"Treat Yo' Selves"

A Machine Learning Story

Inspired by a true story

Group Members

Michael Damas
Ami Patel
Bob Pickerel
Gina Schulte
Austin Skorb
Connor Starrett
Chloe Stitik
Preethi Reddy Vontela

Our Question

01

Can a user's profile be used to predict the rating a customer will give a restaurant?

Step 01.

Importing the Data

We used PostgreSQL to import the five CSV files found on Kaggle. We then used a series of inner and left joins to make one CSV.

Datasets Used in Analysis:

- Kaggle Dataset - User Cuisine
- Kaggle Dataset - User Profile
- Kaggle Dataset - Chef Moz Cuisine
- Kaggle Dataset - Final Ratings
- Geo Locations

The Great Merge

1st Table:

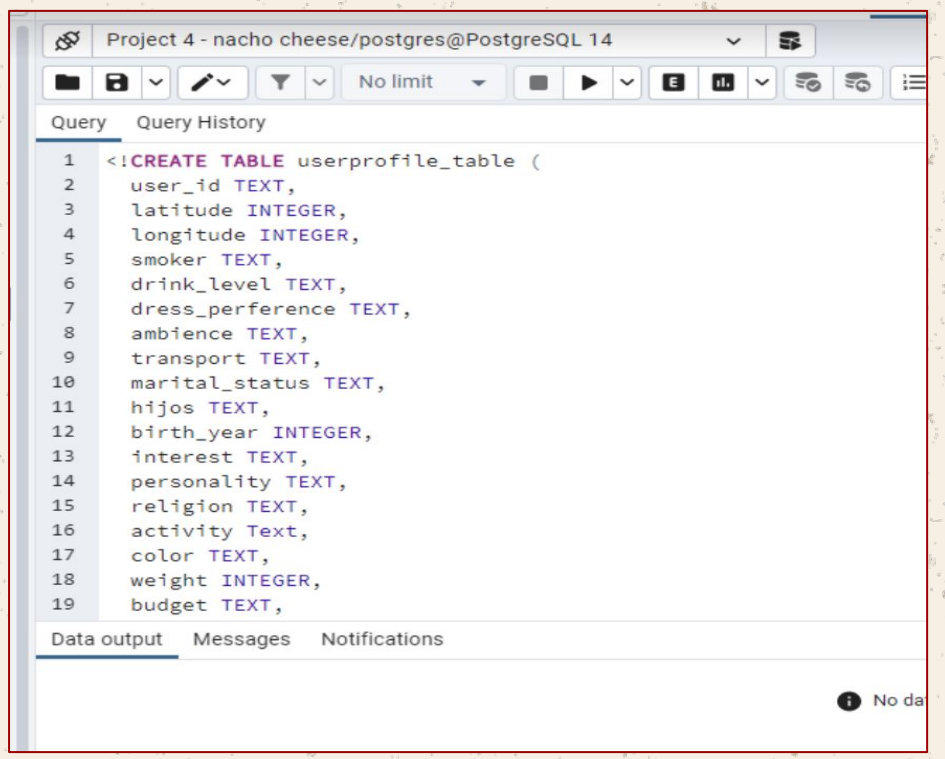
In order to create our first table we merged the datasets named userprofile and rating profile to get a combination of Users and their ratings.

2nd Table:

We then took the geoplaces dataset and merged it with rcuisine, in order to combine the cuisine type with the restaurant locations.In order to bring in the ratings, we then joined the table to RatingFinal

Once tables created:

We were able to export the tables we brought the data into pandas, and cleaned the data.



Step 02.

Cleaning the Data

We used Pandas to import the final CSV file into a Jupyter notebook and performed a series of steps clean the titles, remove nulls, and drop unnecessary columns.

