GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Data Analysis Using Python

A Project Report of Capstone Project - 2

Submitted by

AASHISH DUBEY

(1613101012 / 16SCSE101832)

In partial fulfilment for the award of the degree Of

Bachelors of Technology

IN

Computer Science and Engineering

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Under the Supervision of

C. RAMESH KUMAR

Assistant Professor

May 2020

GALGOTIAS UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **"DATA ANALYSIS USING PYTHON"** is the bonafide work of **"AASHISH DUBEY(16SCSE101832 / 1613101012)"** who carried out the project work under my supervision.

SIGNATURE OF HEAD

Dr. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor & Dean
School of Computing Science & Engineering

SIGNATURE OF SUPERVISOR

Mr. C. RAMESH KUMAR
Asst. Professor
School of Computing Science & Engineering

TABLE OF CONTENTS

Abstract

A detailed analysis of a particular substance in a food item McDonalds, where we will find out that which food item has maximum sodium content. The data-set which we will use to perform this analysis is released by McDonalds at kaggle.com .The menu items and nutrition facts were scraped from the McDonald's website. The Dataset Nutrition Label1 (the Label) is a diagnostic framework that lowers the barrier to standardized data analysis by providing a distilled yet comprehensive overview of dataset "ingredients" before AI model development. Building a Label that can be applied across domains and data types requires that the framework itself be flexible and adaptable; as such, the Label is comprised of diverse qualitative and quantitative modules generated through multiple statistical and probabilistic modelling back-ends, but displayed in a standardized format.

The benefits of the Label are many fold. For data specialists, the Label will drive more robust data analysis practices, provide an efficient way to select the best dataset for their purposes, and increase the overall quality of AI models as a result of more robust training datasets and the ability to check for issues at the time of model development. For those building and publishing datasets, the Label creates an expectation of explanation, which will drive better data collection practices. We also explore the limitations of the Label, including the challenges of generalizing across diverse datasets, and the risk of using "ground truth" data as a comparison dataset. We discuss ways to move forward given the limitations identified. Lastly, we lay out future directions for the Dataset Nutrition Label project, including research and public policy agendas to further advance consideration of the concept.

Introduction

(i) Overall description

This dataset provides a nutrition analysis of every menu item on the US McDonald's menu, including breakfast, beef burgers, chicken and fish sandwiches, fries, salads, soda, coffee and tea, milkshakes, and desserts. We will perform the detail analysis and try to find out the insights of the calories and other nutritional values.

(ii) Purpose

Our purpose is to find out these information:

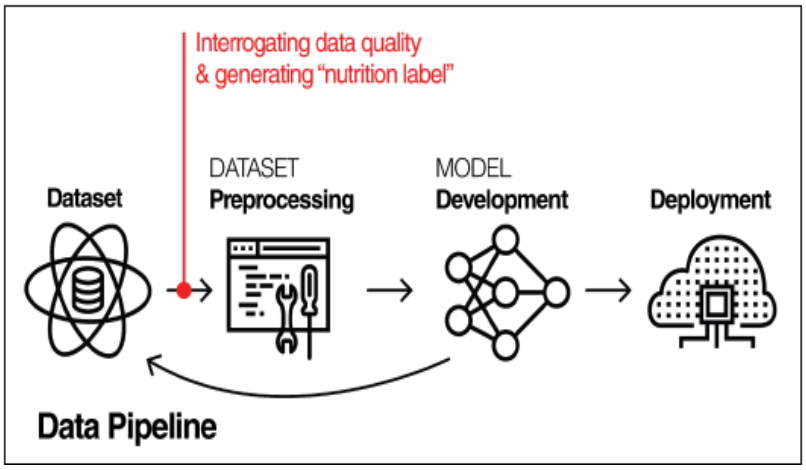
- How many calories does the average McDonald's value meal contain? How much do beverages, like soda or coffee, contribute to the overall caloric intake?
- Does ordered grilled chicken instead of crispy increase a sandwich's nutritional value?
- What about ordering egg whites instead of whole eggs?
- What is the least number of items could you order from the menu to meet one day's nutritional requirements?

(iii) Motivations and scope

Data market is constantly increasing each year. In March 2012, The White House announced a national "Big Data Initiative" that consisted of six Federal departments and agencies committing more than $200 million to big data research projects. Global Pulse which is an innovative lab that is based on the big data mining is also using the Big data to improve the life in developing countries. In today's competitive & complex business world the various aspects of business are intermingled. Change in one aspect has direct or indirect effect on the other aspect. Within an organization, this complexity makes it difficult for business leaders to rely solely on experience (or intuition) to make decisions. This project is another step to deal with this problem and to show the insights of the data.

