Policygenius Data Assignment
By: Wajdan Mahbub
Date: 5/10/22

1. **The first step in providing recommendations is to start with a high-level analysis. *(Please present this answer visually)***
    a. **How does our conversion rate change through the funnel overall and by marketing channels?**

Image 1.1) Life Insurance Product Funnel

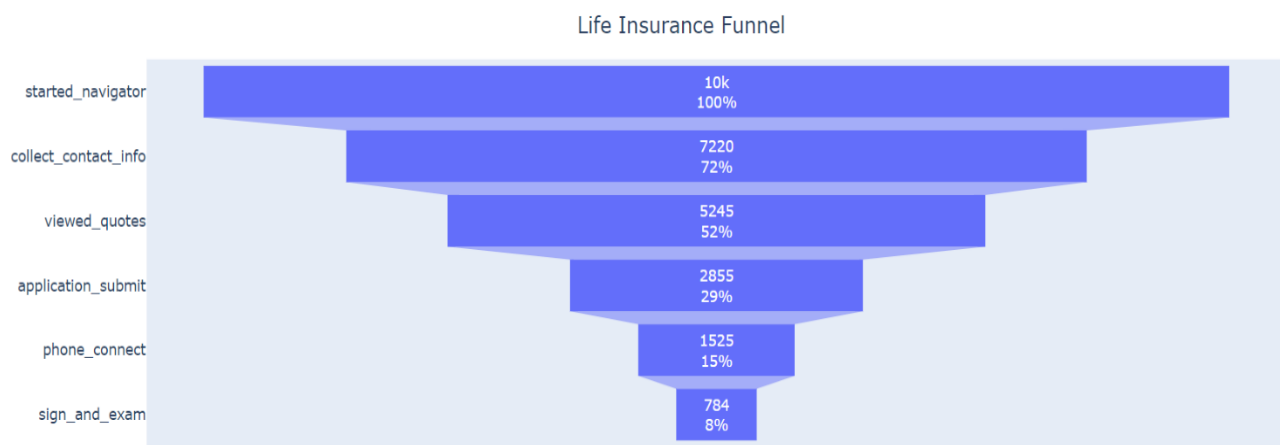

Image 1.2) Life Insurance Product Funnel by Marketing Channel

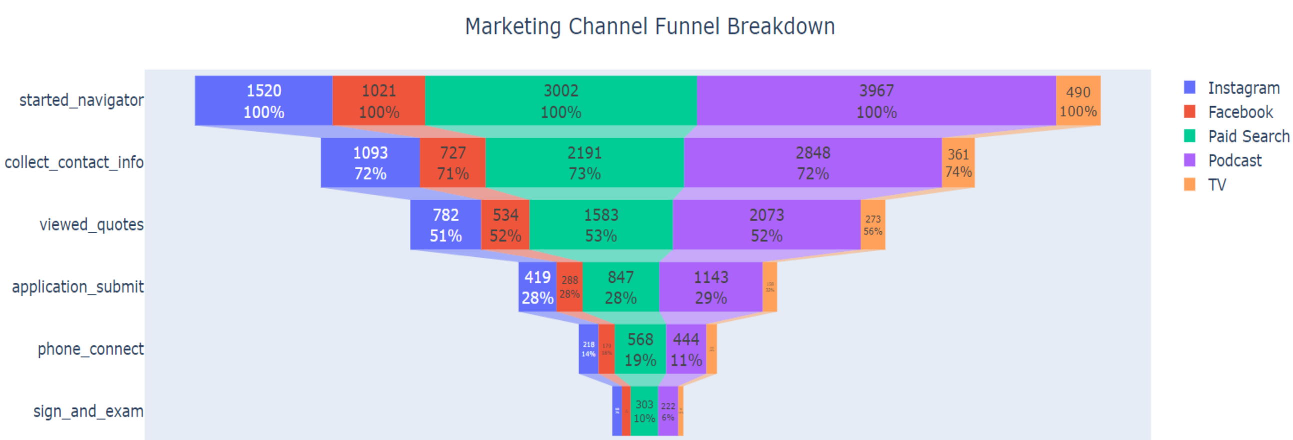

Image 1.3) Conversion Rate for Marketing Channels

	Marketing_Channel	Conversion_Rate_%
0	TV	11.43
1	Paid Search	10.09
2	Facebook	9.50
3	Overall Average	7.84
4	Instagram	6.97
5	Podcast	5.60

