INSY 336 - Final Report

How to Get a Deal on Shark Tank

Group 7

April 13, 2021

Professor Hyunji So

Camille Amyouni, 260838124

Frank Esposito, 260780021

Zehra Kazandag, 260837262

Katrin Maliatski, 260766001

Melis May L Sarfati, 260855675

Introduction

The unexpected COVID-19 pandemic has brought with it many positive and negative unintended consequences. While for many, the pandemic encouraged activities such as mastering cooking and picking up reading, the rise in startup companies is less noticed. Entrepreneurs, stimulated by new ideas and opportunities, are being pushed to think creatively to come up with impactful ideas, much like they did during the 2008 financial crisis. Today, the United States is facing an entrepreneurship boom; in March 2020, there were 804,398 businesses that were less than 1 year old, an increase on the March 2018 figure of 733,825 (Statista, 2020). Many of these businesses are succeeding, with 78 percent of US small businesses currently being profitable (Guidant Financial, 2021). These trends are reassuring amidst the uncertainty brought about by the pandemic, and are inspiring to our group of aspiring entrepreneurs. Naturally, we were interested in seeing what makes entrepreneurs successful, and decided what is better than receiving a deal offer from a shark on the show, Shark Tank? Shark Tank is an American business reality television show where entrepreneurs pitch their business to a panel of five sharks (investors), who decide whether or not to invest in their company. This highly-rated ABC series includes many successful investors such as Kevin O'Leary also known as Mr. Wonderful as a reference to his reputation of being mean, Barbara Corcoran who founded the Corcoran Group, a real estate brokerage in New York City which she sold for $66 million and Mark Cuban, an American billionaire entrepreneur whose net worth is $4.3 billion (ABC, n.d.). Entrepreneurs come from all around the U.S. in hopes of having one of the sharks invest in their company and help them grow it. However, the process can be grueling, and the sharks are known to be ruthless when they believe an idea misses the mark.

Problem

As management students and aspiring entrepreneurs, we put our data analytics skills to use to create a model that would predict which factors contribute to entrepreneur participants receiving a deal on Shark Tank. Using our Kaggle dataset, we explored both binary and continuous variables in terms of distribution and importance to our model (Sathyajit, 2017). With the help of logistic regression, decision trees, random forest and the python SKlearn package, we created a model that predicts what factors make an entrepreneur more likely to land a deal on Shark Tank.

Our goal was to find out if factors such as the location of the venture, amount asked for by the entrepreneur, equity value to be exchanged, venture valuation, category, multiple entrepreneurs, or if there was a particular shark or shark duo that makes a deal more likely to be made with the sharks. This report will outline the steps we took in detail while explaining the choices we made along the way and the rationale behind our model. Our results will also be discussed to advise future Shark Tank participants how to increase their odds of getting a deal on Shark Tank.

Data and Approach

Before building our model, we first began by exploring the variables individually. Our dataset included 495 observations, 18 independent variables, and a binary target variable which is whether the entrepreneur got a deal with a shark or not. Of these 18 predictors, we distinguished both binary and continuous variables that are going to help us predict the best model. This dataset includes continuous variables "askedFor," which is how much funding entrepreneurs would like to receive from the investors, "exchangeForStake" which is the equity percentage of the company that they will be giving to the investor in the case of a deal, and "valuation" which is the value in dollars of the participant venture. There is also the season and the episode number that was not used for building the model, as it did not contribute to answering our problem of providing advice for future entrepreneurs. The dataset also included a description of each company as well as the category of the business, which ranges from baby products and kitchen tools to alcoholic beverages and party supplies. We were also provided with the location of the startup, the name of the entrepreneur, the website of the company (if they have one) and the names of the five sharks that were present during the pitch. By exploring the variables individually, we were able to learn the distribution of our variables. For example, we discovered that the startups' valuations ranged from values as low as $40,000 to as high as $30 million. Furthermore, the asking price to investors ranged from $10,000 to as high as $5 million in exchange for up to 100 percent of the company's equity.

Variable Exploration and Findings

The following part of the report will discuss the various steps we took, both preliminary and final, that led us to our conclusive model and results.

After our analysis of the initial data and understanding of the variables at play, we began cleaning our dataset and conducting further variable exploration. In doing so, we decided to remove some columns in order to simplify our initial exploration of variables. Using variables deal, category, location, askedFor, exchangeForStake, valuation, and Multiple Entrepreneurs we ensured that all null observations were removed or updated to ensure all rows contained sufficient data. By observing each variable on its own, we were able to identify variable classifications that could be combined. The 'category' variable initially included 54 different categories and upon further investigation, we noticed that many categories had overlap and could thus be grouped into a more general classification. It was believed that this would be a problem later on following the dummification of variables and would make our final results easier to interpret by limiting the number of variables at play. For example, all accessory related categories (i.e., Women's Apparel, Men's Apparel, Men and Women's Shoes etc.) could be combined under one category named 'Apparel_Shoes.' This same procedure was applied to several different categories, resulting in an updated categorical count of 23. Similar steps were taken with the 'location' variable. Here we decided to take the original location provided as [city, state] and group all locations by state for ease of interpretation and model building.

With our data cleaned and simplified, we could begin conducting variable exploration. Starting off with a broader outlook of the dataset, we charted the deal frequency of entrepreneurial ventures presented on Shark Tank (Figure 1). Here it could be seen that ventures receive a deal at an approximate 50 percent rate. This was useful when interpreting our results and gaining an understanding that succeeding on Shark Tank is a coin toss, to say the least. We then decided to look at deal frequency by category of the project (Figure 2). The top three most common categories of projects were specialty foods, online tech electronics, and shoe apparel having 64, 45, and 45 observations respectively. On the other hand, projects under the category education had the least number of projects at 4, which we were mindful of when creating the model to ensure our variables contribute meaningfully to our model. When stacking deals and categories on the same chart, we were able to see that the majority of categories had an approximate 50 percent split of having received deal to no deal which is logical considering the overall deal frequency of projects (Figure 3). Lastly, we visualized the state variable which revealed that 142 projects originated from California (roughly 28 percent of all projects).

Intuitively, as Shark Tank is filmed in California, there would be more projects from this state due to the proximity for the entrepreneurs in the area (Figure 5).

Having a more in-depth understanding of our dataset we could begin crafting our first attempt at using the selected model, logistic regression. Taking the non-categorical variables: amount asked for by entrepreneur, equity value to be exchanged, and project valuation, we decided to scale these variables. One practical reason to scale these variables in our regression concerns the fact that some variables may have a larger scale compared to others. In our case, a project's equity value exchanged for had a considerably smaller scale compared to valuation and amount asked for by the entrepreneur (See Figure 5 for distribution plots). Following this we dummified all categorical variables and ranked the variables according to their contribution using RFE, an sklearn package performing feature ranking with recursive feature elimination using a logistic regression model. From here we selected the top 20 variables and ran our first logistic regression on the said variables (Figure 6). Our results were underwhelming having an R-Squared of 0.075, meaning only 7.5 percent of deals can be explained by our model, and a logistic regression model accuracy of 53 percent.

In an effort to create a more accurate model, we brainstormed what kind of data would be helpful in determining an entrepreneur's likelihood of securing a deal. We determined that it would be interesting to see if there is a shark or group of sharks that are more likely to offer a deal, however, our dataset did not offer that information. We found a comprehensive dataset from a Shark Tank fan that recorded the sharks or groups of sharks that made deals on the show across 7 seasons (Tecco, 2015). Using the names of the ventures in our dataset, we matched the information with the new dataset and created a column with the initials of the individual or group of sharks that made a deal with a venture. We only lost a dozen of rows due to missing information, allowing us to utilize 480 observations in the new model.

Using RFE, we updated the 20 most contributing variables and added our three continuous variables. The resulting model included deals made by Mark Cuban (Mark) and Lori Greiner (Lori), the duos of: Barbara Corcoran (Barbara) with Lori, Barababara and Robert Herjavec (Rob), Daymond John (Daymond) and Lori, Daymond and Robert, Kevin O'Leary (Kevin) and Lori, Kevin and Rob, Lori and Rob, Mark and Rob, and lastly, triple shark deals with Kevin, Mark, and Robert. The model included venture categories of alcoholic beverages, baby/children, pet products, and specialty food as well as whether the venture has multiple entrepreneurs. Lastly, in addition to the continuous variables of the amount asked for by the entrepreneur, equity value to be exchanged, and venture valuation, the model included ventures from the states of Nevada, Wisconsin, and Tennessee.

While our R-Squared is still quite low at 0.258, meaning only 25.8 percent of deals can be explained by our model, it is a significant improvement from 0.075. Using our test data set, our logistic regression model achieved 73 percent accuracy (Figure 8). We wanted to cross-check our logistic regression model's performance with other model types to see if the results would improve. We created a decision tree model that achieved 69 percent accuracy and a random forest model that achieved 68 percent accuracy, which assured us that the logistic regression model demonstrates the best performance (Figure 9).

Recommendation

Our logistic regression model suggests that getting an offer from Lori or Kevin would likely render a deal for a venture on Shark Tank. However, the extremely high standard error also suggests this may not be a reliable finding. Moreover, deals with multiple sharks also make a venture more likely to secure a deal. Notably, the Kevin and Rob as well as Mark and Rob duos and more unique combinations of shark groups under 'Others' will make a venture more likely to secure a deal.

The categories that are more likely to get a deal include alcoholic beverages and baby/children ventures, while pet products and specialty food categories render a venture less likely to get a deal. Therefore, when brainstorming venture ideas to pitch on Shark Tank, entrepreneurs should know that an alcoholic drink or a child-related business is more likely to receive a deal.

In terms of location, if an entrepreneur is from Nevada, Wisconsin, or Tennessee, they should consider moving to California for more of a 50 percent chance of getting a deal, as ventures from those states are less likely to land a deal on Shark Tank. Moreover, if an entrepreneur has a business partner, they should consider either leaving them or starting another venture on their own, because having multiple entrepreneurs in a Shark Tank pitch renders you less likely to get a deal.

In terms of continuous variables, the higher valuation and equity value a venture has, the lower the chances of it getting a deal. Therefore, entrepreneurs should be conservative with their valuation and equity values in order to increase their chances of securing a deal. On the other hand, entrepreneurs should not be afraid to ask for a bigger investment for their venture from the sharks, as the higher the value, the more likely the entrepreneur is to get a deal.

Conclusion

Looking back on our observations, data analysis, and final model it is important to mention the existence of possible caveats and limitations that may have affected our findings. Starting with the original dataset, we believe that all observations were accurate having reviewed the Kaggle dataset in preliminary stages. Having said that, there is always the chance that the data could be mislabelled due to human error. Considering the data analysis stages of our report, we conducted various explorations and visualization which aided in understanding the data at hand. We believe this was a crucial stage towards understanding the individual independent variables and their distribution across the dependent variable, deal. On balance, our findings should be taken with a grain of salt as there is a lot of variability of factors that lead to a deal that are outside the scope of our model.

Our model presents some interesting conclusions as to what venture characteristics increase or decrease the possibility of leaving the Shark Tank with a deal. Our findings of preferred categories such as alcoholic beverages and baby/children products and presenting a conservative valuation and equity value while asking for a bigger investment provide an answer to our initial problem. Our findings inspired our entrepreneurial minds to create a venture, and we hope it will prompt the readers of this paper to create their own venture and pitch it on Shark Tank using our recommendations.