Proposed model

Data driven decision making systems play an increasingly important and impactful role in our lives.
These frameworks are built on increasingly sophisticated artificial intelligence (AI) systems and are tuned by a growing population of data specialists[3] to infer a vast diversity of outcomes: the song that plays next on your playlist, the type of advertisement you are most likely to see, or whether you qualify for a mortgage and at what rate. These systems deliver untold societal and economic benefits, but they can also pose harm. Researchers continue to uncover troubling consequences of these systems. Data is a fundamental ingredient in AI, and the quality of a dataset used to build a model will directly influence the outcomes it produces. Like the fruit of a poisoned tree, an AI model trained on problematic or missing data will likely produce problematic outcomes. Examples of these problems include gender bias in language translations surfaced through natural language processing, and skin shade bias in facial recognition systems due to non-representative data. Typically the model development pipeline begins with a question or goal. Within the realm of supervised learning, for example, a data specialist will curate a labeled dataset of previous answers in response to the guiding question. Such data is then used to train a model to respond in a way that accurately correlates with past occurrences. In this way, past answers are used to forecast the future. This is particularly problematic when outcomes of past events are contaminated with (often unintentional) bias.



RELATED WORK

More recently, in an effort to improve transparency, accountability, and outcomes of AI systems, AI researchers have proposed methods for standardizing practices and communicating information about the data itself. The first draws from computer hardware and industry safety standards where datasheets are an industry-wide standard. In datasets, however, they are a novel concept. Datasheets are functionally comparable to the label concept and, like labels that by and large objectively surface empirical information, can often include other information such as recommended uses which are more subjective. "Datasheets for Datasets," a proposal from researchers at Microsoft Research, Georgia Tech, University of Maryland, and the AI Now Institute seeks to standardize information about public datasets, commercial APIs, and pre trained models. The proposed datasheet includes dataset provenance, key characteristics, relevant regulations and test results, but also significant yet more subjective information such as potential bias, strengths and weaknesses of the dataset, API, or model, and suggested uses. As domain experts, dataset, API, and model creators would be responsible for creating the datasheets, not end users or other parties. We are also aware of a forthcoming study from the field of natural language processing (NLP), "Data Statements for NLP: Toward Mitigating System Bias and Enabling Better Science". The researchers seek to address ethics, exclusion, and bias issues in NLP systems. Borrowing from similar practices in other fields of practice, the position paper puts forward the concept and practice of "data statements," which are qualitative summaries that provide detailed information and important context about the populations the datasets represent. The information contained in data statements can be used to surface potential mismatches between the populations used to train a system and the populations in planned use prior to deployment, to help diagnose sources of bias that are discovered in deployed systems, and to help understand how experimental results might generalize. The paper's authors suggest that data statements should eventually become required practice for system documentation and academic publications for NLP systems and should be extended to other data types (e.g. image data) albeit with tailored schema.We take a different, yet complementary, approach. We hypothesize that the concept of a "nutrition label" for datasets is an effective means to provide a scalable and efficient tool to improve the process of dataset interrogation and analysis prior to and during model development. In supporting our hypothesis, we created a prototype, the Dataset Nutrition Label (the Label). Three goals drive this work. First, to inform and improve data specialists' selection and interrogation of datasets and to prompt critical analysis. Consequently, data specialists are the primary intended audience. Second, to gain traction as a practical, readily deployable tool, we prioritize efficiency and flexibility. To that end, we do not suggest one specific approach to the Label, or charge one specific community with creating the Label. Rather, our prototype is modular, and the underlying framework is one that anyone can use. Lastly, we leverage probabilistic computing tools to surface potential corollaries, anomalies, and proxies. This is particularly beneficial because resolving these issues requires excess development time, and can lead to undesired correlations in trained models.

Code/ Chapters

Introductory Chapter: Loading the packages and libraries

```
 import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import warnings
warnings.filterwarnings('ignore')
```

Like every standard data exploration, let us load the data via the Pandas package

```
menu = pd.read_csv('menu.csv')
menu.head(2)
```

Chapters Developing the main theme of the project work

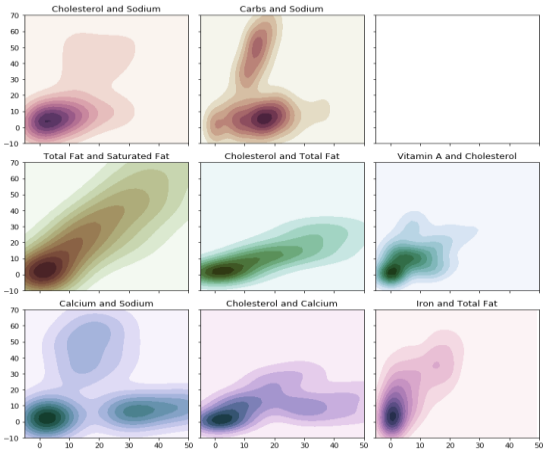
Quick checks on Data quality Always imperative to check first on the quality of the data - i.e whether there are any nulls or blanks in the columns/features, the row and column wise sizes as well as whether any of the numbers don't make sense (like having any infinities in the values. We can accomplish all these as such

```
# Check for Nulls
print(menu.isnull().any())
print("--------------------------------------")
# check for numbers
print(menu.describe())
print("--------------------------------------")
Category False
Item False
```