Data Cleaning

In [1]:
```
# Import our dependencies

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sqlalchemy import create_engine

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import make_classification
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
```

In [2]:
```
engine = create_engine('postgresql://postgres:password@localhost/postgres')
conn = engine.connect()
```

In [3]:
```
data_df = pd.read_sql("select * from finalcombouser", conn)
data_df.head()
```

Out[3]:

	user_id	latitude	longitude	smoker	drink_level	dress_perference	ambience	transport	marital_status	hijos	...	activity	color	weight	budget
0	U1001	22.139997	-100.978803	false	abstemious	informal	family	on foot	single	independent	...	student	black	69	medium
1	U1001	22.139997	-100.978803	false	abstemious	informal	family	on foot	single	independent	...	student	black	69	medium
2	U1001	22.139997	-100.978803	false	abstemious	informal	family	on foot	single	independent	...	student	black	69	medium
3	U1001	22.139997	-100.978803	false	abstemious	informal	family	on foot	single	independent	...	student	black	69	medium
4	U1001	22.139997	-100.978803	false	abstemious	informal	family	on foot	single	independent	...	student	black	69	medium

5 rows × 24 columns

In [6]:
```
combo_user = combo_user[(combo_user.smoker != '?') & (combo_user.drink_level != '?') & (combo_user.dress_perference != '?') &\
                        (combo_user.ambience != '?') & (combo_user.transport != '?') & (combo_user.marital_status != '?')\
                            & (combo_user.hijos != '?') & (combo_user.birth_year != '?') & (combo_user.activity != '?')\
                          & (combo_user.budget != '?')]
```

In [ ]:
```
```

In [7]:
```
new_data_df = combo_user.drop(["user_id", "latitude", "longitude", "placeid"], axis='columns')
new_data_df.head()
```

Out[7]:

	smoker	drink_level	dress_perference	ambience	transport	marital_status	hijos	birth_year	activity	budget	rating	food_rating	service_rating	rcui
0	false	abstemious	informal	family	on foot	single	independent	1989	student	medium	2	2	1	Ame
1	false	abstemious	informal	family	on foot	single	independent	1989	student	medium	1	1	1	Ame
2	false	abstemious	informal	family	on foot	single	independent	1989	student	medium	1	1	2	Ame
3	false	abstemious	informal	family	on foot	single	independent	1989	student	medium	2	2	2	Ame
4	false	abstemious	informal	family	on foot	single	independent	1989	student	medium	1	1	1	Ame

Steps taken to clean dataset:

- ★ Imported dependencies
- ★ Connected to postgres
- ★ Dropped null values and question marks in original dataset

Step 03.

Machine Learning

Attempt 1

- Dropped unused features (color, user_id, latitude, longitude, placeid)
- Determined the number of unique values per feature
- Interrogated the cuisine feature and decided to drop it due to the large number of values
- Ran get_dummies to convert features to numeric values
- Assigned Rating as the target and dropped Rating from the features
- Split Data to Test and Train and ran StandardScaler
- Created RandomForestClassifier and fit it with the data
- **Testing Score** was **0.93711** and **Training Score** of **0.86189**
- Determined the features that had the most impact on the score

Attempt 2

- The most impactful features from Attempt 1 were the service_rating and the food_rating. We figured that was a kind of cheating so we dropped those features.
    - Dropped unused features (food_rating, service_rating)
- We reran the RandomForestClassifier and got passing results again.
- **Training Score** was **0.83682** and the **Test Score** was **0.78668**
- Most impactful features were hijos_kids, hijos_independent,and personality_hunter_ostentatious