Bibliography

2021 small business trends & statistics. (2021). Retrieved from: http://www.guidantfinancial.com/small-business-trends/

ABC. (n.d.). Shark Tank - About. Retrieved from: https://abc.com/shows/shark-tank/about-the-show

Department, P., & 20, J. (2021, January 20). New entrepreneurial Businesses U.S. 2020. Retrieved from: https://www.statista.com/statistics/235494/new-entrepreneurial-businesses-in-the-us/

Sathyajit, Rahul. (2017). Shark Tank Pitches. Retrieved from: https://www.kaggle.com/rahulsathyajit/shark-tank-pitches?select=shark_tank.csv

Tecco, Halle. (2015). Shark Tank Investment Data (by @halletecco). Retrieved from: https://docs.google.com/spreadsheets/d/1Lr0gi_QJB_JU0lBMjJ7WiBRxA0loml1FlM-KlmKsaEY/edit#gid=1213351262

Appendix

Figure 1. Deal Frequency Count

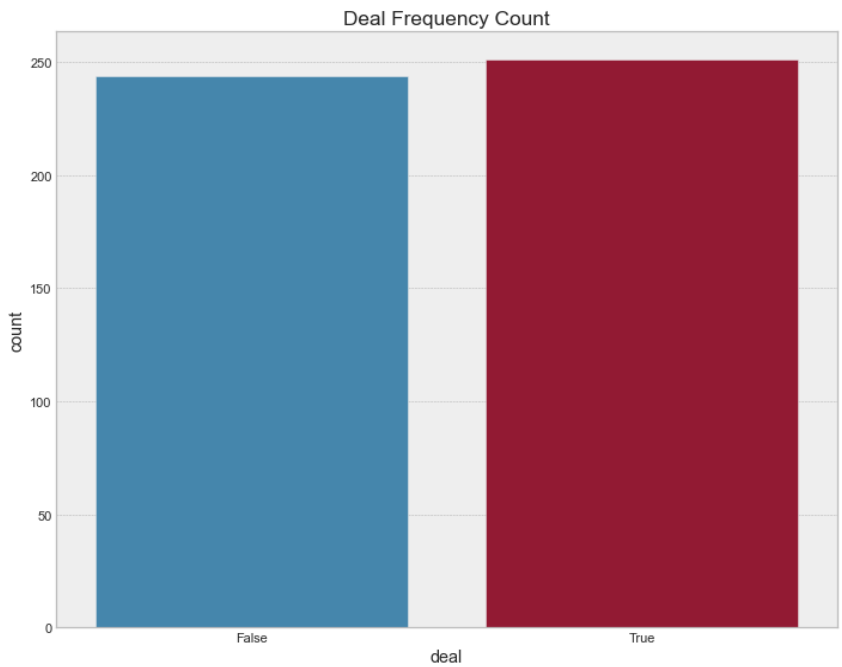


Figure 2. Deal Frequency by Category (Split)

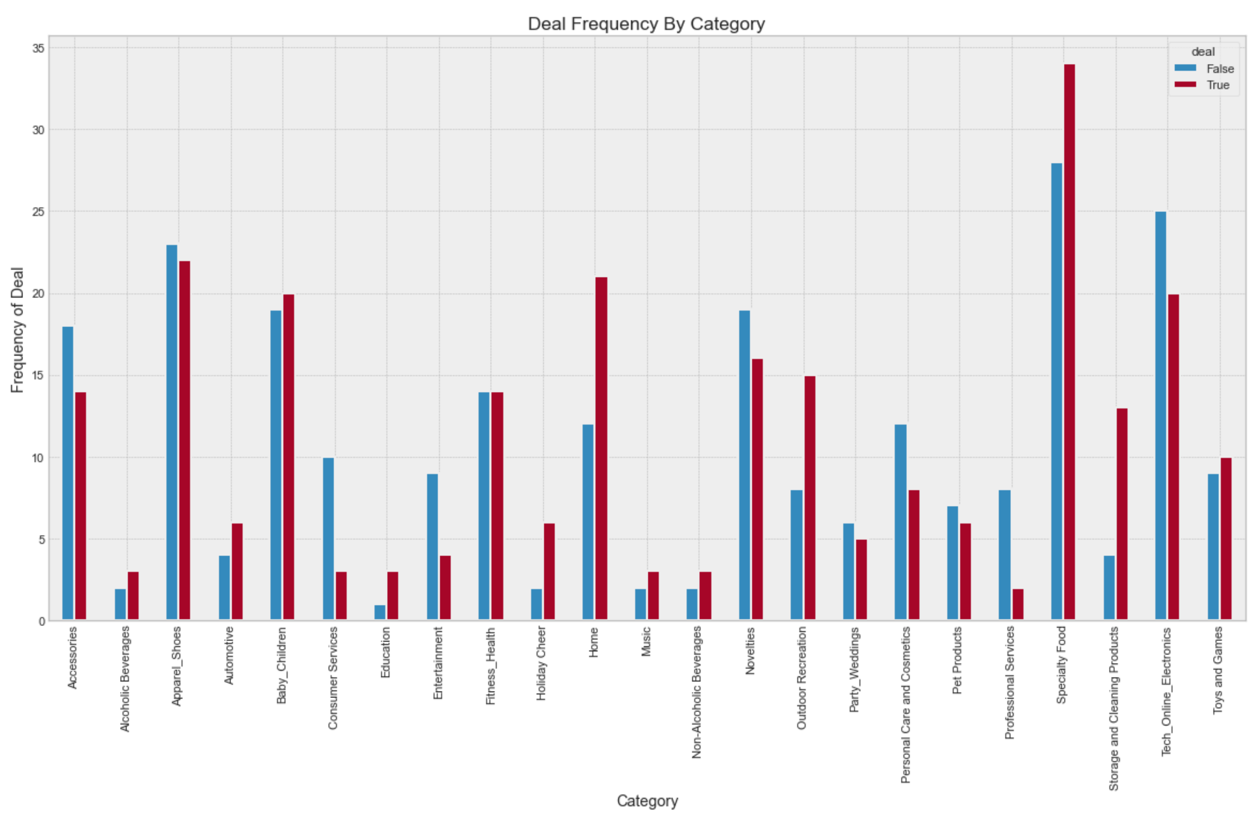


Figure 3. Deal Frequency by Category (Stacked)

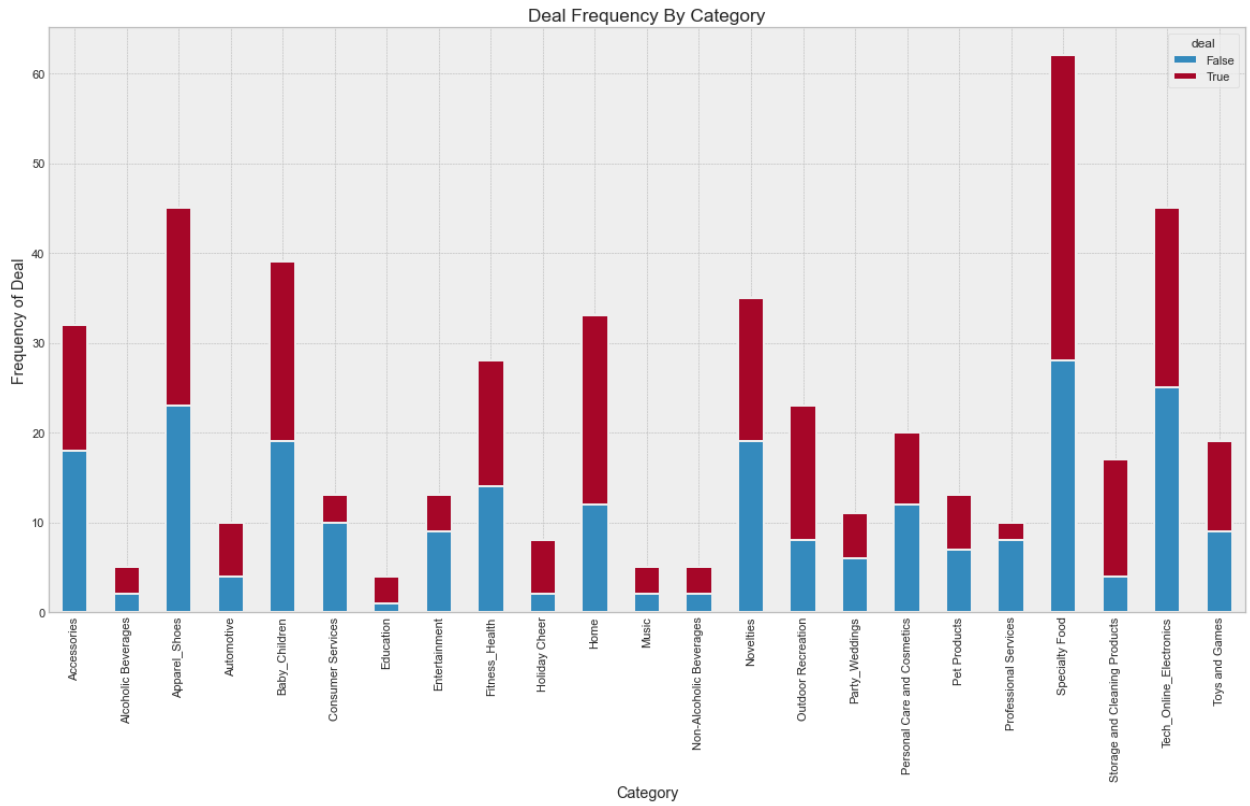


Figure 4. Deal Frequency by State

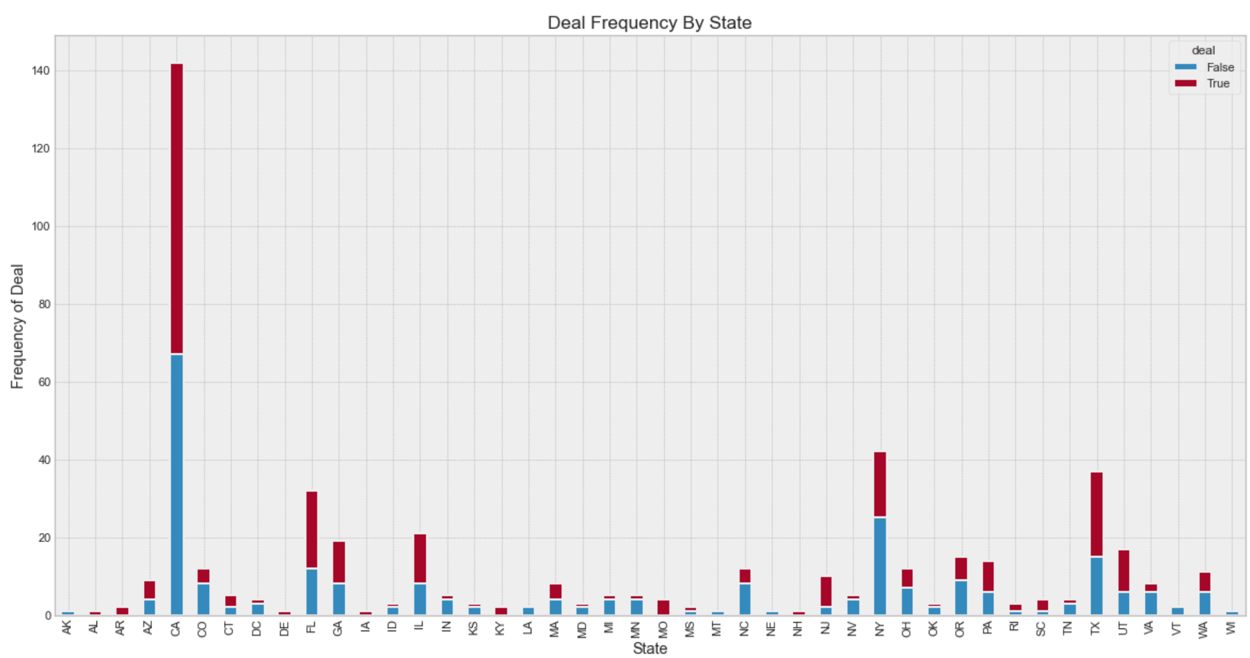
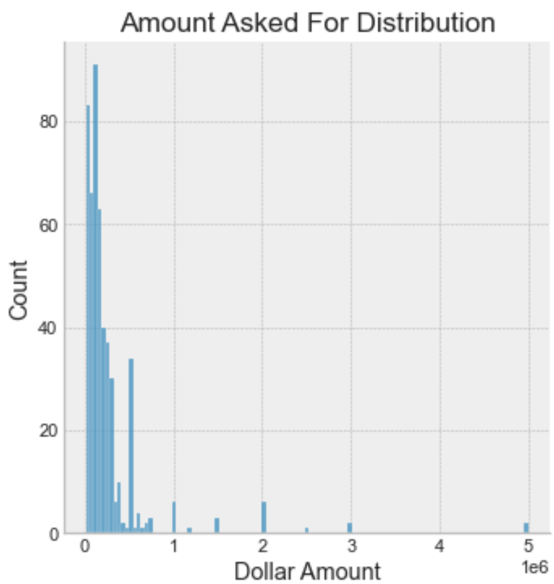
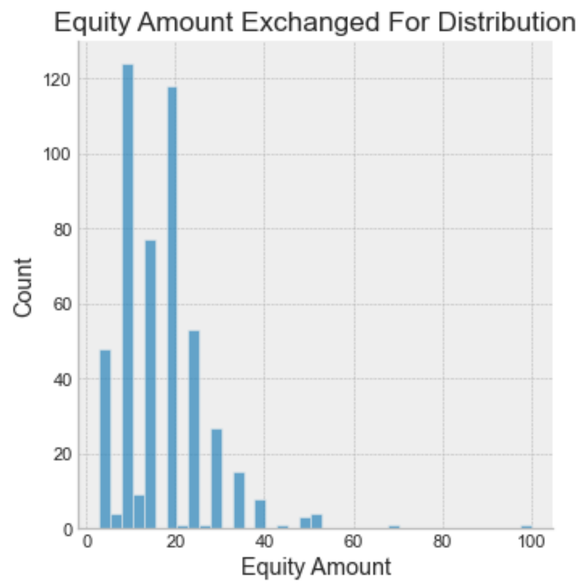