It seems that our brief data quality checks have now all passed. So onto the data itself, we can see that a rich source of nutritional information is provided to us where we Comparisons of Features via Contour and Correlation plots First up on our analysis, let us take a look at how one feature feeds into the other. In particular I will plot a Contour or Kernel Density Estimation (KDE) plots which will provide the distribution of one feature to another. This is to simply get a quick feel for the data that we have in a quantitative manner as well as to introduce the plotting capabilities of the brilliant Seaborn library. Contour plots To generate a Contour plot, it is a very simple Seaborn invocation of "kdeplot( )" as follows :

```
# Plotting the KDEplots
f, axes = plt.subplots(3, 3, figsize=(10, 10), sharex=True, sharey=True)
s = np.linspace(0, 3, 10)
cmap = sns.cubehelix_palette(start=0.0, light=1, as_cmap=True)
x = menu['Cholesterol (% Daily Value)'].values
y = menu['Sodium (% Daily Value)'].values
sns.kdeplot(x, y, cmap=cmap, shade=True, cut=5, ax=axes[0,0])
axes[0,0].set(xlim=(-10, 50), ylim=(-30, 70), title = 'Cholesterol and Sodium')
cmap = sns.cubehelix_palette(start=0.333333333333, light=1, as_cmap=True)
x = menu['Carbohydrates (% Daily Value)'].values
y = menu['Sodium (% Daily Value)'].values
sns.kdeplot(x, y, cmap=cmap, shade=True, ax=axes[0,1])
axes[0,1].set(xlim=(-5, 50), ylim=(-10, 70), title = 'Carbs and Sodium')
cmap = sns.cubehelix_palette(start=0.666666666667, light=1, as_cmap=True)
x = menu['Carbohydrates (% Daily Value)'].values
```

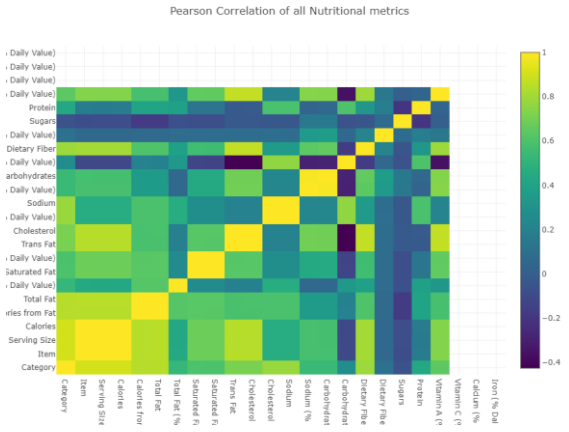
```
y = menu['Cholesterol (% Daily Value)'].values
sns.kdeplot(x, y, cmap=cmap, shade=True, ax=axes[0,2])
axes[0,2].set(xlim=(-5, 50), ylim=(-10, 70), title = 'Carbs and Cholesterol')
cmap = sns.cubehelix_palette(start=1.0, light=1, as_cmap=True)
x = menu['Total Fat (% Daily Value)'].values
y = menu['Saturated Fat (% Daily Value)'].values
sns.kdeplot(x, y, cmap=cmap, shade=True, ax=axes[1,0])
axes[1,0].set(xlim=(-5, 50), ylim=(-10, 70), title = 'Total Fat and Saturated Fat')
cmap = sns.cubehelix_palette(start=1.333333333333, light=1, as_cmap=True)
x = menu['Total Fat (% Daily Value)'].values
y = menu['Cholesterol (% Daily Value)'].values
sns.kdeplot(x, y, cmap=cmap, shade=True, ax=axes[1,1])
axes[1,1].set(xlim=(-5, 50), ylim=(-10, 70), title = 'Cholesterol and Total Fat')
cmap = sns.cubehelix_palette(start=1.666666666667, light=1, as_cmap=True)
x = menu['Vitamin A (% Daily Value)'].values
y = menu['Cholesterol (% Daily Value)'].values
sns.kdeplot(x, y, cmap=cmap, shade=True, ax=axes[1,2])
axes[1,2].set(xlim=(-5, 50), ylim=(-10, 70), title = 'Vitamin A and Cholesterol')
cmap = sns.cubehelix_palette(start=2.0, light=1, as_cmap=True)
x = menu['Calcium (% Daily Value)'].values
y = menu['Sodium (% Daily Value)'].values
sns.kdeplot(x, y, cmap=cmap, shade=True, ax=axes[2,0])
axes[2,0].set(xlim=(-5, 50), ylim=(-10, 70), title = 'Calcium and Sodium')
cmap = sns.cubehelix_palette(start=2.333333333333, light=1, as_cmap=True)
x = menu['Calcium (% Daily Value)'].values
y = menu['Cholesterol (% Daily Value)'].values
sns.kdeplot(x, y, cmap=cmap, shade=True, ax=axes[2,1])
axes[2,1].set(xlim=(-5, 50), ylim=(-10, 70), title = 'Cholesterol and Calcium')
cmap = sns.cubehelix_palette(start=2.666666666667, light=1, as_cmap=True
x = menu['Iron (% Daily Value)'].values
y = menu['Total Fat (% Daily Value)'].values
sns.kdeplot(x, y, cmap=cmap, shade=True, ax=axes[2,2])
axes[2,2].set(xlim=(-5, 50), ylim=(-10, 70), title = 'Iron and Total Fat')
f.tight_layout()
```



Pearson Correlation by Plotly package by plotting a Heatmap of the correlation between features as follows :

```
data = [
go.Heatmap(
z= menu.ix[:,3:].corr().values,
x=menu.columns.values,
y=menu.columns.values,
colorscale='Viridis',
```