Overall, the conversion rate is 7.84% for the life insurance product. Significant drops in customer conversion occur from viewed quotes to application submit (46% drop), application submit to phone connect (47% drop), and phone connect to sign and exam (49% drop).

When the conversion rate is analyzed by marketing channel, TV, Paid Search, and Facebook have conversion rates of 11.43%, 10.09%, and 9.5%. These numbers are substantially above the average conversion rate of 7.84%. Instagram and Podcasts have conversion rates of 6.97% and 5.6%. Furthermore, all marketing channels experience significant drops in customer conversion at each step, from viewed quotes to sign and exam.

In the last stage of the product funnel (phone connect to sign and exam), Instagram, TV, and Podcasts experience large drops, approximately 50%, in customer conversion. This drop drags down the overall conversion rate and the conversion rates for the three marketing channels.

2. **The second step in providing recommendations is to look at conversion at a segment level through the funnel.** ***(Approach this from a model-based perspective)***
    a. **Which features are associated with people converting through the funnel?**

Image 2.1) Conversion Rates for Device Type and Marketing Channels

	Device_Type_&_Marketing_Channel	Conversion_Rate_%
0	iOS - Paid Search	19.73
1	iOS - TV	17.71
2	iOS - Facebook	17.04
3	Desktop - TV	13.41
4	iOS - Podcast	11.77
5	iOS - Instagram	11.22
6	Desktop - Paid Search	10.15
7	Desktop - Instagram	8.17
8	Overall Average	7.84
9	Desktop - Podcast	5.40
10	Android - TV	4.05
11	Android - Paid Search	3.64
12	Android - Facebook	3.22
13	Desktop - Facebook	2.54
14	Android - Instagram	2.51
15	Android - Podcast	1.73

When the conversion rate is analyzed through the lens of marketing channel and device type, it is clear that users of iOS-powered devices have above-average conversion rates. The top three are iOS users who come to Policygenius through Paid Search, TV, and Facebook. The conversion rates for these three groups are 19.73%, 17.71%, and 17.04%. Even in the two worst performing marketing channels, iOS users from Podcasts and Instagram have conversion rates of 11.77% and 11.22%, which are much higher than the overall conversion rate of 7.84%.

On the opposite end of the spectrum, users of Android-powered devices have the lowest conversion rates. Four of the bottom five conversion rates come from Android users. The two worst conversion rates belong to Android users who come to Policygenius through Podcasts (1.73%) and Instagram (2.51%).

When customer demographics for iOS, Android, and Desktop users are observed, iOS and Desktop users have higher incomes than Android users. The higher income results in a lower premium-to-income ratio for iOS and Desktop customers. The positive correlation between income and conversion rate, with respect to device type and marketing channel, is an area that needs to be further investigated.

3. What hypotheses do you take away from the funnel and segmentation analysis above?

   a. What recommendations would you make to the product team?

As Android customers from Podcasts and Instagram have the two worst conversion rates, I would work with the Product team to make adjustments that increase the conversion rate for this subset of users. I would recommend that the Product team develop a better pricing strategy and product recommendation. Pricing and product must cater to the specific income level and needs of Android users.

On a larger scale, I would work with the Product team to analyze the exchange between customers and insurance agents, and see if the step from phone connect to sign and exam can be improved. This needs to be done to prevent many Android and Desktop customers from dropping off at this stage of the process. Analyzing the conversation between iOS users and insurance agents could potentially yield insights that can be used to improve the conversion rate for Android and Desktop users.

b. How would you test your recommendations?

To test recommendations individually, we would devise an A/B test. To test multiple recommendations simultaneously, we would develop a multivariate test.