Optimization

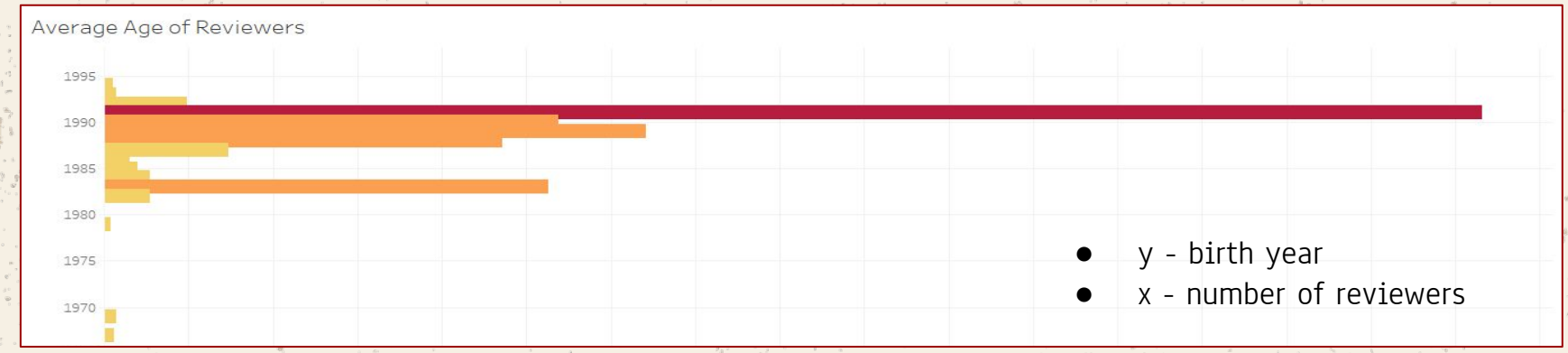
- We took the data preparation from Attempt 2
- Used RandomForestRegressor library to run the Optimization
- Created variables with n_estimators, max_features and random_state
- Fit the data with the Optimizer and waiting 8 minutes for it to run
- The **Best Parameters** were....

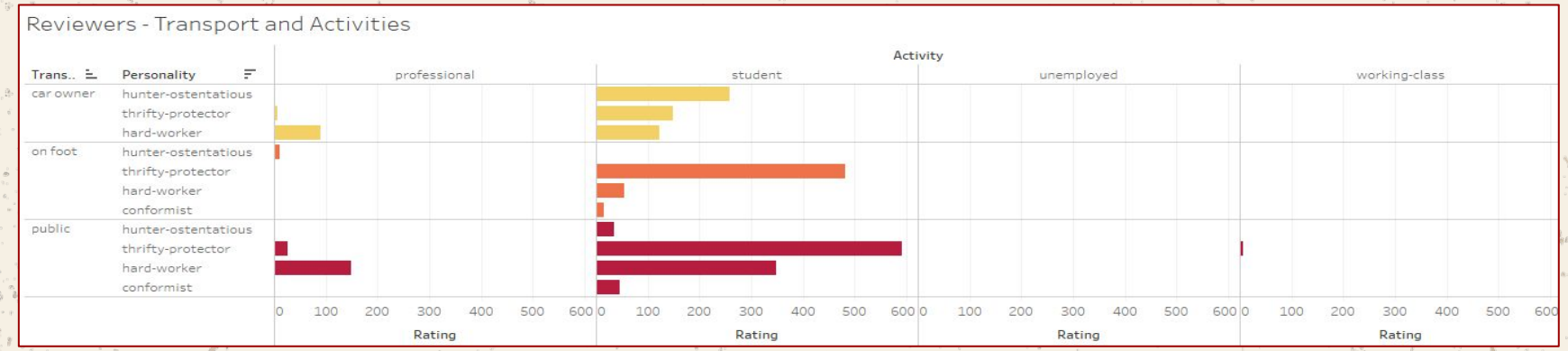
Random grid: {'n_estimators': [20, 50, 100, 200, 300, 500, 700, 900, 1000], 'max_features': ['auto', 'sqrt'], 'random_state': [1, 2, 4, 8, 10]}

Best Parameters: {'random_state': 1, 'n_estimators': 300, 'max_features': 'sqrt'}