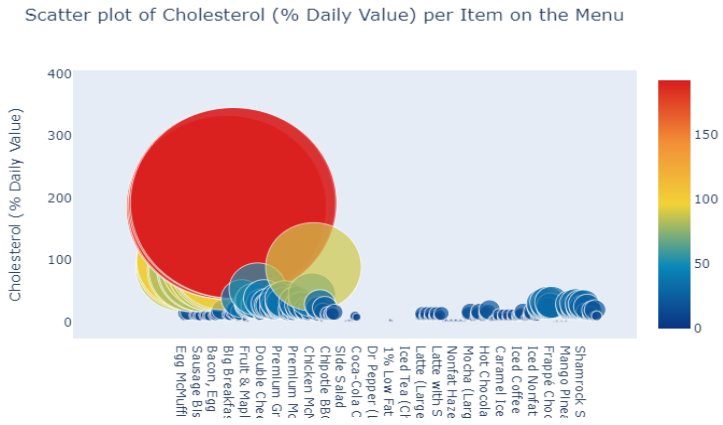
```
text = True ,
opacity = 1.0
) ]
layout = go.Layout(
title='Pearson Correlation of all Nutritional metrics',
xaxis = dict(ticks='', nticks=36),
yaxis = dict(ticks='' ),
width = 900, height = 700,
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig, filename='labelled-heatmap')
```



As evinced from the correlation plots, one can already see features that obviously tie into one another (the more yellow sections of the plot). For example serving size and calories.
However there are some surprising correlations which are pretty unintuitive. For example there are quite weak correlations between Total Fats and Saturated/Trans Fats although from a non-health expert's outset (like me), I would have thought it logical for one to contribute to the other. The heatmap also throws up interesting findings from the blotches of negative correlated plots (dark blue/black). For example it shows that Carbohydrates in general are quite negatively correlated to Trans Fat, Cholesterol, Sodium, Dietary Fiber and Vitamin A. That is a really whooping number of negative correlations from Carbs. Is there any possible issue with the data quality? Now since it is evident that the Carbohydrate column is quite negatively correlated with the other columns, I have just the slightest question in my mind with regards to the quality of the data for that metric. However, I could also very well be wrong in the sense that carbohydrate laden foods could tend not to have much else in them (apart from carbs that is) - i.e no Vitamins, salt, cholesterol and hence accounting for the negative correlations. 2. Analysing Nutritional Content per Item Having had a high-level overview of the different features/columns that are contained within our dataset, let us proceed to a more granular level of analysis. For any who follow my kernels, now turn to the Plotly interactive visualisation package. Scatter Plot of Cholesterol (% Daily Value) per MacDonald's Item

```
trace = go.Scatter(
y = menu['Cholesterol (% Daily Value)'].values,
x = menu['Item'].values,
mode='markers',
marker=dict(
size= menu['Cholesterol (% Daily Value)'].values,
color = menu['Cholesterol (% Daily Value)'].values,
colorscale='Portland',
showscale=True
),
text = menu['Item'].values
)
data = [trace]
layout= go.Layout(
```

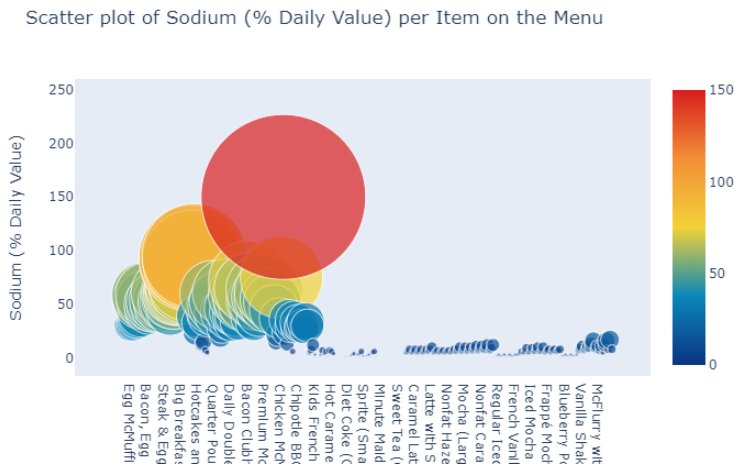
```
autosize= True,
title= 'Scatter plot of Cholesterol (% Daily Value) per Item on the Menu',
hovermode= 'closest',
xaxis=dict(
showgrid=False,
zeroline=False,
showline=False
),
yaxis=dict(
title= 'Cholesterol (% Daily Value)',
ticklen= 5,
gridwidth= 2,
showgrid=False,
zeroline=False,
showline=False
),
showlegend= False
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig,filename='scatterChol')
```



Takeaway from the plot The most striking visuals from the scatter plot are the few large red plots close to the top. These circular plots were scaled such as that the higher the Cholesterol (% Daily value), the larger the plot thereby making for intuitive visuals. As we can see, the main culprit (red circle) is the MacDonald's Big Breakfast range, accounting for a whooping 185% of Cholesterol (% Daily value). Further down from larger red plots, there are a greater number of yellow circular plots which can be attributed to items such as the Egg/Sausage McMuffin range contributing to nearly a day's worth of Cholesterol. Most Cholesterol-laden item : Big Breakfast (Large Biscuit) Scatter Plot of Sodium (% Daily Value) per MacDonald's Item

```
trace = go.Scatter(
y = menu['Sodium (% Daily Value)'].values,
x = menu['Item'].values,
mode='markers',
marker=dict(
size= menu['Sodium (% Daily Value)'].values,
color = menu['Sodium (% Daily Value)'].values,
colorscale='Portland',
showscale=True
),
text = menu['Item'].values
)
data = [trace]
layout= go.Layout(
autosize= True,
title= 'Scatter plot of Sodium (% Daily Value) per Item on the Menu',
hovermode= 'closest',
```

