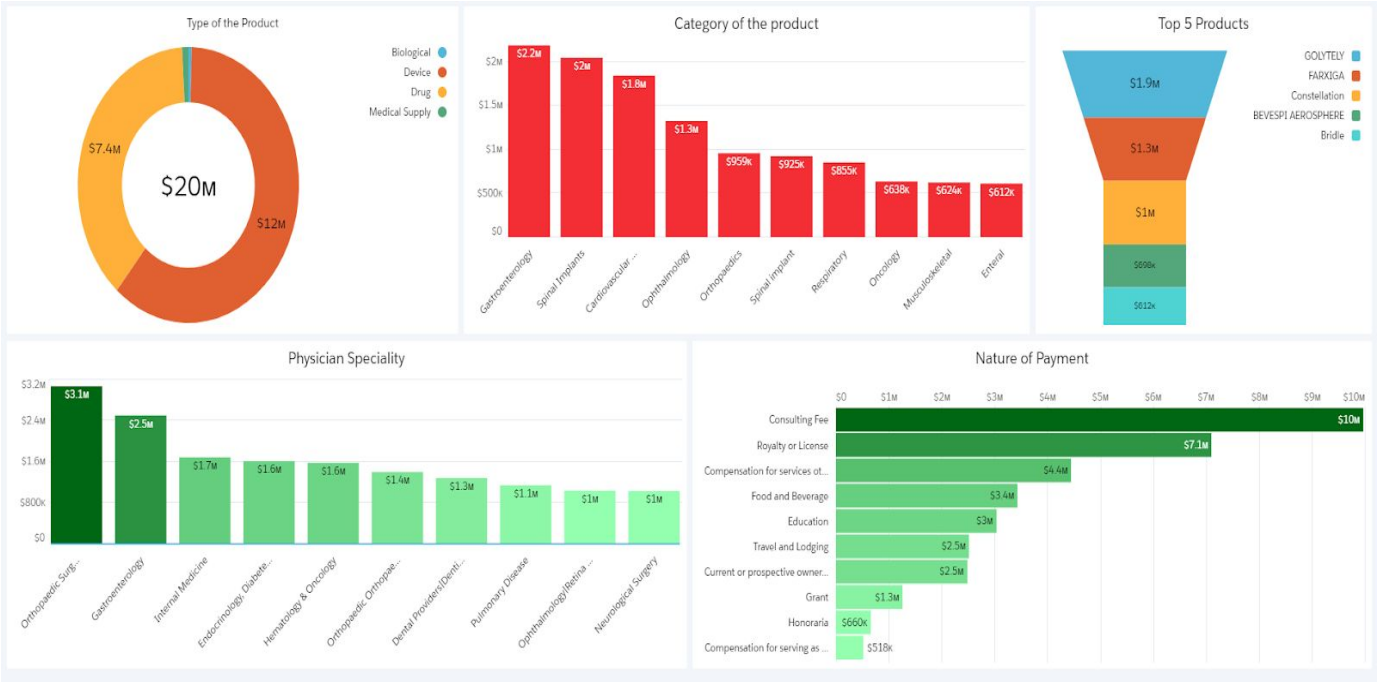
General Dashboard:



This is the main dashboard which contains a significant amount of information. Here we have created yearly statistics about general spend, research amounts and investment amount.

In the main dashboard we can find the manufacturers, their product categories and products as well as physicians and thier specialties. Everything is sorted by the total general spending for testing.

Pricing:

