Client Report - [Project 5: The war with Star Wars]

Course CSE 250 James Lule

Elevator pitch

paste your elevator pitch here

GRAND QUESTION 1

Shorten the column names and clean them up for easier use with pandas.

##I shortened the column names *type your results and analysis here*

TECHNICAL DETAILS

```
#paste your table code in this snippet box```

q_1 = (sw_col.iloc[0, :]
    .replace("Have you seen any of the 6 films in the Star Wars franchise?",
'seen_any')
    .replace("Do you consider yourself to be a fan of the Star Wars film
franchise?", 'fan_sw')
    .replace("Which of the following Star Wars films have you seen? Please select
all that apply.", 'seen')
    .replace("Please rank the Star Wars films in order of preference with 1 being
your favorite film in the franchise and 6 being your least favorite film.",
'film_rank')
    .replace('Please state whether you view the following characters favorably,
unfavorably, or are unfamiliar with him/her.', 'character_view_')
    .replace('Which character shot first?', 'shot_first')
    .replace('Are you familiar with the Expanded Universe?',
'familiar_w_expanded_universe')
    .str.replace("Do you consider yourself to be a fan of the Expanded Universe?",
"fan_expanded_universe")
    .str.replace("Œ", "")
    .str.replace('æ', '')
    .str.replace('Œ', '')
    .str.replace('?', '')
    .str.replace('(', '')
    .str.replace(')', '')
    .str.replace('Do you consider yourself to be a fan of the Star Trek
franchise?', 'fan_star_trek')
    .str.lower()
    .str.replace(" ", "_")
    .ffill() # forward fill take seen variable and copy it down. fills in empty
columns with the one above it.

)
```

```
q_2 = (sw_col.iloc[1,:]
    .replace('Response', '')
    .str.replace('Star Wars: Episode', '')
    .str.lower()
    .str.replace(' ', '_')
    .fillna('')
)

column_names = q_1 + q_2

column_names

# adding to the data that we skipped the first two rows.
sw_data.columns = column_names

sw_data

print(sw_data.head(2).to_markdown())
```

replace the table below with your table

0 respondentid 1 seen_any 2 fan_sw 3 seen_i__the_phantom_menace 4 seen_ii__attack_of_the_clones 5 seen_iii__revenge_of_the_sith 6 seen_iv__a_new_hope 7 seen_v_the_empire_strikes_back 8 seen_vi_return_of_the_jedi 9 film_rank_i__the_phantom_menace 10 film_rank_ii__attack_of_the_clones 11 film_rank_iii__revenge_of_the_sith 12 film_rank_iv__a_new_hope 13 film_rank_v_the_empire_strikes_back 14 film_rank_vi_return_of_the_jedi 15 character_view_han_solo 16 character_view_luke_skywalker 17 character_view_princess_leia_organa 18 character_view_anakin_skywalker 19 character_view_obi_wan_kenobi 20 character_view_emperor_palpatine 21 character_view_darth_vader 22 character_view_lando_calrissian 23 character_view_boba_fett 24 character_view_c-3p0 25 character_view_r2_d2 26 character_view_jar_jar_binks 27 character_view_padme_amidala 28 character_view_yoda 29 shot_first 30 familiar_w_expanded_universe 31 fan_expanded_universe□ 32 fan_star_trek 33 gender 34 age 35 household_income 36 education 37 location_census_region dtype: object

GRAND QUESTION 2

Filter the dataset to those that have seen at least one film.