Figure 5. Distribution of Discrete Variables





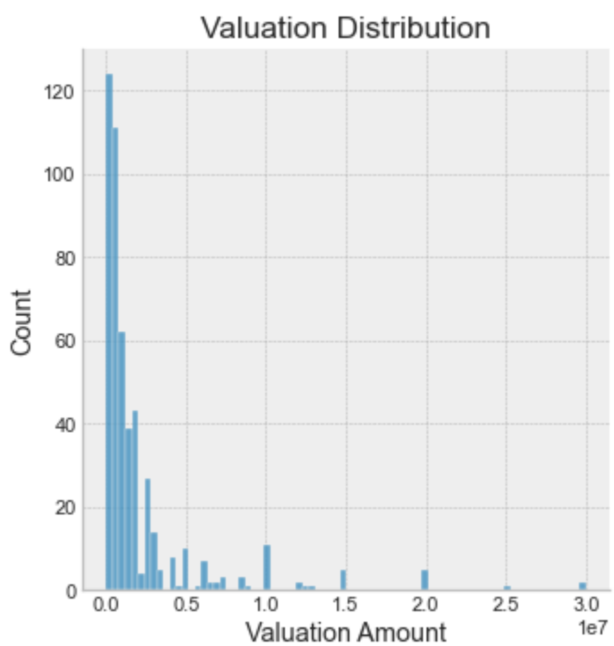


Figure 6. First Attempted Logistic Regression

Results: Logit

Model:	Logit	Pseudo R-squared:	0.075
Dependent Variable:	deal	AIC:	674.7540
Date:	2021-04-11 14:37	BIC:	758.8451
No. Observations:	495	Log-Likelihood:	-317.38
Df Model:	19	LL-Null:	-343.06
Df Residuals:	475	LLR p-value:	8.2365e-05
Converged:	0.0000	Scale:	1.0000
No. Iterations:	35.0000		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
Multiple Entreprenuers	0.2111	0.1965	1.0740	0.2828	-0.1741	0.5962
askedFor	0.0000	0.0000	0.3712	0.7105	-0.0000	0.0000
category_Consumer Services	-1.0547	0.6747	-1.5632	0.1180	-2.3771	0.2677
category_Home	0.6363	0.3913	1.6262	0.1039	-0.1306	1.4032
category_Outdoor Recreation	0.7994	0.4659	1.7158	0.0862	-0.1138	1.7125
category_Professional Services	-1.4530	0.8421	-1.7255	0.0844	-3.1035	0.1975
category_Specialty Food	0.3116	0.2826	1.1027	0.2701	-0.2422	0.8654
category_Storage and Cleaning Products	1.0402	0.6053	1.7184	0.0857	-0.1462	2.2266
exchangeForStake	-0.0253	0.0076	-3.3511	0.0008	-0.0401	-0.0105
state_CA	0.4729	0.2176	2.1730	0.0298	0.0464	0.8994
state_FL	0.9698	0.4081	2.3762	0.0175	0.1699	1.7698
state_NJ	1.8865	0.8453	2.2319	0.0256	0.2298	3.5431
state_NY	0.0599	0.3569	0.1679	0.8666	-0.6396	0.7595
state_TX	0.8295	0.3746	2.2145	0.0268	0.0953	1.5637
valuation	-0.0000	0.0000	-1.3509	0.1767	-0.0000	0.0000
state_UT	0.9055	0.5413	1.6728	0.0944	-0.1554	1.9664
state_IL	0.9163	0.4745	1.9311	0.0535	-0.0137	1.8464
category_Entertainment	-0.7835	0.6415	-1.2212	0.2220	-2.0408	0.4739
category_Holiday Cheer	1.1616	0.8342	1.3925	0.1638	-0.4734	2.7965
state_MO	31.6511	4114708.4547	0.0000	1.0000	-8064648.7269	8064712.0291

Figure 7. Deal Frequency by Category (Stacked by Shark Pairings)

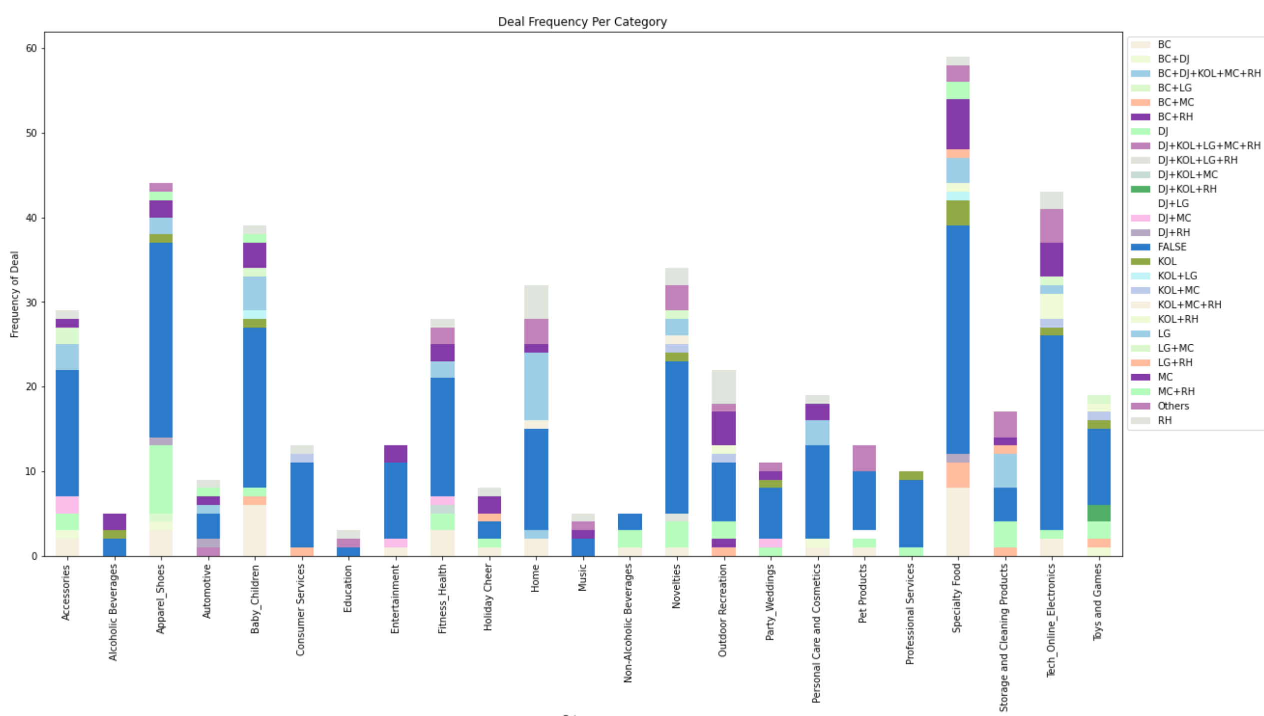


Figure 8. Second Attempted Logistic Regression

```
                                    Results: Logit
=======================================================================================
Model:                      Logit                   Pseudo R-squared:        0.289
Dependent Variable:         deal                    AIC:                     518.9771
Date:                       2021-04-11 11:56        BIC:                     614.9742
No. Observations:           480                     Log-Likelihood:          -236.49
Df Model:                   22                      LL-Null:                 -332.51
Df Residuals:               457                     LLR p-value:             4.0824e-29
Converged:                  0.0000                  Scale:                   1.0000
---------------------------------------------------------------------------------------
                             Coef.    Std.Err.      z     P>|z|      [0.025        0.975]
---------------------------------------------------------------------------------------
Deal_Shark_BC+LG             1.8062       2.4672  0.7321 0.4641        -3.0295        6.6419
Deal_Shark_BC+RH             2.2017       2.4664  0.8927 0.3720        -2.6323        7.0357
Deal_Shark_DJ+LG             2.1871       2.2444  0.9745 0.3298        -2.2119        6.5861
Deal_Shark_DJ+RH             4.5444       4.5862  0.9909 0.3217        -4.4444       13.5332
Deal_Shark_KOL+LG            3.4169       3.1348  1.0900 0.2757        -2.7271        9.5609
Deal_Shark_KOL+MC+RH         3.5472       3.3832  1.0485 0.2944        -3.0838       10.1783
Deal_Shark_KOL+RH            8.6524      24.5764  0.3521 0.7248       -39.5164       56.8212
Deal_Shark_LG               33.0495 2078247.6119  0.0000 1.0000  -4073257.4207  4073323.5197
Deal_Shark_LG+RH             4.4902       4.5231  0.9927 0.3208        -4.3749       13.3553
Deal_Shark_MC               36.1689 9527136.5403  0.0000 1.0000 -18672808.3259 18672880.6637
Deal_Shark_MC+RH             7.8758      16.6972  0.4717 0.6372       -24.8501       40.6017
Deal_Shark_Others           27.3523  124482.2634  0.0002 0.9998   -243953.4007   244008.1053
category_Alcoholic Beverages 1.2051       1.2625  0.9546 0.3398        -1.2693        3.6796
category_Baby_Children       0.0144       0.3962  0.0363 0.9710        -0.7622        0.7910
category_Pet Products       -0.9695       0.8943 -1.0841 0.2783        -2.7223        0.7833
category_Specialty Food     -0.1564       0.3518 -0.4445 0.6566        -0.8459        0.5331
state_NV                    -1.3495       1.6387 -0.8235 0.4102        -4.5613        1.8624
state_WI                    -1.3589       3.4753 -0.3910 0.6958        -8.1704        5.4527
state_TN                    -1.0114       1.5273 -0.6622 0.5079        -4.0049        1.9822
exchangeForStake            -0.2189       0.1384 -1.5817 0.1137        -0.4902        0.0524
askedFor                     0.1219       0.1749  0.6972 0.4857        -0.2208        0.4647
valuation                   -0.3054       0.2124 -1.4375 0.1506        -0.7218        0.1110
Multiple Entreprenuers_False -0.5122      0.1261 -4.0614 0.0000        -0.7594       -0.2650
=======================================================================================
```

Figure 9. DecisionTreeClassifier and RandomForestClassifier Models

```
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)
y_pred = dtc.predict(X_test)
Accuracy = accuracy_score(y_test, y_pred)
Confusion_matrix = confusion_matrix(y_test, y_pred)
print('==='*20)
print('Accuracy = '+str(Accuracy))
print('==='*20)
print(Confusion_matrix)
```

```
============================================================
Accuracy = 0.6916666666666667
============================================================
[[49 18]
 [19 34]]
```

```
RFC = RandomForestClassifier()
RFC.fit(X_train, y_train)
y_pred = RFC.predict(X_test)
Accuracy = accuracy_score(y_test, y_pred)
Confusion_matrix = confusion_matrix(y_test, y_pred)
print('==='*20)
print('Accuracy = '+str(Accuracy))
print('==='*20)
print(Confusion_matrix)
```

```
============================================================
Accuracy = 0.6833333333333333
============================================================
[[47 20]
 [18 35]]
```

INSY 336: Final Project Code

Shark Tank Projects

Group Members:

Camille Amyouni, 260838124

Frank Esposito, 260780021

Zehra Kazandag, 260837262

Katrin Maliatski, 260766001

Melis May L Sarfati, 260855675

Date: April 13, 2021

FIRST MODEL

Library

In [2]:
```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.feature_selection import RFE
import statsmodels.api as sm
```

In [3]:
```
print(plt.style.available)
```

```
['Solarize_Light2', '_classic_test_patch', 'bmh', 'classic', 'dark_backgr
ound', 'fast', 'fivethirtyeight', 'ggplot', 'grayscale', 'seaborn', 'seab
orn-bright', 'seaborn-colorblind', 'seaborn-dark', 'seaborn-dark-palett
e', 'seaborn-darkgrid', 'seaborn-deep', 'seaborn-muted', 'seaborn-noteboo
k', 'seaborn-paper', 'seaborn-pastel', 'seaborn-poster', 'seaborn-talk',
'seaborn-ticks', 'seaborn-white', 'seaborn-whitegrid', 'tableau-colorblin
d10']
```

Importing Data

In [4]:
```
df = pd.read_csv("shark_tank.csv")
df.head()
```

Out[4]:

	deal	description	episode	category	entrepreneurs	location	website
0	False	Bluetooth device implant for your ear.	1	Novelties	Darrin Johnson	St. Paul, MN	NaN
1	True	Retail and wholesale pie factory with two reta...	1	Specialty Food	Tod Wilson	Somerset, NJ	http://whybake.com/
2	True	Ava the Elephant is a godsend for frazzled par...	1	Baby and Child Care	Tiffany Krumins	Atlanta, GA	http://www.avatheelephant.com/
3	False	Organizing, packing, and moving services deliv...	1	Consumer Services	Nick Friedman, Omar Soliman	Tampa, FL	http://collegehunkshaulingjunk.com/
4	False	Interactive media centers for healthcare waiti...	1	Consumer Services	Kevin Flannery	Cary, NC	http://www.wispots.com/

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 495 entries, 0 to 494
Data columns (total 19 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   deal                   495 non-null    bool
 1   description            495 non-null    object
 2   episode                495 non-null    int64
 3   category               495 non-null    object
 4   entrepreneurs          423 non-null    object
 5   location               495 non-null    object
 6   website                457 non-null    object
 7   askedFor               495 non-null    int64
 8   exchangeForStake       495 non-null    int64
 9   valuation              495 non-null    int64
 10  season                 495 non-null    int64
 11  shark1                 495 non-null    object
 12  shark2                 495 non-null    object
 13  shark3                 495 non-null    object
 14  shark4                 495 non-null    object
 15  shark5                 495 non-null    object
 16  title                  495 non-null    object
 17  episode-season         495 non-null    object
 18  Multiple Entreprenuers 495 non-null    bool
dtypes: bool(2), int64(5), object(12)
memory usage: 66.8+ KB
```

Removing some columns

In [6]:
```
# removing columns
cols = ['deal','category','location','askedFor','exchangeForStake','val
df = df.loc[:,cols]
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 495 entries, 0 to 494
Data columns (total 7 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   deal                   495 non-null    bool
 1   category               495 non-null    object
 2   location               495 non-null    object
 3   askedFor               495 non-null    int64
 4   exchangeForStake       495 non-null    int64
 5   valuation              495 non-null    int64
 6   Multiple Entreprenuers 495 non-null    bool
dtypes: bool(2), int64(3), object(2)
memory usage: 20.4+ KB
```

In [8]:
```
df.head()
```

Out[8]:

	deal	category	location	askedFor	exchangeForStake	valuation	Multiple Entreprenuers
0	False	Novelties	St. Paul, MN	1000000	15	6666667	False
1	True	Specialty Food	Somerset, NJ	460000	10	4600000	False
2	True	Baby and Child Care	Atlanta, GA	50000	15	333333	False
3	False	Consumer Services	Tampa, FL	250000	25	1000000	False
4	False	Consumer Services	Cary, NC	1200000	10	12000000	False

In [9]:

```python
# checking the number of categories...
df.category.value_counts()

# might want to combine some of the smaller categories i.e. home, home
```

Out[9]:

```
Specialty Food                                  62
Novelties                                       35
Baby and Child Care                             24
Online Services                                 22
Personal Care and Cosmetics                     20
Toys and Games                                  19
Storage and Cleaning Products                   17
Outdoor Recreation                              16
Electronics                                     14
Entertainment                                   13
Pet Products                                    13
Consumer Services                               13
Kitchen Tools                                   12
Automotive                                      10
Professional Services                           10
Women's Apparel                                 10
Men and Women's Apparel                          9
Baby and Children's Entertainment                9
Women's Accessories                              8
Baby and Children's Apparel and Accessories      8
Holiday Cheer                                    8
Undergarments and Basics                         7
Fitness Programs                                 7
Home Accessories                                 7
Fitness Apparel and Accessories                  6
Homeopathic Remedies                             6
Weddings                                         6
Men's Accessories                                5
Productivity Tools                               5
Alcoholic Beverages                              5
Non-Alcoholic Beverages                          5
Music                                            5
Men and Women's Shoes                            5
Party Supplies                                   5
Furniture                                        5
Gardening                                        5
Home Improvement                                 5
Health and Well-Being                            5
Mobile Apps                                      4
Golf Products                                    4
Education                                        4
Fitness Equipment                                4
Women's Shoes                                    4
Men and Women's Accessories                      4
Pest Control                                     3
Water Bottles                                    3
Wine Accessories                                 3
Cycling                                          3
Home Security Solutions                          3
Maternity                                        2
Fashion Accessories                              2
Costumes                                         2
```

```
Baby and Children's Bedding                    2
Baby and Children's Food                       2
Name: category, dtype: int64
```

In [10]:

```
# Combining some categories
df['category']=np.where(df['category'] =="Women's Apparel", 'Apparel_Sh
df['category']=np.where(df['category'] =="Men and Women's Apparel", 'Ap
df['category']=np.where(df['category'] =="Baby and Children's Apparel a
df['category']=np.where(df['category'] =="Undergarments and Basics", 'A
df['category']=np.where(df['category'] =="Men and Women's Shoes", 'Appa
df['category']=np.where(df['category'] =="Costumes", 'Apparel_Shoes', d
df['category']=np.where(df['category'] =="Women's Shoes", 'Apparel_Shoe

df['category']=np.where(df['category'] =="Fitness Programs", 'Fitness_H
df['category']=np.where(df['category'] =="Fitness Apparel and Accessori
df['category']=np.where(df['category'] =="Fitness Equipment", 'Fitness_
df['category']=np.where(df['category'] =="Health and Well-Being", 'Fitn
df['category']=np.where(df['category'] =="Homeopathic Remedies", 'Fitne


df['category']=np.where(df['category'] =="Women's Accessories", 'Access
df['category']=np.where(df['category'] =="Home Accessories", 'Accessori
df['category']=np.where(df['category'] =="Men's Accessories", 'Accessor
df['category']=np.where(df['category'] =="Wine Accessories", 'Accessori
df['category']=np.where(df['category'] =="Fashion Accessories", 'Access
df['category']=np.where(df['category'] =="Water Bottles", 'Accessories'
df['category']=np.where(df['category'] =="Men and Women's Accessories",

df['category']=np.where(df['category'] =="Baby and Children's Bedding",
df['category']=np.where(df['category'] =="Baby and Children's Food", 'B
df['category']=np.where(df['category'] =="Baby and Children's Apparel a
df['category']=np.where(df['category'] =="Baby and Children's Entertain
df['category']=np.where(df['category'] =="Baby and Child Care", 'Baby_C
df['category']=np.where(df['category'] =="Maternity", 'Baby_Children',

df['category']=np.where(df['category'] =="Kitchen Tools", 'Home', df['c
df['category']=np.where(df['category'] =="Gardening", 'Home', df['categ
df['category']=np.where(df['category'] =="Furniture", 'Home', df['categ
df['category']=np.where(df['category'] =="Home Improvement", 'Home', df
df['category']=np.where(df['category'] =="Pest Control", 'Home', df['ca
df['category']=np.where(df['category'] =="Home Security Solutions", 'Ho

df['category']=np.where(df['category'] =="Golf Products", 'Outdoor Recr
df['category']=np.where(df['category'] =="Cycling", 'Outdoor Recreation

df['category']=np.where(df['category'] =="Online Services", 'Tech_Onlin
df['category']=np.where(df['category'] =="Electronics", 'Tech_Online_El
df['category']=np.where(df['category'] =="Mobile Apps", 'Tech_Online_El
df['category']=np.where(df['category'] =="Productivity Tools", 'Tech_On

df['category']=np.where(df['category'] =="Party Supplies", 'Party_Weddi
df['category']=np.where(df['category'] =="Weddings", 'Party_Weddings',
```

In [11]:
```
# check
df.category.value_counts()
```

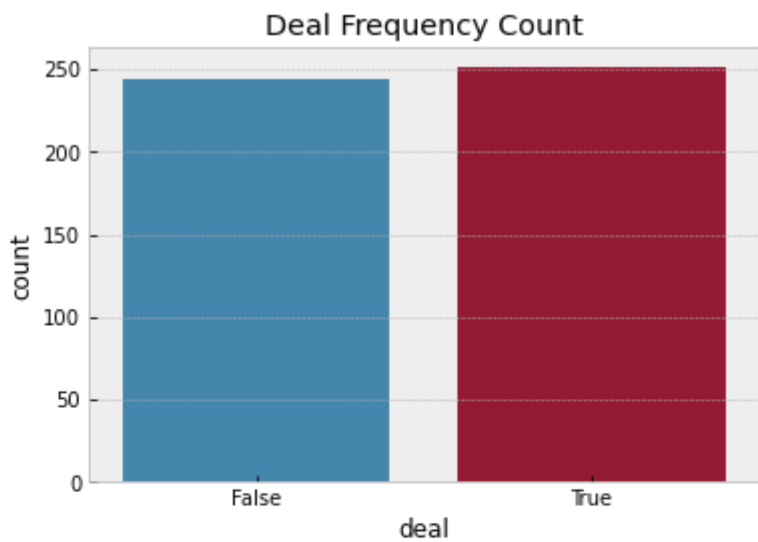
Out[11]:
```
Specialty Food                     62
Apparel_Shoes                      45
Tech_Online_Electronics            45
Baby_Children                      39
Novelties                          35
Home                               33
Accessories                        32
Fitness_Health                     28
Outdoor Recreation                 23
Personal Care and Cosmetics        20
Toys and Games                     19
Storage and Cleaning Products      17
Consumer Services                  13
Entertainment                      13
Pet Products                       13
Party_Weddings                     11
Professional Services              10
Automotive                         10
Holiday Cheer                       8
Non-Alcoholic Beverages             5
Alcoholic Beverages                 5
Music                               5
Education                           4
Name: category, dtype: int64
```

Variable Exploration

Deal Frequency

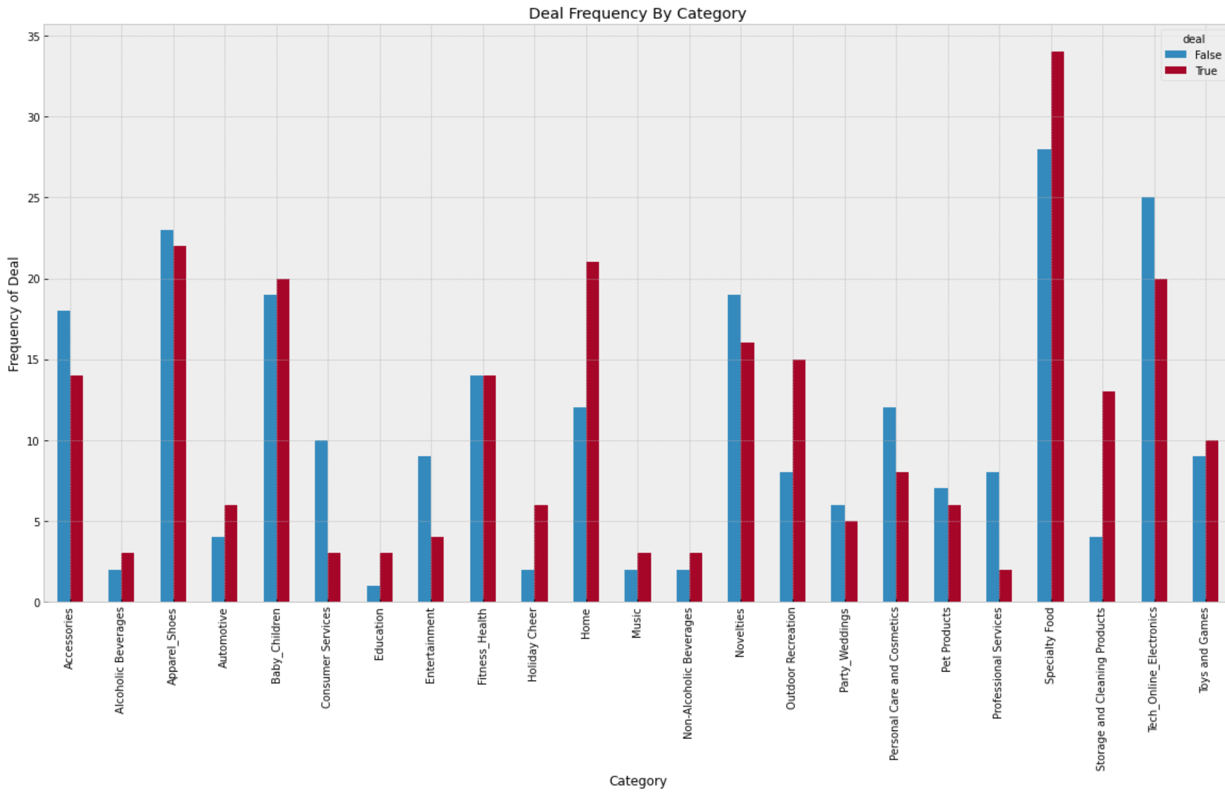
In [28]:
```
plt1 = sns.countplot(x='deal',data=df)
plt.title('Deal Frequency Count')
plt.style.use('bmh')