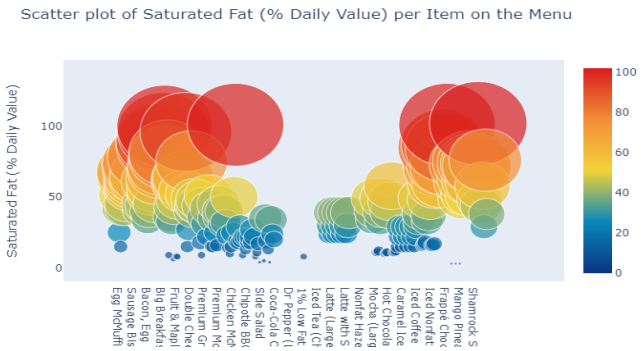
```
xaxis=dict(
showgrid=False,
zeroline=False,
showline=False
),
yaxis=dict(
title= 'Sodium (% Daily Value)',
ticklen= 5,
gridwidth= 2,
showgrid=False,
zeroline=False,
showline=False,
),
showlegend= False
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig,filename='scatterChol')
```



Takeaway from the plot The scatter plots for Sodium (% Daily Value) seem to follow a similar distribution of points whereby MacDonald food items contributing the greatest amount of sodium are scaled largest. As evinced by the largest red circular plot, the 40-piece Chicken McNuggets are the greatest contributor to Sodium intake. The Big Breakfast range with Hotcakes follow up as a close second as a contributor to the sodium amount. Greatest amount of Sodium : Chicken McNuggets (40 piece) Scatter Plot of Saturated Fat (% Daily Value) per MacDonald's Item

```
trace = go.Scatter(
y = menu['Saturated Fat (% Daily Value)'].values,
x = menu['Item'].values,
mode='markers',
marker=dict(
size= menu['Saturated Fat (% Daily Value)'].values,
color = menu['Saturated Fat (% Daily Value)'].values,
colorscale='Portland',
showscale=True
),
text = menu['Item'].values
)
data = [trace]
layout= go.Layout(
autosize= True,
title= 'Scatter plot of Saturated Fat (% Daily Value) per Item on the Menu',
hovermode= 'closest',
xaxis=dict(
showgrid=False,
zeroline=False,
showline=False
```

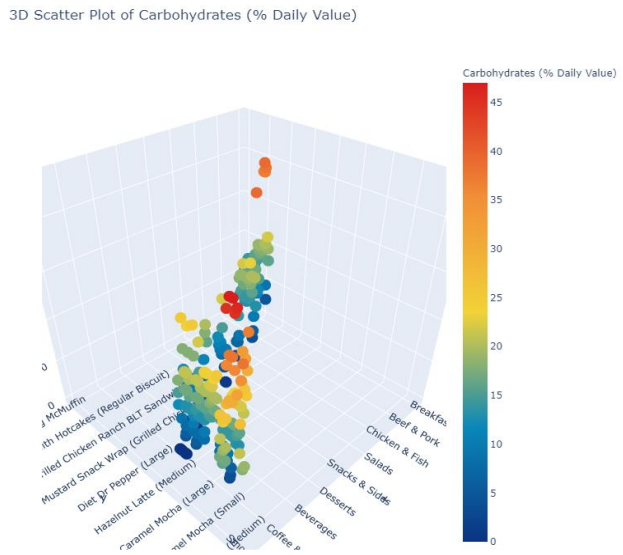
```
),
yaxis=dict(
title= 'Saturated Fat (% Daily Value)',
ticklen= 5,
gridwidth= 2,
showgrid=False,
zeroline=False,
showline=False,
),
showlegend= False
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig,filename='scatterChol')
```