type your results and analysis here

TECHNICAL DETAILS

```
#paste code here
```

insert your chart png here

```
#paste your table code in this snippet box
```

GRAND QUESTION 3

COPY PASTE GRAND QUESTION 3Â FROM THE PROJECT HERE

type your results and analysis here

TECHNICAL DETAILS

```
#paste chart code in this snippet box
```

insert your chart png here

```
#paste your table code in this snippet box
```

replace the table below with your table

	animal
0	elk
1	pig
2	dog
3	quetzal

GRAND QUESTION 4

COPY PASTE GRAND QUESTION 4 FROM THE PROJECT HERE

type your results and analysis here

TECHNICAL DETAILS

```
#paste chart code in this snippet box
```

insert your chart png here

```
#paste your table code in this snippet box
```

replace the table below with your table

	animal
0	elk
1	pig
2	dog
3	quetzal

GRAND QUESTION 5

COPY PASTE GRAND QUESTION 5 FROM THE PROJECT HERE

type your results and analysis here

TECHNICAL DETAILS

```
#paste chart code in this snippet box
```

insert your chart png here

```
#paste your table code in this snippet box
```

replace the table below with your table

	animal
0	elk
1	pig
2	dog
3	quetzal

APPENDIX A (PYTHON CODE)

```
#%%
import pandas as pd
import altair as alt
import numpy as np
# from altair_saver import save
#%%
url = 'https://raw.githubusercontent.com/fivethirtyeight/data/master/star-wars-survey/StarWars.csv'
```

```
data = pd.read_csv(url, encoding='ISO-8859-1')

sw_col = pd.read_csv(url, encoding='ISO-8859-1', header=None, nrows=2)
sw_data = pd.read_csv(url, encoding='ISO-8859-1', header=None, skiprows=2)

#%%

# Question 1
q_1 = (sw_col.iloc[0, :]
    .replace("Have you seen any of the 6 films in the Star Wars franchise?", 'seen_any')
    .replace("Do you consider yourself to be a fan of the Star Wars film franchise?", 'fan_sw')
    .replace("Which of the following Star Wars films have you seen? Please select all that apply.", 'seen')
    .replace("Please rank the Star Wars films in order of preference with 1 being your favorite film in the franchise and 6 being your least favorite film.", 'film_rank')
    .replace('Please state whether you view the following characters favorably, unfavorably, or are unfamiliar with him/her.', 'character_view_')
    .replace('Which character shot first?', 'shot_first')
    .replace('Are you familiar with the Expanded Universe?', 'familiar_w_expanded_universe')
    .str.replace("Do you consider yourself to be a fan of the Expanded Universe?", "fan_expanded_universe")
    .str.replace("Œ", "")
    .str.replace('æ', '')
    .str.replace('Œ', '')
    .str.replace('?', '')
    .str.replace('(', '')
    .str.replace(')', '')
    .str.replace('Do you consider yourself to be a fan of the Star Trek franchise?', 'fan_star_trek')
    .str.lower()
    .str.replace(" ", "_")
    .ffill() # forward fill take seen variable and copy it down. fills in empty columns with the one above it.

)

q_2 = (sw_col.iloc[1,:]
    .replace('Response', '')
    .str.replace('Star Wars: Episode', '')
    .str.lower()
    .str.replace(' ', '_')
    .fillna('')
)

column_names = q_1 + q_2

column_names
```

```
# adding to the data that we skipped the first two rows.
sw_data.columns = column_names

sw_data

print(sw_data.head(2).to_markdown())

#%%
# Question 2

# Favorability chart
alt.data_transformers.disable_max_rows()

favorability_data = (sw_data
#.query('seen_any == "Yes"')
.filter(regex = 'character_view_*')
.dropna()
.replace(to_replace=['Very favorably', 'Somewhat favorably'], value="Favorable")
.replace(to_replace=['Somewhat unfavorably', 'Very unfavorably'],
value='Unfavorable')
.replace(to_replace=['Unfamiliar (N/A)'], value='Unfamiliar')
.replace(to_replace='Neither favorably nor unfavorably (neutral)',
value='Neutral')
.melt(var_name = 'character' , value_name = 'view')
# .str(['character', 'view'])
.replace(to_replace = '^character_view_*', value = '', regex = True)
.groupby('character')['view'] # ?
.value_counts(normalize=True) #
.reset_index(name='percent')
)

alt.themes.enable('fivethirtyeight')

test = (alt.Chart(favorability_data)
        .encode(
        alt.X('percent', title = '', axis = None)
        , alt.Y('character',title = '', sort= '-x')
        # , alt.Color(color = 'view')
        , column = 'view'
        )
        .mark_bar()
        # .facet(columns='view', title='view')
        .properties(title={'text': "`Star Wars` Character Favorability Rating",
'subtitle': 'By 834 respondents'})
        .configure_title(anchor='start')
)

test

# who shot first chart
response = (pd.DataFrame(
    sw_data.shot_first.value_counts(normalize=True)
    .round(2) * 100)
    .reset_index().rename(columns={'index': 'response'})
```

```
)

order = ['Han', 'Greedo', "I don't understand this question"]


chart = (alt.Chart(response)
    .encode(x = alt.X('shot_first', title= '', axis = None)
        , y = alt.Y('response', title = '', sort = order)
        , text = 'shot_first')
    .mark_bar()
)

alt.themes.enable('fivethirtyeight')

text = (alt.Chart(response)
    .mark_text(
        align='left'
        , baseline='middle'
        , dy=-3)
    .encode( alt.X('shot_first', title = '', axis = None)
        , alt.Y('response', title = '', sort = order)
        , text = 'shot_first')
)

final_chart = (chart + text).properties(title={'text': 'Who Shot First?', 'subtitle': 'According to 843 respondents'}).configure_title(anchor='start')

final_chart

sw_data.seen_any.isna()

# normalize = true will give us a precent a group is in a column
(sw_data.gender.value_counts(normalize=True))

# table 1
print(sw_data.groupby('gender')
    .seen_any
    .value_counts(normalize=True)
    .round(4)
    .to_markdown())

# table 2
print(sw_data.query('seen_any == "Yes"')
.groupby('gender')
.fan_sw
.value_counts(normalize = True)

)

#%%
# Question 3
# Clean and format the data
sw_ml = (sw_data
    .drop(columns = ['respondentid']) # dropping since it is useless
    .query("seen_any == 'Yes'") # seen at least one film
    .drop(columns = ['seen_any'])
    .replace('18-29', 18) # converting age to a single number
    .replace('30-44', 30)
    .replace('45-60', 45)
    .replace('> 60', 61)
    .replace('Less than high school degree', 9) # convert education to a single
number
    .replace('High school degree', 12)
    .replace('Some college or Associate degree', 14)
    .replace('Bachelor degree', 16)
    .replace('Graduate degree', 20))

print(sw_ml.to_markdown())
#%%

# Question 4
# Loading packages for machine learning
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics

## splitting the data
x_train, x_test, y_train, y_test = train_test_split(
    features
    , target
    , test_size=.3
    , random_state=76
)

# classify the model
classifier_DT = DecisionTreeClassifier(max_depth = 3)

# train the model x_train and y_train
classifier_DT.fit(x_train, y_train)

# make predictions x_test
ml_predictions = classifier_DT.predict(x_test)

# test the model y_train
metrics.accuracy_score(y_test, ml_predictions)

feature_importance = pd.DataFrame(
    {'Features':features.columns,
'importance':classifier_DT.feature_importances_.round(4)}
)

# Table
feature_importance.sort_values('importance', ascending=False)
print(feature_importance.sort_values(
    'importance', ascending=False).to_markdown())

# Confusion matrix
confusion_metric = metrics.plot_confusion_matrix(classifier_DT, x_test, y_test)
```