For example, a product choice A/B test would involve one set of Android users (control group) seeing the current product listings and another set of users (test group) seeing a new set of product listings that cater to their demographics. This A/B test will be applied to the Podcast and Instagram marketing channels. To avoid bias, the experiment will run for one month on randomly selected and equally balanced samples. At the end of the test, the results will be evaluated to see how changing the product selection impacted conversion rates.

c. What recommendations are likely to drive the largest impact?

Improving the conversion rate between phone connect and sign and exam for Android and Desktop users will likely have the biggest impact. However, this recommendation can potentially take longer to implement as a lot of text data needs to be analyzed between insurance agents and customers.

Improving the pricing and product recommendations for Android users coming from Instagram and Podcasts will be easier to implement and evaluate. If the findings from this recommendation are successfully applied to all Android marketing channels, then it can potentially have a more significant impact.

Image 3.1) Conversion Rate Improvement Scenarios

	Scenarios	Conversion_Rate_%
0	Android - Podcast - save 60%	11.184211
1	Android - Instagram - save 60%	10.855950
2	Android - Podcast - save 50%	6.496711
3	Android - Instagram - save 50%	6.263048
4	Android - Instagram - save 40%	3.340292
5	Android - Podcast - save 40%	3.289474
6	Android - Instagram - Default	2.505219
7	Android - Podcast - Default	1.726974

Using the scenario table from above as reference, if the targeted approach reaches a 50% (middle outcome) save rate for Android users coming in from Instagram and Podcasts, the conversion rate for the respective marketing channels jump to 6.5% and 6.3%.

4. What are the caveats of your analysis?

The recommendations are based on data that is assumed to be randomly sampled and accurate. Furthermore, the sample used in this analysis is a representative sample of the population of life insurance customers.

As no date range is given for the data, it is assumed that all the observations occurred during the same time period.

It is assumed that actual people, not bots, clicked on the ads from the various marketing channels, or that bot traffic was filtered out when preparing this dataset.

5. If you had more time and resources to develop this analysis further.

a. What additional attributes would you want to gain more insight about optimizing conversion?

I want to add in attributes such as customer gender, time and date of purchase, name of insurance provider selected, time taken to receive a call from an insurance agent, length of call with an insurance agent, actual premium rate chosen, the reason for purchase and types of insurance already purchased.

Finally, I would like to have data on the cost of ads in each marketing channel, and the revenue Poligycenius receives from each purchase.

b. What other ways could you expand this work?

I want to run a Principal Component Analysis (PCA) to remove attributes that do not impact purchases. If date and historic data were provided, I could create a machine learning model to predict future purchases. I would experiment with random forests, decision trees, and gradient boosted trees to see which model has the lowest root mean squared error and r2 value. This would be useful for making financial projections and planning capital expenditure. It would also be helpful for workforce planning, as there needs to be an adequate number of insurance agents ready to call customers.

Finally, using the cost and revenue data, I can run a cost-benefit analysis of each recommendation.

Python Script

```
# Overall Table
df2= df1.pivot_table(values= 'user_id', index = ['Order', 'funnel_steps'], aggfunc= 'count')
df3= df2.diff(periods=1, axis=0)
df4= df2.pct_change()*100

df5= df2.merge(df3, on= ['Order', 'funnel_steps'], how= 'inner').merge(df4, on= ['Order', 'funnel_steps'],
how= 'inner')
new_cols= {'user_id_x': 'Count_Users', 'user_id_y': 'Delta', 'user_id': 'Percent_Change'}
df5.rename(columns= new_cols, inplace= True)

df2_mkt= df1.pivot_table(values= 'user_id', index = ['Order', 'funnel_steps'], columns= 'marketing_chan
nels', aggfunc= 'count')
df3_mkt= df2_mkt.diff(periods=1, axis=0)
df4_mkt= df2_mkt.pct_change()*100

# Marketing Channel Table
df5_mkt= df2_mkt.merge(df3_mkt, on= ['Order', 'funnel_steps'], how= 'inner').merge(df4_mkt, on= ['Or
der', 'funnel_steps'], how= 'inner')
new_cols= {'facebook_x': 'FB_Count',
        'instagram_x': 'IG_Count',
        'paid_search_nb_x': 'PS_Count',
        'podcast_x': 'PD_Count',
        'tv_x':'TV_Count',
        'facebook_y':'FB_Delta',
        'instagram_y': 'IG_Delta',
        'paid_search_nb_y': 'PS_Delta',
        'podcast_y': 'PD_Delta',
        'tv_y': 'TV_Delta',
        'facebook': 'FB_Pct_Change',
        'instagram':'IG_Pct_Change',
        'paid_search_nb': 'PS_Pct_Change',
        'podcast': 'PD_Pct_Change',
        'tv': 'TV_Pct_Change'}

df5_mkt.rename(columns= new_cols, inplace= True)
df6= df5.reset_index(drop= False)
```