deal_freq = df['deal'].value_counts("True")
deal_freq
```

Out[28]:
```
True     0.507071
False    0.492929
Name: deal, dtype: float64
```



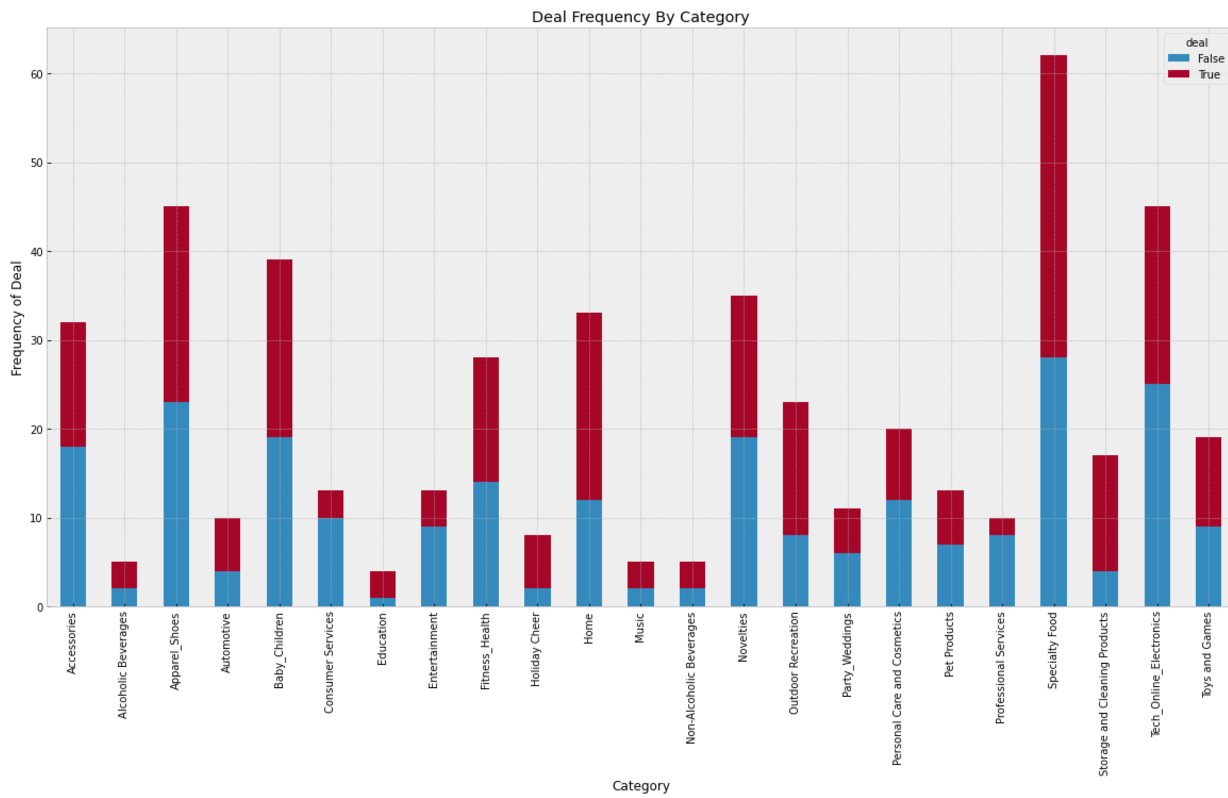
Deal Frequency by Category

In [13]:
```
plt2 = pd.crosstab(df.category,df.deal).plot(kind='bar', stacked = Fals
plt.title('Deal Frequency By Category')
plt.xlabel('Category')
plt.ylabel('Frequency of Deal')
plt.style.use('bmh')
```



In [14]:
```
plt2 = pd.crosstab(df.category,df.deal).plot(kind='bar', stacked = True
plt.title('Deal Frequency By Category')
plt.xlabel('Category')
plt.ylabel('Frequency of Deal')
plt.style.use('bmh')

deal_freq_cat = df['deal'].value_counts("true")
deal_freq_cat
```

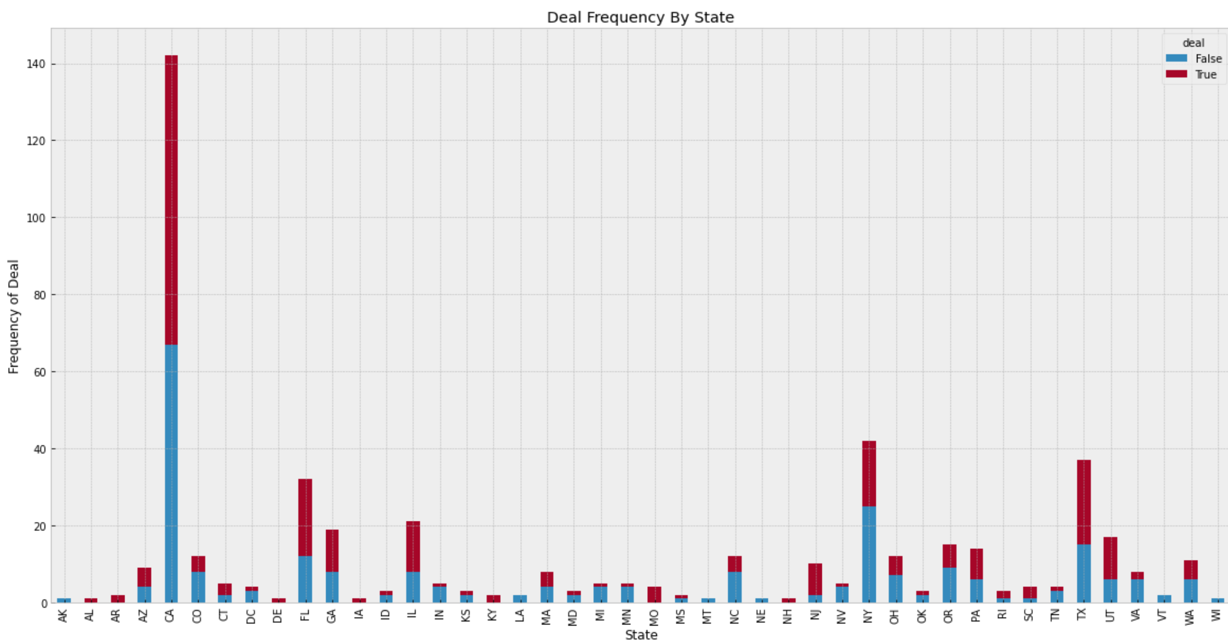
Out[14]:
```
True     0.507071
False    0.492929
Name: deal, dtype: float64
```



Deal Frequency by State

In [15]:
```
# there are a lot of location when we do it by city
df.location
# add column: State
df['state'] = df['location'].str[-2:]
```

In [16]:
```
plt3 = pd.crosstab(df.state,df.deal).plot(kind='bar', stacked = True, l
plt.title('Deal Frequency By State')
plt.xlabel('State')
plt.ylabel('Frequency of Deal')
plt.style.use('bmh')
```



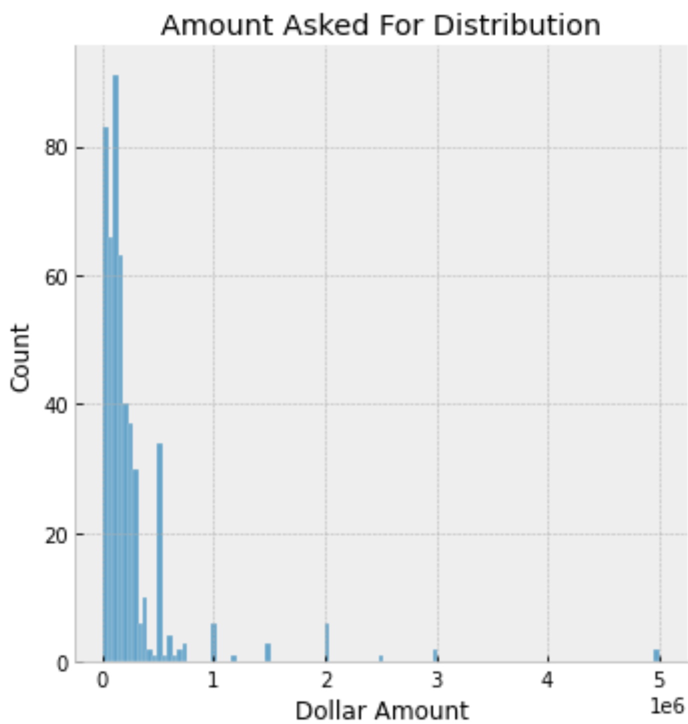
In [17]:
```
deal_freq_state = df.state.value_counts() / 495
deal_freq_state *100
```

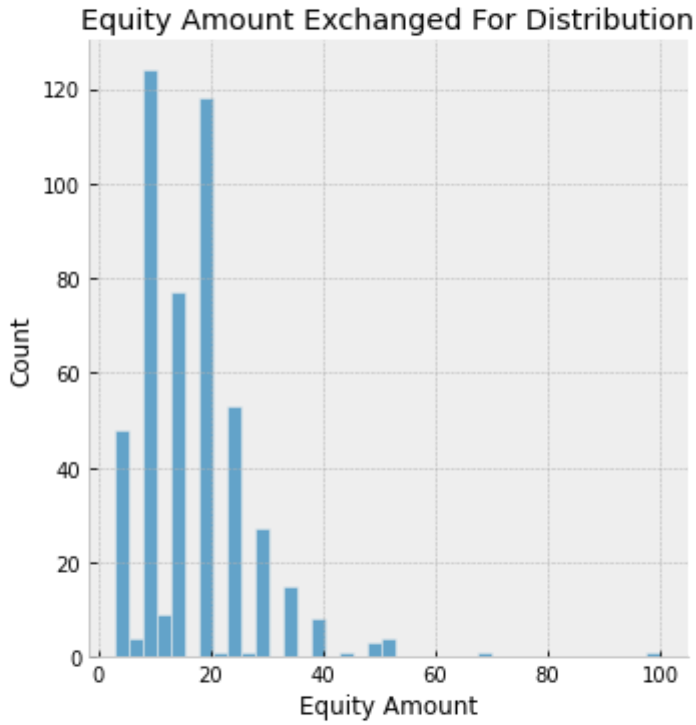
Out[17]:
```
CA    28.686869
NY     8.484848
TX     7.474747
FL     6.464646
IL     4.242424
GA     3.838384
UT     3.434343
OR     3.030303
PA     2.828283
CO     2.424242
OH     2.424242
NC     2.424242
WA     2.222222
NJ     2.020202
AZ     1.818182
VA     1.616162
MA     1.616162
IN     1.010101
MN     1.010101
```

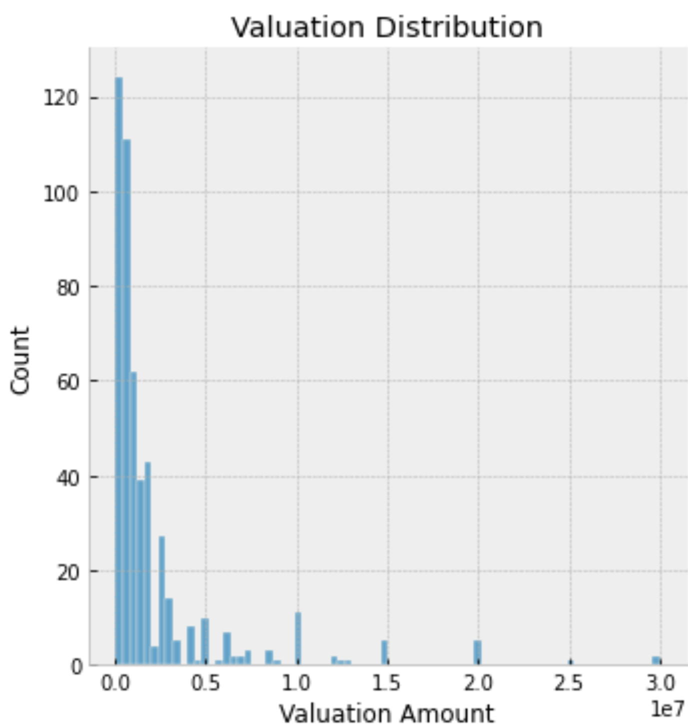
Variable Scales (askedFor, exchangedForStake, valuation)

In [18]:

```
plt1 = sns.displot(x='askedFor',data=df)
plt.title('Amount Asked For Distribution')
plt.xlabel('Dollar Amount')
plt.style.use('bmh')

plt1 = sns.displot(x='exchangeForStake',data=df)
plt.title('Equity Amount Exchanged For Distribution')
plt.xlabel('Equity Amount')
plt.style.use('bmh')

plt1 = sns.displot(x='valuation',data=df)
plt.title('Valuation Distribution')
plt.xlabel('Valuation Amount')
plt.style.use('bmh')
```







Scaling Continous Variables

In [19]:

```
# scaling
continuos = ['askedFor','exchangeForStake','valuation']
scaled_features = df.copy()
features = scaled_features[continuos]

scaler = StandardScaler().fit(features.values)
features = scaler.transform(features.values)
scaled_features[continuos] = features
```

Selecting Columns Part 2

In [20]:
```
df.columns
```

Out[20]:
```
Index(['deal', 'category', 'location', 'askedFor', 'exchangeForStake',
       'valuation', 'Multiple Entreprenuers', 'state'],
      dtype='object')
```

In [21]:
```
cols = ['category', 'state','askedFor','exchangeForStake','valuation',
X = df[cols]
y = df.deal
```

Dummification

In [22]:
```
# BEFORE: df = df_dummified = pd.get_dummies(df, drop_first=True)
X = pd.get_dummies(X, drop_first = True)
features = list(X)
```

Feature Selection using SKlearn

In [23]:
```
logreg = LogisticRegression()
rfe = RFE(logreg, 15)
rfe = rfe.fit(X, y.values.ravel())
print(rfe.support_)
print(rfe.ranking_)
print ("Features sorted by their rank:")
print (sorted(zip(map(lambda x: round(x, 2), rfe.ranking_), features)))
```

```
/opt/anaconda3/lib/python3.8/site-packages/sklearn/utils/validation.py:7
0: FutureWarning: Pass n_features_to_select=15 as keyword args. From vers
ion 1.0 (renaming of 0.25) passing these as positional arguments will res
ult in an error
  warnings.warn(f"Pass {args_msg} as keyword args. From version "

[ True  True  True  True False False False False  True False False False
 False  True False False False  True False False False  True  True  True
 False False False False False  True False False False False  True False
 False False False False False False False False False False False False
 False False False False False  True False  True False False False False
 False False False  True False False False False False]
[ 1  1  1  1 31 54 18 29  1 19  4 50  5  1 33 35 20  1 52 10 48  1  1  1
 12 32 41 23 34  1  8 37 28 38  1 11 40 47  3 14 45 22 27 53 44 15 16  6
 55 43  9 42 39  1 13  1 30 49 17 21 36 24 25  1  2  7 26 51 46]
Features sorted by their rank:
[(1, 'Multiple Entreprenuers'), (1, 'askedFor'), (1, 'category_Consumer S
ervices'), (1, 'category_Home'), (1, 'category_Outdoor Recreation'), (1,
'category_Professional Services'), (1, 'category_Specialty Food'), (1, 'c
ategory_Storage and Cleaning Products'), (1, 'exchangeForStake'), (1, 'st
ate_CA'), (1, 'state_FL'), (1, 'state_NJ'), (1, 'state_NY'), (1, 'state_T
X'), (1, 'valuation'), (2, 'state_UT'), (3, 'state_IL'), (4, 'category_En
tertainment'), (5, 'category_Holiday Cheer'), (6, 'state_MO'), (7, 'state
_VA'), (8, 'state_CO'), (9, 'state_NC'), (10, 'category_Personal Care and
Cosmetics'), (11, 'state_GA'), (12, 'category_Tech_Online_Electronics'),
(13, 'state_NV'), (14, 'state_IN'), (15, 'state_MI'), (16, 'state_MN'),
(17, 'state_OR'), (18, 'category_Automotive'), (19, 'category_Educatio
n'), (20, 'category_Novelties'), (21, 'state_PA'), (22, 'state_KY'), (23,
'state_AR'), (24, 'state_SC'), (25, 'state_TN'), (26, 'state_VT'), (27,
'state_LA'), (28, 'state_DC'), (29, 'category_Baby_Children'), (30, 'stat
e_OH'), (31, 'category_Alcoholic Beverages'), (32, 'category_Toys and Gam
es'), (33, 'category_Music'), (34, 'state_AZ'), (35, 'category_Non-Alcoho
lic Beverages'), (36, 'state_RI'), (37, 'state_CT'), (38, 'state_DE'), (3
9, 'state_NH'), (40, 'state_IA'), (41, 'state_AL'), (42, 'state_NE'), (4
3, 'state_MT'), (44, 'state_MD'), (45, 'state_KS'), (46, 'state_WI'), (4
7, 'state_ID'), (48, 'category_Pet Products'), (49, 'state_OK'), (50, 'ca
tegory_Fitness_Health'), (51, 'state_WA'), (52, 'category_Party_Wedding
s'), (53, 'state_MA'), (54, 'category_Apparel_Shoes'), (55, 'state_MS')]
```

Logistic Regression

In [24]:

```
cols=['Multiple Entreprenuers','askedFor', 'category_Consumer Services'
      'exchangeForStake','state_CA', 'state_FL', 'state_NJ', 'state_NY'
      'state_UT', 'state_IL', 'category_Entertainment', "category_Holid
X=X[cols]

logit_model=sm.Logit(y,X.astype(float))
result=logit_model.fit()
print(result.summary2())
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.641166
         Iterations: 35
                                  Results: Logit
==========================================================================
============================
Model:                           Logit                            Pseudo R-squ
ared:               0.075
Dependent Variable:              deal                             AIC:
674.7540
Date:                            2021-04-13 19:04                 BIC:
758.8451
No. Observations:                495                              Log-Likeliho
od:                 -317.38
Df Model:                        19                               LL-Null:
-343.06
Df Residuals:                    475                              LLR p-value:
8.2365e-05
Converged:                       0.0000                           Scale:
1.0000
No. Iterations:                  35.0000
--------------------------------------------------------------------------
----------------------------
                                         Coef.    Std.Err.      z     P>|z|
[0.025       0.975]
--------------------------------------------------------------------------
----------------------------
Multiple Entreprenuers                   0.2111       0.1965  1.0740 0.282
8        -0.1741       0.5962
askedFor                                 0.0000       0.0000  0.3712 0.710
5        -0.0000       0.0000
category_Consumer Services              -1.0547       0.6747 -1.5632 0.118
0        -2.3771       0.2677
category_Home                            0.6363       0.3913  1.6262 0.103
9        -0.1306       1.4032
category_Outdoor Recreation              0.7994       0.4659  1.7158 0.086
2        -0.1138       1.7125
category_Professional Services          -1.4530       0.8421 -1.7255 0.084
4        -3.1035       0.1975
category_Specialty Food                  0.3116       0.2826  1.1027 0.270
1        -0.2422       0.8654
category_Storage and Cleaning Products   1.0402       0.6053  1.7184 0.085
7        -0.1462       2.2266
exchangeForStake                        -0.0253       0.0076 -3.3511 0.000
8        -0.0401      -0.0105
```

```
state_CA                                    0.4729       0.2176  2.1730 0.029
8        0.0464       0.8994
state_FL                                    0.9698       0.4081  2.3762 0.017
5        0.1699       1.7698
state_NJ                                    1.8865       0.8453  2.2319 0.025
6        0.2298       3.5431
state_NY                                    0.0599       0.3569  0.1679 0.866
6       -0.6396       0.7595
state_TX                                    0.8295       0.3746  2.2145 0.026
8        0.0953       1.5637
valuation                                  -0.0000       0.0000 -1.3509 0.176
7       -0.0000       0.0000
state_UT                                    0.9055       0.5413  1.6728 0.094
4       -0.1554       1.9664
state_IL                                    0.9163       0.4745  1.9311 0.053
5       -0.0137       1.8464
category_Entertainment                     -0.7835       0.6415 -1.2212 0.222
0       -2.0408       0.4739
category_Holiday Cheer                      1.1616       0.8342  1.3925 0.163
8       -0.4734       2.7965
state_MO                                   31.6511 4114708.4547  0.0000 1.000
0 -8064648.7269 8064712.0291
=========================================================================
============================


/opt/anaconda3/lib/python3.8/site-packages/statsmodels/base/model.py:566:
ConvergenceWarning: Maximum Likelihood optimization failed to converge. C
heck mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

In [25]:

```python
import statsmodels.api as sm
# logit_model=sm.Logit(y,X, missing='drop')
# result=logit_model.fit()
# print(result.summary())
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.25,
```

In [26]:

```python
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

Out[26]: LogisticRegression()

In [27]:

```python
y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.
```

```
Accuracy of logistic regression classifier on test set: 0.53
```

SECOND MODEL

Library/Data

In [121]:
```
import pandas as pd
import numpy as np

df = pd.read_csv("shark_tank2.csv")
```

Scaling Variables

In [122]:
```
scaled_features = df.copy()
```

In [123]:
```
from sklearn.preprocessing import StandardScaler
col_names = ['askedFor', 'exchangeForStake', 'valuation']
features = scaled_features[col_names]
scaler = StandardScaler().fit(features.values)
features = scaler.transform(features.values)
```

In [124]:
```
scaled_features[col_names] = features
df = scaled_features
```

Combining Categories on New Dataset

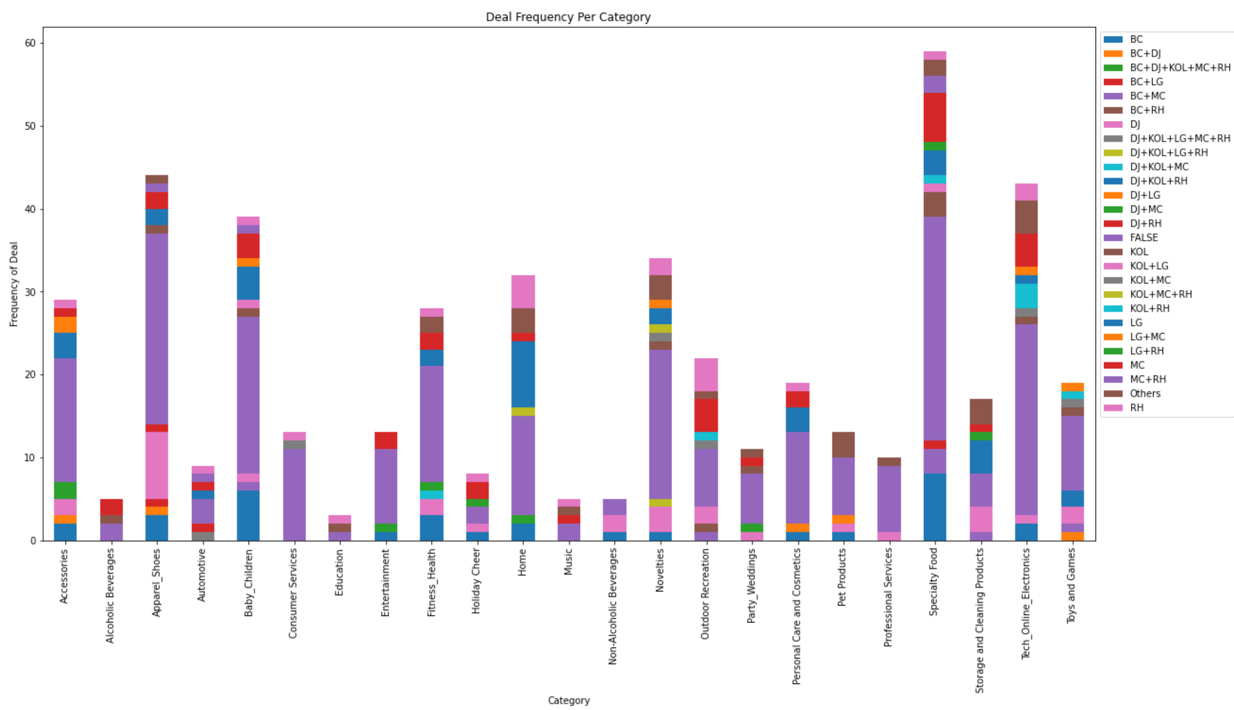
In [125]:

```
df['category']=np.where(df['category'] =="Women's Apparel", 'Apparel_Sh
df['category']=np.where(df['category'] =="Men and Women's Apparel", 'Ap
df['category']=np.where(df['category'] =="Baby and Children's Apparel a
df['category']=np.where(df['category'] =="Undergarments and Basics", 'A
df['category']=np.where(df['category'] =="Men and Women's Shoes", 'Appa
df['category']=np.where(df['category'] =="Costumes", 'Apparel_Shoes', d
df['category']=np.where(df['category'] =="Women's Shoes", 'Apparel_Shoe

df['category']=np.where(df['category'] =="Fitness Programs", 'Fitness_H
df['category']=np.where(df['category'] =="Fitness Apparel and Accessori
df['category']=np.where(df['category'] =="Fitness Equipment", 'Fitness_
df['category']=np.where(df['category'] =="Health and Well-Being", 'Fitn
df['category']=np.where(df['category'] =="Homeopathic Remedies", 'Fitne


df['category']=np.where(df['category'] =="Women's Accessories", 'Access
df['category']=np.where(df['category'] =="Home Accessories", 'Accessori
df['category']=np.where(df['category'] =="Men's Accessories", 'Accessor
df['category']=np.where(df['category'] =="Wine Accessories", 'Accessori
df['category']=np.where(df['category'] =="Fashion Accessories", 'Access
df['category']=np.where(df['category'] =="Water Bottles", 'Accessories'
df['category']=np.where(df['category'] =="Men and Women's Accessories",

df['category']=np.where(df['category'] =="Baby and Children's Bedding",
df['category']=np.where(df['category'] =="Baby and Children's Food", 'B
df['category']=np.where(df['category'] =="Baby and Children's Apparel a
df['category']=np.where(df['category'] =="Baby and Children's Entertain
df['category']=np.where(df['category'] =="Baby and Child Care", 'Baby_C
df['category']=np.where(df['category'] =="Maternity", 'Baby_Children',

df['category']=np.where(df['category'] =="Kitchen Tools", 'Home', df['c
df['category']=np.where(df['category'] =="Gardening", 'Home', df['categ
df['category']=np.where(df['category'] =="Furniture", 'Home', df['categ
df['category']=np.where(df['category'] =="Home Improvement", 'Home', df
df['category']=np.where(df['category'] =="Pest Control", 'Home', df['ca
df['category']=np.where(df['category'] =="Home Security Solutions", 'Ho

df['category']=np.where(df['category'] =="Golf Products", 'Outdoor Recr
df['category']=np.where(df['category'] =="Cycling", 'Outdoor Recreation

df['category']=np.where(df['category'] =="Online Services", 'Tech_Onlin
df['category']=np.where(df['category'] =="Electronics", 'Tech_Online_El
df['category']=np.where(df['category'] =="Mobile Apps", 'Tech_Online_El
df['category']=np.where(df['category'] =="Productivity Tools", 'Tech_On

df['category']=np.where(df['category'] =="Party Supplies", 'Party_Weddi
df['category']=np.where(df['category'] =="Weddings", 'Party_Weddings',
```

Variable Exploration With New Data

In [51]:
```
import matplotlib.pyplot as plt

pd.crosstab(df.category,df.Deal_Shark).plot(kind='bar', stacked = True,
plt.legend(bbox_to_anchor=(1.0, 1.0))
plt.title('Deal Frequency Per Category')
plt.xlabel('Category')
plt.ylabel('Frequency of Deal')
```

Out[51]: Text(0, 0.5, 'Frequency of Deal')



Adding State Variable

In [126]:
```
df['state'] = df['location'].str[-2:]
```

Dummification

In [127]:
```
columns = ['category', 'state','askedFor','exchangeForStake', 'Deal_Sha
X_df = df[columns]
y_df = df.deal
```

In [128]:

```
# Multiple Entreprenuers variable dummification needed to be done outsi
X_df['Multiple Entreprenuers'] = (X_df['Multiple Entreprenuers'] == 'TR
X_df['Multiple Entreprenuers'] = (X_df['Multiple Entreprenuers'] == 'FA
```

```
<ipython-input-128-c0539eb29a05>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  X_df['Multiple Entreprenuers'] = (X_df['Multiple Entreprenuers'] == 'TR
UE').astype(str)
<ipython-input-128-c0539eb29a05>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  X_df['Multiple Entreprenuers'] = (X_df['Multiple Entreprenuers'] == 'FA
LSE').astype(str)
```

In [129]:

```
X_df = pd.get_dummies(X_df)
features = list(X_df)
```

In [130]:

```
# Check
X_df.head()
```

Out[130]:

	askedFor	exchangeForStake	valuation	season	category_Accessories	category_Alcoholic Beverages	cate
0	-0.448355	-0.246259	-0.486032	1	0	0	
1	-0.020493	0.249565	-0.245122	1	0	0	
2	0.193438	-0.246259	0.039590	1	0	0	
3	-0.170245	0.249565	-0.337106	1	0	0	
4	-0.448355	-0.742084	-0.442230	1	0	0	

5 rows × 99 columns

In [131]:
```
1  X_df.info()
```

```
 40  state_IL                                  480 non-null    uint8
 41  state_IN                                  480 non-null    uint8
 42  state_KS                                  480 non-null    uint8
 43  state_KY                                  480 non-null    uint8
 44  state_LA                                  480 non-null    uint8
 45  state_MA                                  480 non-null    uint8
 46  state_MD                                  480 non-null    uint8
 47  state_MI                                  480 non-null    uint8
 48  state_MN                                  480 non-null    uint8
 49  state_MO                                  480 non-null    uint8
 50  state_MS                                  480 non-null    uint8
 51  state_MT                                  480 non-null    uint8
 52  state_NC                                  480 non-null    uint8
 53  state_NE                                  480 non-null    uint8
 54  state_NH                                  480 non-null    uint8
 55  state_NJ                                  480 non-null    uint8
 56  state_NV                                  480 non-null    uint8
 57  state_NY                                  480 non-null    uint8
 58  state_OH                                  480 non-null    uint8
```

Feature Selection using SKlearn

In [40]:

```python
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
rfe = RFE(logreg, 20)
rfe = rfe.fit(X_df, y_df.values.ravel())
print(rfe.support_)
print(rfe.ranking_)
print ("Features sorted by their rank:")
print (sorted(zip(map(lambda x: round(x, 2), rfe.ranking_), features)))
```

```
/opt/anaconda3/lib/python3.8/site-packages/sklearn/utils/validation.py:6
7: FutureWarning: Pass n_features_to_select=20 as keyword args. From vers
ion 0.25 passing these as positional arguments will result in an error
  warnings.warn("Pass {} as keyword args. From version 0.25 "

[False False False False False False  True False False  True False False
 False False False False False False False False False False  True False
  True False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False  True False False
 False False False False False  True False False False False False  True
 False False False  True False  True False False False False False  True
 False  True  True False  True False  True  True  True False  True  True
  True  True]
[42 10 12 77 65 51  1 19 28  1 78  4 33 35 26 72 60 16  7 68 14 71  1 67
  1 24 43 54 61 36 76 64  6 70 27 63  5 39 59 73 40 23 52 34 25 79 69 29
 22 20 75 50  8 56 66  3 48  1 32 47  9 55 58 37 74  1 38 11 30 62 57  1
  2 46 49  1 53  1 44 45 41 17 31  1 15  1  1 21  1 18  1  1  1 13  1  1
  1  1]
Features sorted by their rank:
[(1, 'Deal_Shark_BC+LG'), (1, 'Deal_Shark_BC+RH'), (1, 'Deal_Shark_DJ+L
G'), (1, 'Deal_Shark_DJ+RH'), (1, 'Deal_Shark_FALSE'), (1, 'Deal_Shark_KO
L+LG'), (1, 'Deal_Shark_KOL+MC+RH'), (1, 'Deal_Shark_KOL+RH'), (1, 'Deal_
Shark_LG'), (1, 'Deal_Shark_LG+RH'), (1, 'Deal_Shark_MC'), (1, 'Deal_Shar
k_MC+RH'), (1, 'Deal_Shark_Others'), (1, 'category_Alcoholic Beverages'),
(1, 'category_Baby_Children'), (1, 'category_Pet Products'), (1, 'categor
y_Specialty Food'), (1, 'state_NV'), (1, 'state_TN'), (1, 'state_WI'),
(2, 'Deal_Shark_BC'), (3, 'state_NH'), (4, 'category_Education'), (5, 'st
ate_DE'), (6, 'state_CA'), (7, 'category_Novelties'), (8, 'state_MT'),
(9, 'state_OK'), (10, 'exchangeForStake'), (11, 'state_UT'), (12, 'valuat
ion'), (13, 'Deal_Shark_LG+MC'), (14, 'category_Party_Weddings'), (15, 'D
eal_Shark_DJ+MC'), (16, 'category_Non-Alcoholic Beverages'), (17, 'Deal_S
hark_DJ+KOL+MC'), (18, 'Deal_Shark_KOL+MC'), (19, 'category_Apparel_Shoe
s'), (20, 'state_MN'), (21, 'Deal_Shark_KOL'), (22, 'state_MI'), (23, 'st
ate_IL'), (24, 'category_Storage and Cleaning Products'), (25, 'state_K
Y'), (26, 'category_Holiday Cheer'), (27, 'state_CT'), (28, 'category_Aut
omotive'), (29, 'state_MD'), (30, 'state_VA'), (31, 'Deal_Shark_DJ+KOL+R
H'), (32, 'state_NY'), (33, 'category_Entertainment'), (34, 'state_KS'),
(35, 'category_Fitness_Health'), (36, 'state_AL'), (37, 'state_RI'), (38,
'state_TX'), (39, 'state_FL'), (40, 'state_ID'), (41, 'Deal_Shark_DJ+KOL+
LG+RH'), (42, 'askedFor'), (43, 'category_Tech_Online_Electronics'), (44,
'Deal_Shark_DJ'), (45, 'Deal_Shark_DJ+KOL+LG+MC+RH'), (46, 'Deal_Shark_BC
+DJ'), (47, 'state_OH'), (48, 'state_NJ'), (49, 'Deal_Shark_BC+DJ+KOL+MC+
RH'), (50, 'state_MS'), (51, 'category_Accessories'), (52, 'state_IN'),
(53, 'Deal_Shark_BC+MC'), (54, 'category_Toys and Games'), (55, 'state_O
R'), (56, 'state_NC'), (57, 'state_WA'), (58, 'state_PA'), (59, 'state_G
A'), (60, 'category_Music'), (61, 'state_AK'), (62, 'state_VT'), (63, 'st
```