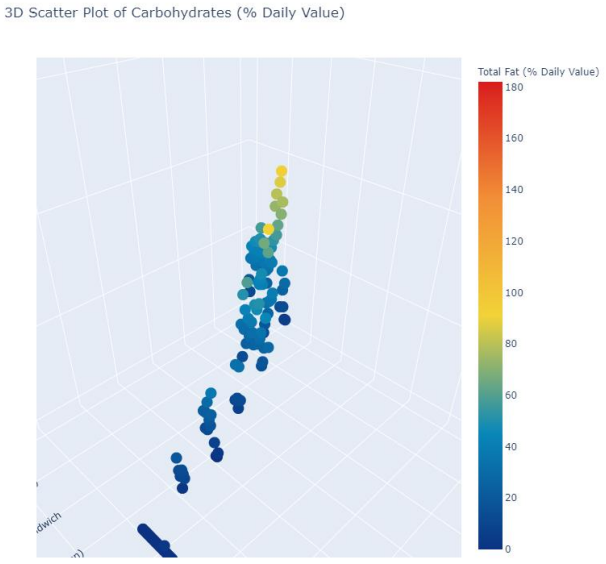


These scatter plots now show a much larger distribution of red plots. This shows that there are quite a handful of MacDonald's food items which contain a dangerous amount of Saturated Fat, where one single food item can contain an amount close to the one's recommended daily allowance. For example the McFlurry with M&M candies or even the Frappe with Chocolate Chips. Greatest amount of Saturate Fats: McFlurry with M&M candies, Chicken McNuggets (40 piece), Frappe Chocolate Chip, Big Breakfast with Hotcakes etc 3D Scatter plots of Total Fat and Carbohydrate levels Let us play around with Plotly's capabilities and mix up our scatter plots a bit. Before we were plotting 2D scatter plots showing the distribution of various nutritional contents against the various food items. Let us now add in a 3rd dimension to the mix (3D Scatter plots) and observe what the distribution might look like by adding in the Category and plotting the scatter plots for carbohydrates and Total Fat content.
3D scatter plot for Total Fats

```
trace1 = go.Scatter3d(
x=menu['Category'].values,
y=menu['Item'].values,
z=menu['Total Fat (% Daily Value)'].values,
text=menu['Item'].values,
mode='markers',
marker=dict(
sizemode='diameter',
#sizeref=750,
#size= dailyValue['Cholesterol (% Daily Value)'].values,
color = menu['Total Fat (% Daily Value)'].values,
colorscale = 'Portland',
colorbar = dict(title = 'Total Fat (% Daily Value)'),
line=dict(color='rgb(255, 255, 255)')
)
)
data=[trace1]
layout=dict(height=800, width=800, title='3D Scatter Plot of Carbohydrates (% Daily Value)')
fig=dict(data=data, layout=layout)
py.iplot(fig, filename='3DBubble')
# 3D scatter plot for Carbohydrate
trace1 = go.Scatter3d(
```

```
x=menu['Category'].values,
y=menu['Item'].values,
z=menu['Carbohydrates (% Daily Value)'].values,
text=menu['Item'].values,
mode='markers',
marker=dict(
sizemode='diameter',
sizeref=750,
size= dailyValue['Cholesterol (% Daily Value)'].values,
color = menu['Carbohydrates (% Daily Value)'].values,
colorscale = 'Portland',
colorbar = dict(title = 'Carbohydrates (% Daily Value)'),
line=dict(color='rgb(255, 255, 255)')
)
)
data=[trace1]
layout=dict(height=800, width=800, title='3D Scatter Plot of Carbohydrates (% Daily Value)')
fig=dict(data=data, layout=layout)
py.iplot(fig, filename='3DBubble')
```



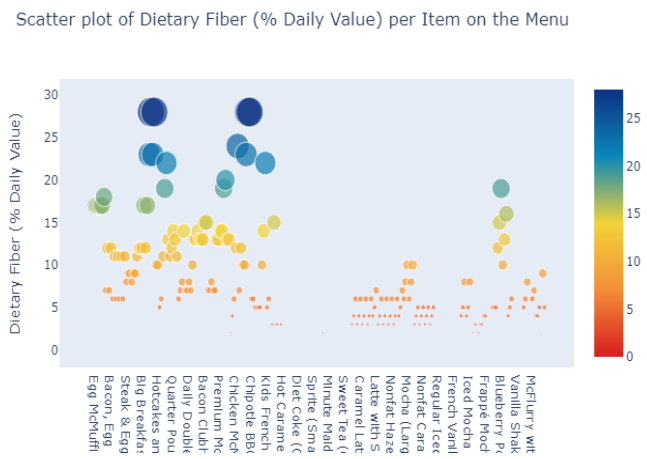


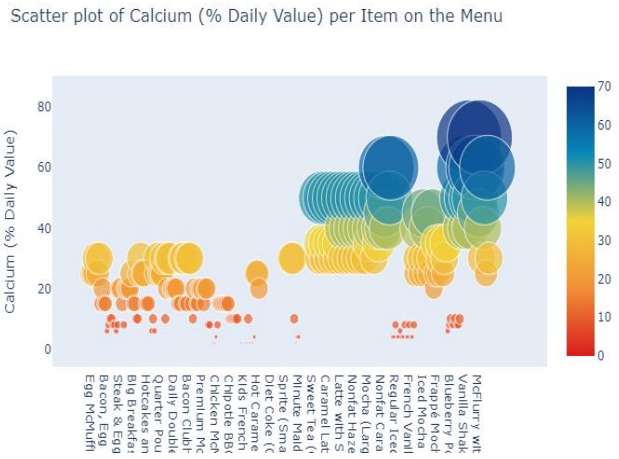
Anyway having looked at nutritional metrics which carry largely negative connotations in society (Cholesterol, Sodium, Total Fat), let us balance this out and generate the scatter plots of nutritional metrics that carry more positive social weights . Namely Calcium, Iron, Dietary Fibre to name a few. Scatter Plots of Dietary Fiber, Calcium and Iron (% Daily Value) per MacDonald's Item