Question 1)

```
# Life Insurance Funnel
fig = go.Figure(go.Funnel(y = df6['funnel_steps'], x = df6['Count_Users'],  textposition = "inside",
   textinfo = "value + percent initial" ))

fig.update_layout(
   title={
```

```
        'text': "Life Insurance Funnel",
        'y':0.88,
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'})
fig.update_layout(font_size=14)
fig.show()

df6_mkt= df5_mkt.reset_index(drop= False)

# Life Insurance Funnel by Marketing Channel
fig = go.Figure()

fig.add_trace(go.Funnel(
    name = 'Instagram',
    y = df6_mkt['funnel_steps'],
    x = df6_mkt['IG_Count'],
    textposition= 'inside',
    textinfo = "value + percent initial"))

fig.add_trace(go.Funnel(
    name = 'Facebook',
    orientation = "h",
    y = df6_mkt['funnel_steps'],
    x = df6_mkt['FB_Count'],
    textposition = "inside",
    textinfo = "value + percent initial"))

fig.add_trace(go.Funnel(
    name = 'Paid Search',
    orientation = "h",
    y = df6_mkt['funnel_steps'],
    x = df6_mkt['PS_Count'],
    textposition = "inside",
    textinfo = "value + percent initial"))

fig.add_trace(go.Funnel(
    name = 'Podcast',
    orientation = "h",
    y = df6_mkt['funnel_steps'],
    x = df6_mkt['PD_Count'],
    textposition = "inside",
    textinfo = "value + percent initial"))

fig.add_trace(go.Funnel(
    name = 'TV',
    orientation = "h",
    y = df6_mkt['funnel_steps'],
```

```
   x = df6_mkt['TV_Count'],
   textposition = "inside",
   textinfo = "value+percent initial"))

fig.update_layout(font_size=15)

fig.update_layout(
   title={
      'text': "Marketing Channel Funnel Breakdown",
      'y':0.9,
      'x':0.5,
      'xanchor': 'center',
      'yanchor': 'top'})

fig.show()

# Marketing Channel Conversion Rates
Conversion_Rate= round(df5['Count_Users'].values[5]/ df5['Count_Users'].values[0]*100,2)
Conversion_Rate_FB= round(df6_mkt['FB_Count'].values[5]/ df6_mkt['FB_Count'].values[0]*100,2)
Conversion_Rate_IG= round(df6_mkt['IG_Count'].values[5]/ df6_mkt['IG_Count'].values[0]*100,2)
Conversion_Rate_PS= round(df6_mkt['PS_Count'].values[5]/ df6_mkt['PS_Count'].values[0]*100,2)
Conversion_Rate_PD= round(df6_mkt['PD_Count'].values[5]/ df6_mkt['PD_Count'].values[0]*100,2)
Conversion_Rate_TV= round(df6_mkt['TV_Count'].values[5]/ df6_mkt['TV_Count'].values[0]*100,2)

dict_cr = {'Marketing_Channel': ['Overall Average', 'Facebook', 'Instagram', 'Paid Search', 'Podcast', 'TV'],
      'Conversion_Rate_%': [Conversion_Rate, Conversion_Rate_FB, Conversion_Rate_IG,
Conversion_Rate_PS,Conversion_Rate_PD, Conversion_Rate_TV]}

df_cr= pd.DataFrame(dict_cr).sort_values('Conversion_Rate_%', ascending=
False).reset_index(drop=True)
```