```
ate_DC'), (64, 'state_AZ'), (65, 'Multiple Entreprenuers'), (66, 'state_N
E'), (67, 'category_Professional Services'), (68, 'category_Outdoor Recre
ation'), (69, 'state_MA'), (70, 'state_CO'), (71, 'category_Personal Care
and Cosmetics'), (72, 'category_Home'), (73, 'state_IA'), (74, 'state_S
C'), (75, 'state_MO'), (76, 'state_AR'), (77, 'season'), (78, 'category_C
onsumer Services'), (79, 'state_LA')]
```

In [132]:

```
cols=['Deal_Shark_BC+LG', 'Deal_Shark_BC+RH', 'Deal_Shark_DJ+LG', 'Deal
      'Deal_Shark_KOL+LG', 'Deal_Shark_KOL+MC+RH', 'Deal_Shark_KOL+RH',
      'Deal_Shark_MC', 'Deal_Shark_MC+RH', 'Deal_Shark_Others', 'catego
      'category_Pet Products', 'category_Specialty Food', 'state_NV', '
      'exchangeForStake', 'askedFor', 'valuation', 'Multiple Entreprenu
X_df = X_df[cols]
```

Logistic Regression

In [133]:

```
import statsmodels.api as sm
logit_model=sm.Logit(y_df,X_df)
result=logit_model.fit(method='bfgs')
print(result.summary2())
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.492685
         Iterations: 35
         Function evaluations: 36
         Gradient evaluations: 36
                                                      Results: Logit
===========================================================================
====================
Model:                         Logit                         Pseudo R-squared:
0.289
Dependent Variable:            deal                          AIC:
518.9771
Date:                          2021-04-11 11:56              BIC:
614.9742
No. Observations:              480                           Log-Likelihood:
-236.49
Df Model:                      22                            LL-Null:
-332.51
Df Residuals:                  457                           LLR p-value:
4.0824e-29
Converged:                     0.0000                        Scale:
1.0000
---------------------------------------------------------------------------
--------------------
                               Coef.    Std.Err.      z     P>|z|      [0.0
25          0.975]
---------------------------------------------------------------------------
--------------------
Deal_Shark_BC+LG               1.8062       2.4672  0.7321 0.4641         -
3.0295         6.6419
Deal_Shark_BC+RH               2.2017       2.4664  0.8927 0.3720         -
2.6323         7.0357
Deal_Shark_DJ+LG               2.1871       2.2444  0.9745 0.3298         -
2.2119         6.5861
Deal_Shark_DJ+RH               4.5444       4.5862  0.9909 0.3217         -
4.4444        13.5332
Deal_Shark_KOL+LG              3.4169       3.1348  1.0900 0.2757         -
2.7271         9.5609
Deal_Shark_KOL+MC+RH           3.5472       3.3832  1.0485 0.2944         -
3.0838        10.1783
Deal_Shark_KOL+RH              8.6524      24.5764  0.3521 0.7248        -3
9.5164        56.8212
Deal_Shark_LG                 33.0495 2078247.6119  0.0000 1.0000  -407325
7.4207  4073323.5197
Deal_Shark_LG+RH               4.4902       4.5231  0.9927 0.3208         -
4.3749        13.3553
Deal_Shark_MC                 36.1689 9527136.5403  0.0000 1.0000 -1867280
8.3259 18672880.6637
Deal_Shark_MC+RH               7.8758      16.6972  0.4717 0.6372        -2
4.8501        40.6017
Deal_Shark_Others             27.3523  124482.2634  0.0002 0.9998   -24395
3.4007   244008.1053
```

```
category_Alcoholic Beverages  1.2051       1.2625  0.9546 0.3398         -
1.2693         3.6796
category_Baby_Children        0.0144       0.3962  0.0363 0.9710         -
0.7622         0.7910
category_Pet Products        -0.9695       0.8943 -1.0841 0.2783         -
2.7223         0.7833
category_Specialty Food      -0.1564       0.3518 -0.4445 0.6566         -
0.8459         0.5331
state_NV                     -1.3495       1.6387 -0.8235 0.4102         -
4.5613         1.8624
state_WI                     -1.3589       3.4753 -0.3910 0.6958         -
8.1704         5.4527
state_TN                     -1.0114       1.5273 -0.6622 0.5079         -
4.0049         1.9822
exchangeForStake             -0.2189       0.1384 -1.5817 0.1137         -
0.4902         0.0524
askedFor                      0.1219       0.1749  0.6972 0.4857         -
0.2208         0.4647
valuation                    -0.3054       0.2124 -1.4375 0.1506         -
0.7218         0.1110
Multiple Entreprenuers_False -0.5122       0.1261 -4.0614 0.0000         -
0.7594        -0.2650
=========================================================================
====================


/opt/anaconda3/lib/python3.8/site-packages/statsmodels/base/model.py:566:
ConvergenceWarning: Maximum Likelihood optimization failed to converge. C
heck mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

In [134]:

```python
import statsmodels.api as sm
# logit_model=sm.Logit(y,X, missing='drop')
# result=logit_model.fit()
# print(result.summary())
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test=train_test_split(X_df, y_df, test_size
```

In [135]:

```python
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

Out[135]: LogisticRegression()

In [136]:

```python
y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.
```

```
Accuracy of logistic regression classifier on test set: 0.73
```

In [137]:

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

       False       0.71      0.90      0.79        67
        True       0.80      0.53      0.64        53

    accuracy                           0.73       120
   macro avg       0.75      0.71      0.71       120
weighted avg       0.75      0.73      0.72       120
```

In [138]:

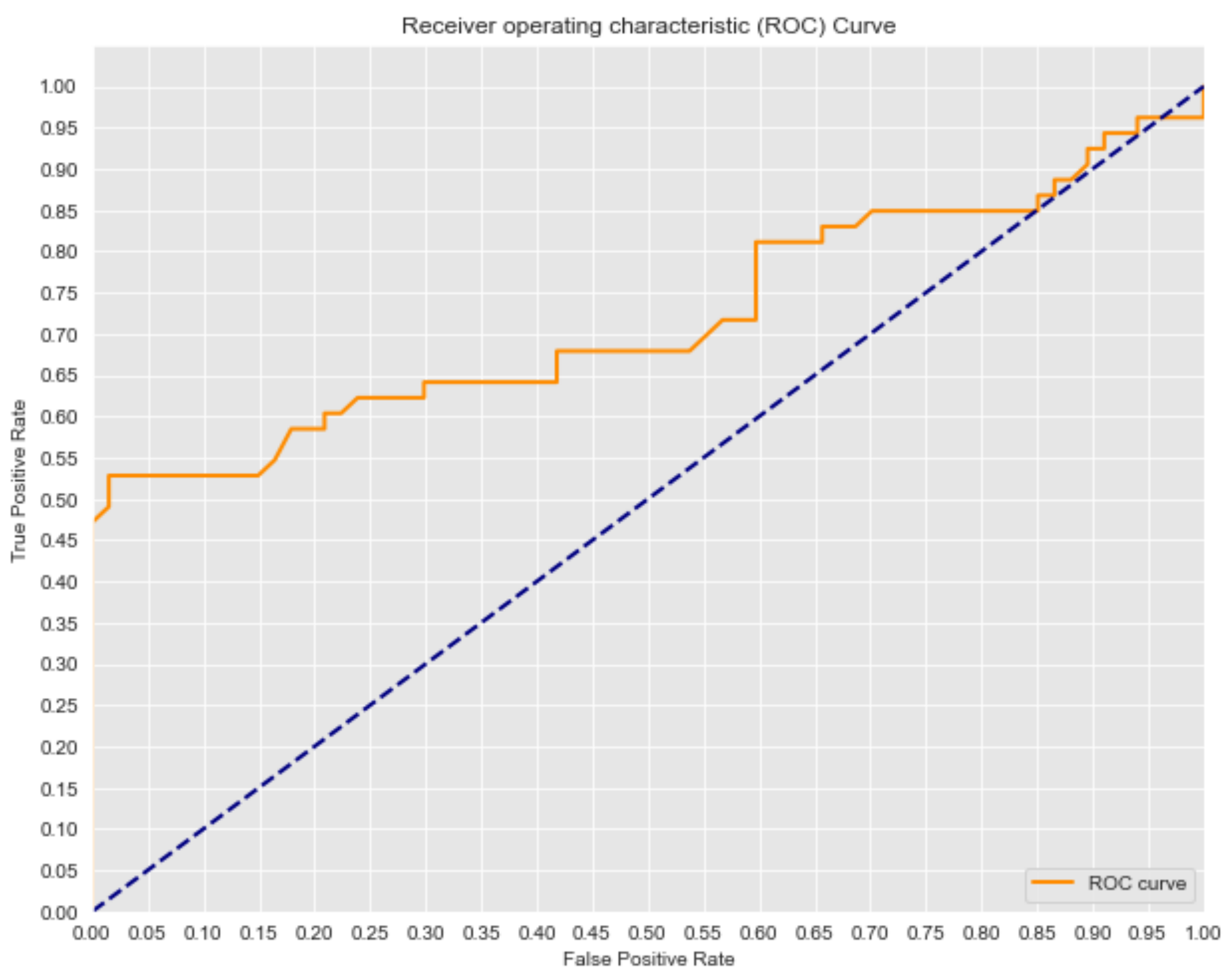
```
from sklearn.metrics import roc_auc_score, auc
from sklearn.metrics import roc_curve

y_score = logreg.fit(X_train, y_train).decision_function(X_test)
fpr, tpr, thresholds = roc_curve(y_test, y_score)
print('AUC: {}'.format(auc(fpr, tpr)))
```

```
AUC: 0.7188116023655309
```

In [139]:

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

#Seaborns Beautiful Styling
sns.set_style("darkgrid", {"axes.facecolor": ".9"})

print('AUC: {}'.format(auc(fpr, tpr)))
plt.figure(figsize=(10,8))
lw = 2
plt.plot(fpr, tpr, color='darkorange',
         lw=lw, label='ROC curve')
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.yticks([i/20.0 for i in range(21)])
plt.xticks([i/20.0 for i in range(21)])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()
```

```
AUC: 0.7188116023655309
```



Decision Tree Classifier and Random Forest

In [140]:

```
from sklearn.linear_model import LogisticRegression # Regression classi
from sklearn.tree import DecisionTreeClassifier # Decision Tree classif
from sklearn import svm # Support Vector Machine
from sklearn.ensemble import RandomForestClassifier, GradientBoostingCl
from sklearn.metrics import accuracy_score, recall_score, confusion_mat
```

In [141]:

```
def ML_Pipeline(clf_name):
    clf = Classifiers[clf_name]
    fit = clf.fit(train_features,train['Deal_Status'])
    pred = clf.predict(test_features)
    Accuracy = accuracy_score(test['Deal_Status'],pred)
    Confusion_matrix = confusion_matrix(test['Deal_Status'],pred)
    print('==='*20)
    print('Accuracy = '+str(Accuracy))
    print('==='*20)
    print(Confusion_matrix)
```

In [142]:

```
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)
y_pred = dtc.predict(X_test)
Accuracy = accuracy_score(y_test, y_pred)
Confusion_matrix = confusion_matrix(y_test, y_pred)
print('==='*20)
print('Accuracy = '+str(Accuracy))
print('==='*20)
print(Confusion_matrix)
```

```
============================================================
Accuracy = 0.6916666666666667
============================================================
[[49 18]
 [19 34]]
```

In [143]:

```
RFC = RandomForestClassifier()
RFC.fit(X_train, y_train)
y_pred = RFC.predict(X_test)
Accuracy = accuracy_score(y_test, y_pred)
Confusion_matrix = confusion_matrix(y_test, y_pred)
print('==='*20)
print('Accuracy = '+str(Accuracy))
print('==='*20)
print(Confusion_matrix)
```

```
============================================================
Accuracy = 0.6833333333333333
============================================================
[[47 20]
 [18 35]]
```