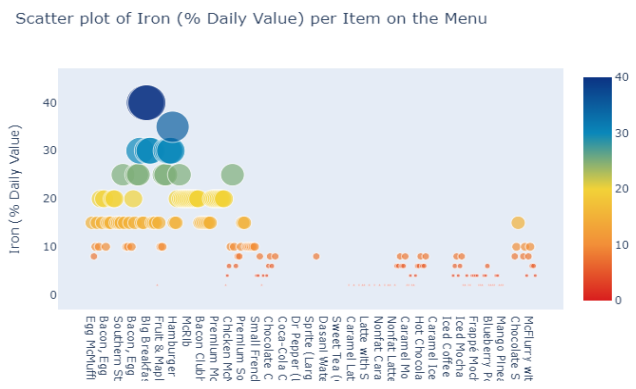
```
trace = go.Scatter(
y = menu['Dietary Fiber (% Daily Value)'].values,
x = menu['Item'].values,
mode='markers',
marker=dict(
size= menu['Dietary Fiber (% Daily Value)'].values,
#color = np.random.randn(500), #set color equal to a variable
color = menu['Dietary Fiber (% Daily Value)'].values,
colorscale='Portland',
reversescale = True,
showscale=True
),
```

```
text = menu['Item'].values
)
data = [trace]
layout= go.Layout(
autosize= True,
title= 'Scatter plot of Dietary Fiber (% Daily Value) per Item on the Menu',
hovermode= 'closest',
xaxis=dict(
showgrid=False,
zeroline=False,
showline=False
),
yaxis=dict(
title= 'Dietary Fiber (% Daily Value)',
ticklen= 5,
gridwidth= 2,
showgrid=False,
zeroline=False,
showline=False,
),
showlegend= False
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig,filename='scatterChol')
# Calcium Scatter plots
trace = go.Scatter(
y = menu['Calcium (% Daily Value)'].values,
x = menu['Item'].values,
mode='markers',
marker=dict(
size= menu['Calcium (% Daily Value)'].values,
#color = np.random.randn(500), #set color equal to a variable
color = menu['Calcium (% Daily Value)'].values,
colorscale='Portland',
reversescale = True,
showscale=True
),
text = menu['Item'].values
)
data = [trace]
layout= go.Layout(
autosize= True,
title= 'Scatter plot of Calcium (% Daily Value) per Item on the Menu',
hovermode= 'closest',
xaxis=dict(
showgrid=False,
zeroline=False,
showline=False
),
yaxis=dict(
title= 'Calcium (% Daily Value)',
ticklen= 5,
gridwidth= 2,
showgrid=False,
zeroline=False,
showline=False,
),
showlegend= False
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig,filename='scatterChol')
# Iron Scatter plots
trace = go.Scatter(
y = menu['Iron (% Daily Value)'].values,
x = menu['Item'].values,
```

```
mode='markers',
marker=dict(
size= menu['Iron (% Daily Value)'].values,
color = np.random.randn(500), #set color equal to a variable
color = menu['Iron (% Daily Value)'].values,
colorscale='Portland',
reversescale = True,
showscale=True
),
text = menu['Item'].values
)
data = [trace]
layout= go.Layout(
autosize= True,
title= 'Scatter plot of Iron (% Daily Value) per Item on the Menu',
hovermode= 'closest',
xaxis=dict(
showgrid=False,
zeroline=False,
showline=False
),
yaxis=dict(
title= 'Iron (% Daily Value)',
ticklen= 5,
gridwidth= 2,
showgrid=False,
zeroline=False,
showline=False,
),
showlegend= False
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig,filename='scatterChol')
```