Question 2)

```
# Conversion Rates for Device Types and Marketing Channels

# Generating numbers and variables
Conversion_Rate_Android_FB= round(df2_dev_mkt[('mobile_android', 'facebook')].values[5]/
df2_dev_mkt[('mobile_android', 'facebook')].values[0]*100,2)
Conversion_Rate_Android_IG= round(df2_dev_mkt[('mobile_android', 'instagram')].values[5]/
df2_dev_mkt[('mobile_android', 'instagram')].values[0]*100,2)
Conversion_Rate_Android_PS= round(df2_dev_mkt[('mobile_android', 'paid_search_nb')].values[5]/
df2_dev_mkt[('mobile_android', 'paid_search_nb')].values[0]*100,2)
Conversion_Rate_Android_PD= round(df2_dev_mkt[('mobile_android', 'podcast')].values[5]/
df2_dev_mkt[('mobile_android', 'podcast')].values[0]*100,2)
Conversion_Rate_Android_TV= round(df2_dev_mkt[('mobile_android', 'tv')].values[5]/
df2_dev_mkt[('mobile_android', 'tv')].values[0]*100,2)
```

```
Conversion_Rate_IOS_FB= round(df2_dev_mkt[('mobile_ios', 'facebook')].values[5]/
df2_dev_mkt[('mobile_ios', 'facebook')].values[0]*100,2)
Conversion_Rate_IOS_IG= round(df2_dev_mkt[('mobile_ios', 'instagram')].values[5]/
df2_dev_mkt[('mobile_ios', 'instagram')].values[0]*100,2)
Conversion_Rate_IOS_PS= round(df2_dev_mkt[('mobile_ios', 'paid_search_nb')].values[5]/
df2_dev_mkt[('mobile_ios', 'paid_search_nb')].values[0]*100,2)
Conversion_Rate_IOS_PD= round(df2_dev_mkt[('mobile_ios', 'podcast')].values[5]/
df2_dev_mkt[('mobile_ios', 'podcast')].values[0]*100,2)
Conversion_Rate_IOS_TV= round(df2_dev_mkt[('mobile_ios', 'tv')].values[5]/
df2_dev_mkt[('mobile_ios', 'tv')].values[0]*100,2)

Conversion_Rate_Desk_FB= round(df2_dev_mkt[('desktop', 'facebook')].values[5]/
df2_dev_mkt[('desktop', 'podcast')].values[0]*100,2)
Conversion_Rate_Desk_IG= round(df2_dev_mkt[('desktop', 'instagram')].values[5]/
df2_dev_mkt[('desktop', 'instagram')].values[0]*100,2)
Conversion_Rate_Desk_PS= round(df2_dev_mkt[('desktop', 'paid_search_nb')].values[5]/
df2_dev_mkt[('desktop', 'paid_search_nb')].values[0]*100,2)
Conversion_Rate_Desk_PD= round(df2_dev_mkt[('desktop', 'podcast')].values[5]/
df2_dev_mkt[('desktop', 'podcast')].values[0]*100,2)
Conversion_Rate_Desk_TV= round(df2_dev_mkt[('desktop', 'tv')].values[5]/ df2_dev_mkt[('desktop',
'tv')].values[0]*100,2)

# Generating final DataFrame
dict_desk_mkt_cr = {'Device_Type_&_Marketing_Channel':
                ['Overall Average', 'Android - Facebook', 'Android - Instagram', 'Android - Paid Search',
'Android - Podcast', 'Android - TV',
                 'iOS - Facebook', 'iOS - Instagram', 'iOS - Paid Search', 'iOS - Podcast', 'iOS - TV',
                 'Desktop - Facebook', 'Desktop - Instagram', 'Desktop - Paid Search', 'Desktop - Podcast',
'Desktop - TV'] ,

                'Conversion_Rate_%':
                [Conversion_Rate, Conversion_Rate_Android_FB, Conversion_Rate_Android_IG,
Conversion_Rate_Android_PS, Conversion_Rate_Android_PD, Conversion_Rate_Android_TV,
                 Conversion_Rate_IOS_FB, Conversion_Rate_IOS_IG, Conversion_Rate_IOS_PS,
Conversion_Rate_IOS_PD, Conversion_Rate_IOS_TV,
                 Conversion_Rate_Desk_FB, Conversion_Rate_Desk_IG, Conversion_Rate_Desk_PS,
Conversion_Rate_Desk_PD, Conversion_Rate_Desk_TV]}

df_desk_mkt_cr= pd.DataFrame(dict_desk_mkt_cr).sort_values('Conversion_Rate_%', ascending=
False).reset_index(drop=True)

df_desk_mkt_cr
```