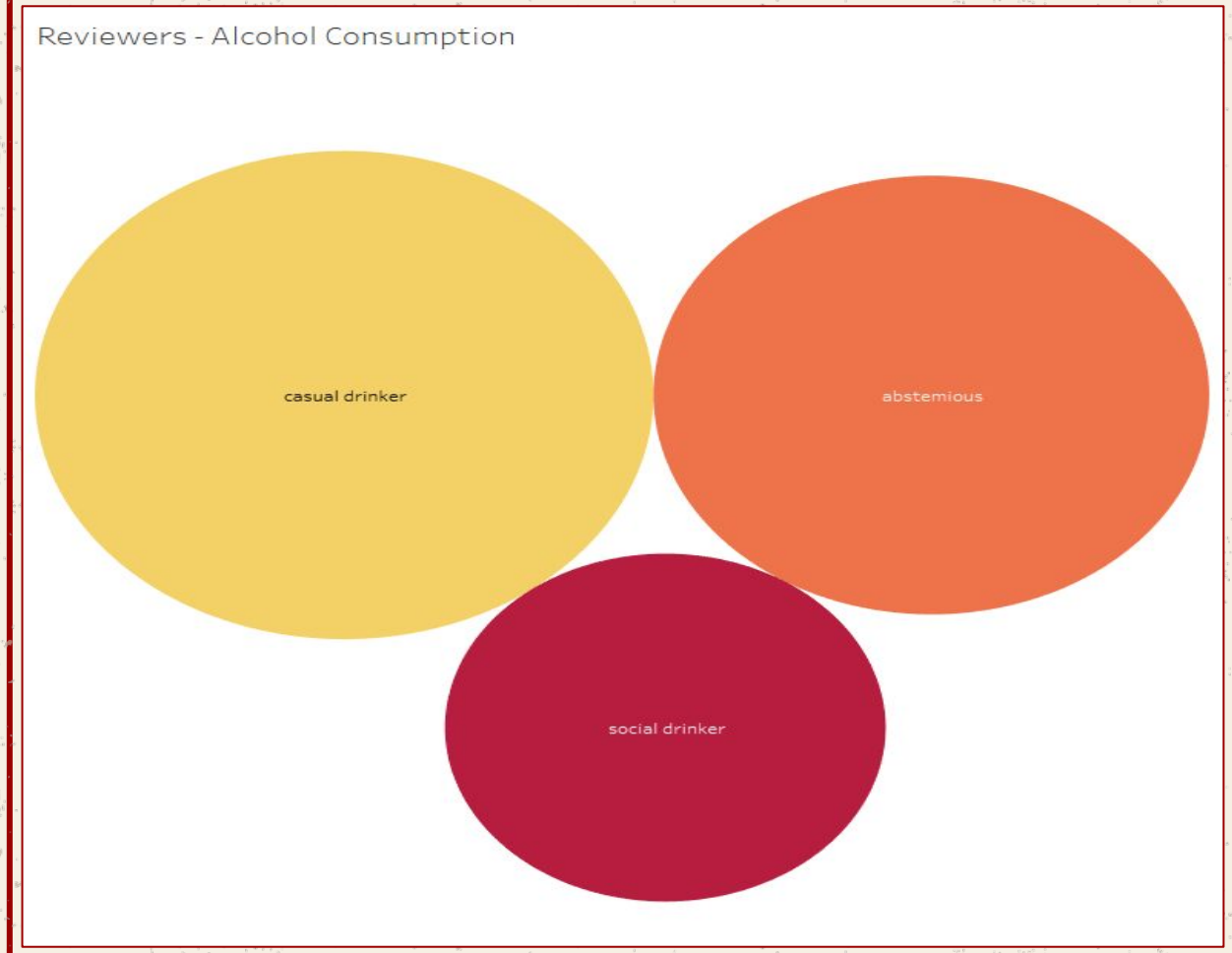
Step 04.

Who Are Our Reviewers?