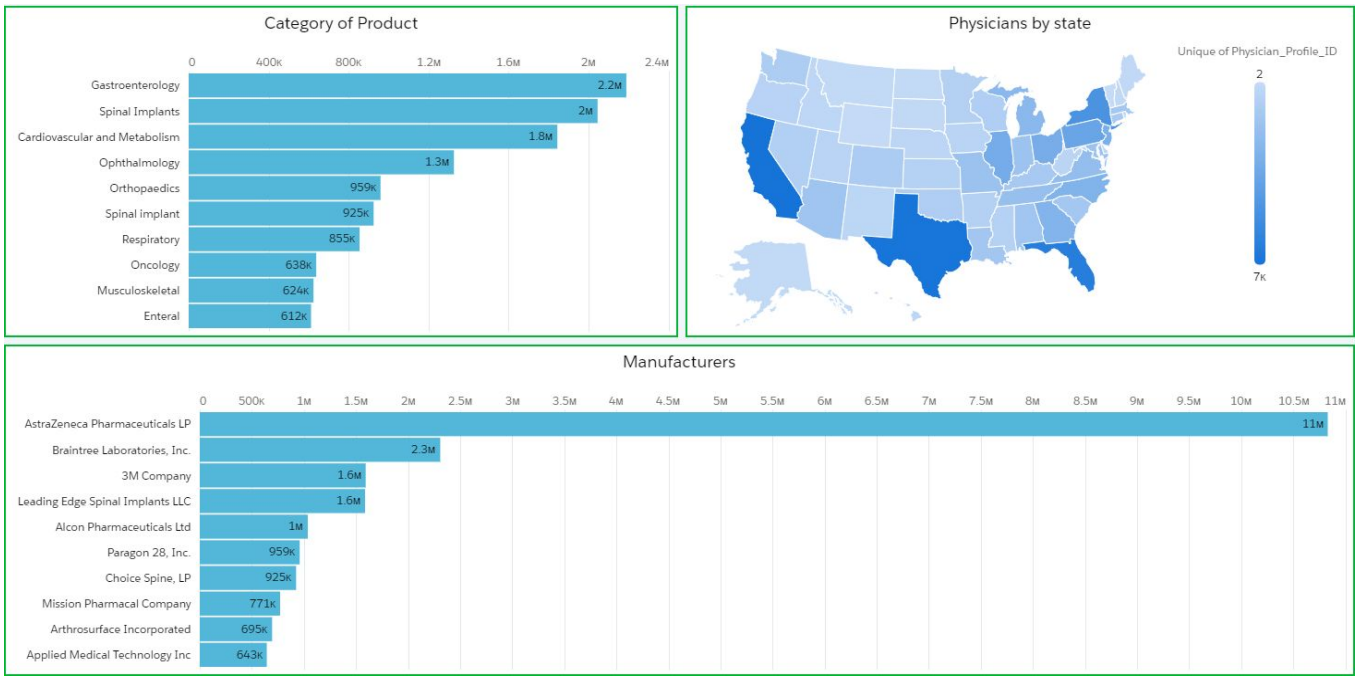


This is a pricing dashboard where we can see the top products for the manufacturers and also the category of the product.
We can also get to know which physician specialties are in demand.

From the dashboard we get to know:
Manufacturers are spending more money for Gastroenterology and implant surgery.
Also orthopedic surgeons and Gastroenterology specialised physicians are more in demand.

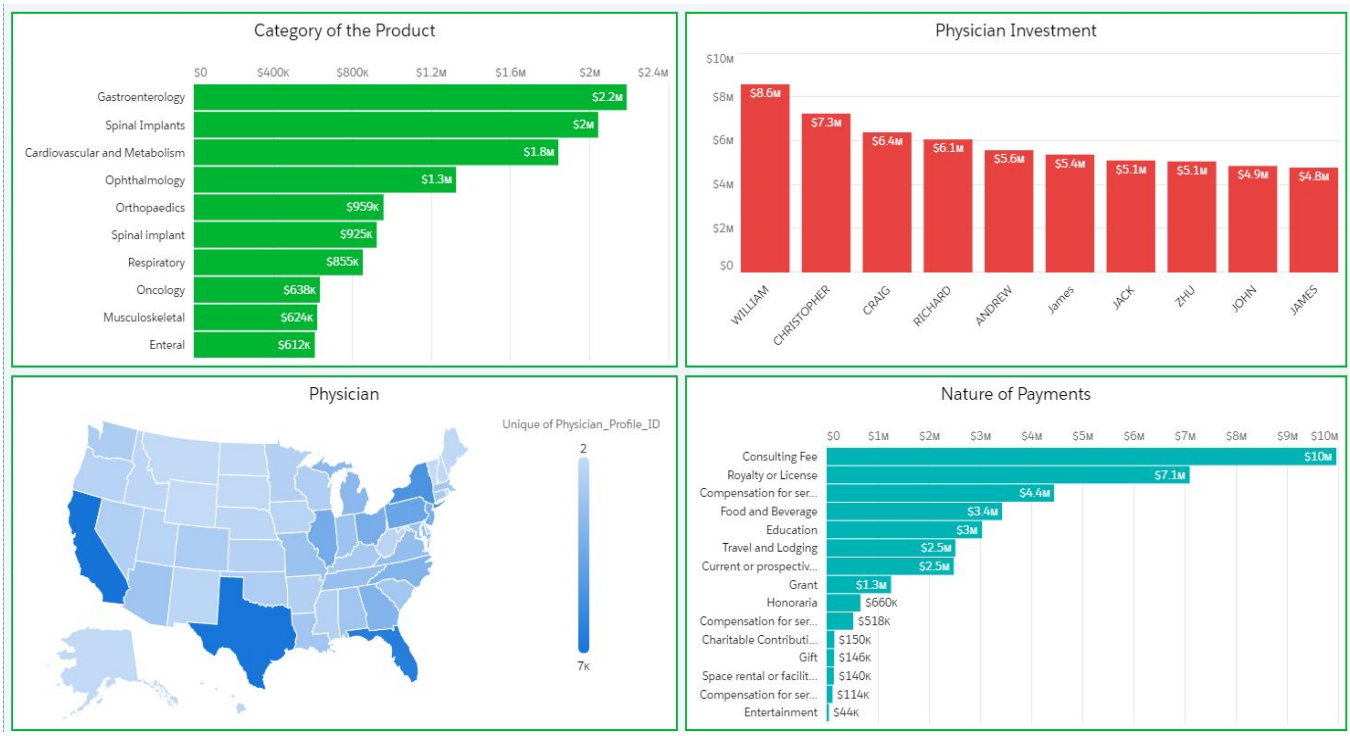
Competitive Analysis - Resource allocation.

In this dashboard a manufacturer can see which other manufacturers in the same product category are targeting which state physicians. So if a manufacturer sees a big concentration in an area the marketing team of the manufacturer can focus on physicians in the different state or area.



Investment analysis.

In the investments analysis manufacturer can get to know which physicians are investing highly in which kind of product. So a new manufacturer can target those physicians to invest in his company.



Conclusion:

For this assignment we used pandas and xsv for data wrangling as trifacta had a problem of row counting.

As the open payments data is accessible to all the manufacturers, they can use this data for competitive intelligence.

Based on the dashboards we can get the data for top product categories and the products. Also competitive intelligence can be used to identify which physicians to target for investments and testing.