Question 3)

```
# Scenario Analysis DataFrame

# Generating the variables and numbers
```

```
micro_df= df2_dev_mkt.iloc[: ,[6,8]]

a=df2_dev_mkt[('mobile_android', 'instagram')].values[0]
b=df2_dev_mkt[('mobile_android', 'instagram')].values[1]
c=df2_dev_mkt[('mobile_android', 'instagram')].values[2]

e=df2_dev_mkt[('mobile_android', 'podcast')].values[0]
f=df2_dev_mkt[('mobile_android', 'podcast')].values[1]
g=df2_dev_mkt[('mobile_android', 'podcast')].values[2]

micro_df[('mobile_android', 'instagram_save_60%')] = [a,b,c, round(c*0.6,0) , round(c*0.6**2,0),
round(c*0.6**3,0)]
micro_df[('mobile_android', 'podcast_save_60%')] = [e,f,g, round(g*0.6,0) , round(g*0.6**2,0),
round(g*0.6**3,0)]

micro_df[('mobile_android', 'instagram_save_50%')] =[a,b,c,round(c*0.5,0) , round(c*0.5**2,0),
round(c*0.5**3,0)]
micro_df[('mobile_android', 'podcast_save_50%')] =[e,f,g, round(g*0.5,0) , round(g*0.5**2,0),
round(g*0.5**3,0)]

micro_df[('mobile_android', 'instagram_save_40%')] = [a,b,c, round(c*0.4,0) , round(c*0.4**2,0),
round(c*0.4**3,0)]
micro_df[('mobile_android', 'podcast_save_40%')] = [e,f,g, round(g*0.4,0) , round(g*0.4**2,0),
round(g*0.4**3,0)]

# Generating final DataFrame
ig_den= micro_df.iloc[0,0]
pd_den= micro_df.iloc[0,1]
default_ig= micro_df.iloc[5,0]/ig_den*100
default_pd= micro_df.iloc[5,1]/pd_den*100
sixty_ig= micro_df.iloc[5,2]/ig_den*100
sixty_pd= micro_df.iloc[5,3]/pd_den*100
fifty_ig= micro_df.iloc[5,4]/ig_den*100
fifty_pd= micro_df.iloc[5,5]/pd_den*100
forty_ig= micro_df.iloc[5,6]/ig_den*100
forty_pd= micro_df.iloc[5,7]/pd_den*100

micro_dict = {'Scenarios':
        ['Android - Instagram - Default', 'Android - Podcast - Default',
         'Android - Instagram - save 60%', 'Android - Podcast - save 60%',
         'Android - Instagram - save 50%', 'Android - Podcast - save 50%',
         'Android - Instagram - save 40%', 'Android - Podcast - save 40%'],
        'Conversion_Rate_%':
        [default_ig, default_pd, sixty_ig, sixty_pd, fifty_ig, fifty_pd, forty_ig, forty_pd]}

df_scenario = pd.DataFrame(micro_dict).sort_values('Conversion_Rate_%', ascending=
False).reset_index(drop=True)
```