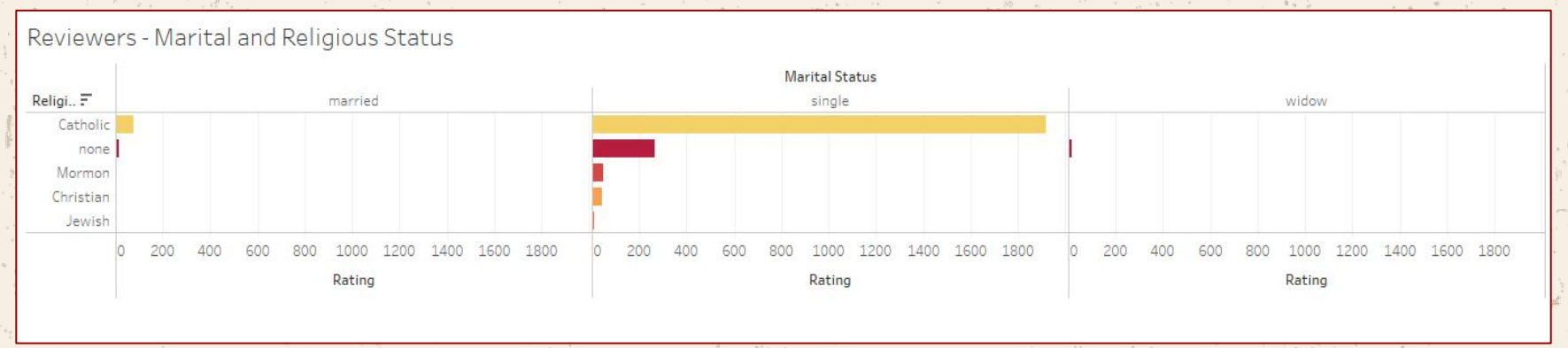


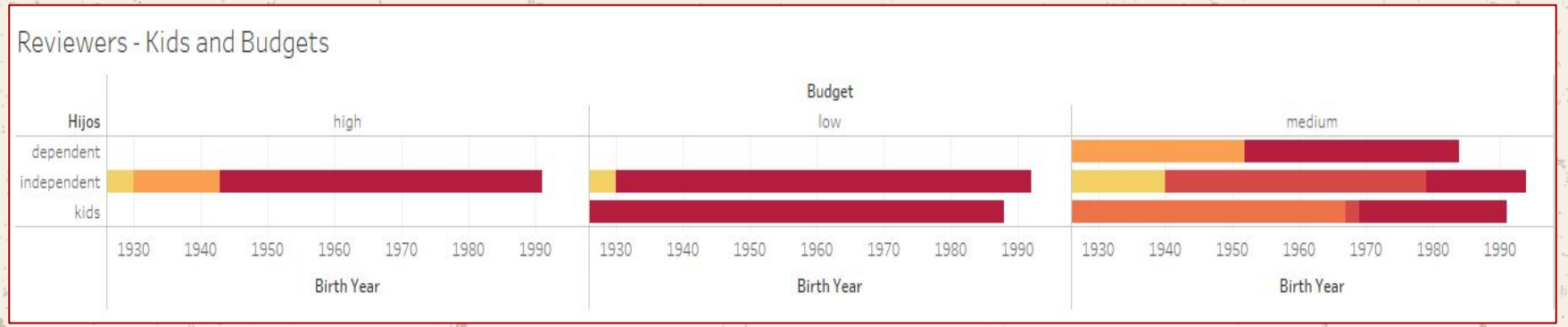


The bulk of our reviewers were Millennials with the top birth year being 1991. This group was primarily made up of students who take public transportation.



The majority of our reviewers classify themselves as casual drinkers with next in line being non-drinkers. Pretty good for a group of students.





Majority of our reviewers are single and catholic with a medium income. The reviewers with a high income were primarily born in 1989 and do not have kids which suggests these are young professionals in the high income category.

Any Questions?