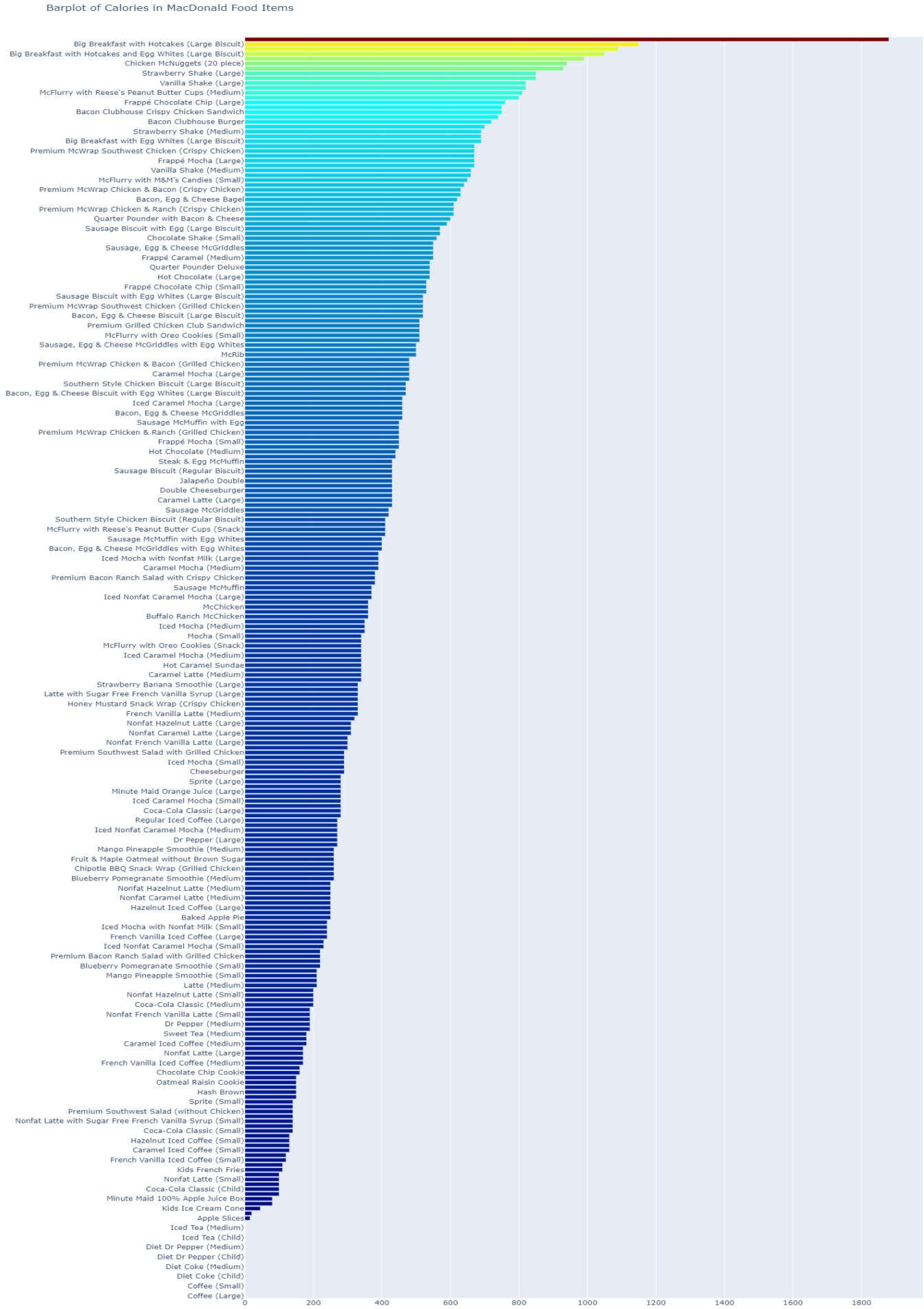




Interactive Barplots of Calorie content per food item. Finally to close out on analysing the nutritional content per item on the menu, let us generate some barplots. Since calories seems to be quite a sticking point with health guidelines and people in general, it may be informative to plot calorific plots for each food item in this dataset just to give readers a general overview of what each item on the MacDonald's menu entails. In this plot, Plotly's statistical plots which give it a Seaborn-type feel to the plots while keeping the interactivity. First we identify the column containing the calorie values and sort them as follows:

```
x, y = (list(x) for x in zip(*sorted(zip(menu.Calories.values, menu.Item.values),
reverse = False))) Then we invoke the Bar plotting functionality within Plotly as such
trace2 = go.Bar(
x=x ,
y=y,
marker=dict(
color=x,
colorscale = 'Jet',
reversescale = False
),
name='Household savings, percentage of household disposable income',
orientation='h',
)
layout = dict(
title='Barplot of Calories in MacDonald Food Items',
width = 1500, height = 2600,
yaxis=dict(
showgrid=False,
showline=False,
showticklabels=True,
# domain=[0, 0.85],
))
fig1 = go.Figure(data=[trace2])
fig1['layout'].update(layout)
py.iplot(fig1, filename='plots')
```

Final Output



Future Enhancements

Large Scale Enterprises are rapidly adopting machine learning for driving their business in several ways. Automation of several tasks is one of the key **future** goals of the industries. As a result, they are able to prevent losses from taking place. With the rise of artificial intelligence (AI) and machine learning (ML), organizations are demanding faster insights to remain competitive. Remarkably, the same technology **advancements** that drive this urgency are also the key to unlocking better efficiency in **data science** work. Most technology research firms are tracking the self-serve data analytics trends. These trends are making it possible for the average enterprise and the average business user to leverage sophisticated analytics, algorithms and techniques without the skills of a data scientist. The solutions are easy to use and allow the average user to build on their core business skills and see data and make decisions in a meaningful way. Think of it as data democratization. These new solutions for advanced analytics and augmented analytics, self-serve data preparation, smart visualization and assisted predictive modeling will allow data scientists to focus on strategic initiatives and business users to leverage sophisticated tools without a lot of training so the enterprise will get rapid ROI and low TCO. Does Data Democratization Result in Data Anarchy and Bad Business Decisions? This article will helps us to understand how data literacy can help the enterprise and the business users and how these advancements in data analytics will bring data democracy to the organization.

References

- International Journal of Scientific & Engineering Research, Volume 4, Issue 12, December-2013 2172 ISSN 2229-5518 IJSER © 2013 http://www.ijser.org
- Python.org
- Google.com
- https://www.kaggle.com/mcdonalds/nutrition-facts
- The Dataset Nutrition Label: A Framework To Drive Higher Data Quality Standards Sarah Holland , Ahmed Hosny, Sarah Newman , Joshua Joseph, and Kasia Chmielinski Assembly, MIT Media Lab and Berkman Klein Center at Harvard University, Dana-Farber Cancer Institute, Harvard Medical School, metaLAB (at) Harvard, Berkman Klein Center for Internet & Society, Harvard University
- Quora.com