Client Report - Project 5: The war with Star Wars

See code

Course DS 250

AUTHOR
Bracken Sant

Elevator pitch

Using the Star Wars survey data I combed through it and tidied the data to better go through a machine learning algorithm so that we could see if we can predict an interviewing job candidate's current income based on a few of their responses.

▼ Read and format project data

```
data =
        pd.DataFrame(pd.read_csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/
        star-wars-survey/StarWars.csv", encoding_errors="ignore", sep = None,engine='python'))

star_wars_data = data.copy()
star_wars_data = star_wars_data.drop(0)
star_wars_data = star_wars_data.dropna(how='all')
```

GRAND QUESTION 1

Shorten the column names and clean them up for easier use with pandas. Provide a table or list that exemplifies how you fixed the names.

I was able to shorten the names and and make them easier to use with pandas. I also did some integer encoding while I was shortening the names.

▼ Read and format data

```
eps = {'Star Wars: Episode I  The Phantom Menace': 1, np.nan : 0, 'Star Wars: Episode II
        Attack of the Clones' : 1, 'Star Wars: Episode III  Revenge of the Sith' : 1, 'Star
        Wars: Episode IV  A New Hope' : 1, 'Star Wars: Episode V The Empire Strikes Back' : 1,
        'Star Wars: Episode VI Return of the Jedi' : 1}
star_wars_data["The Phantom Menace"] = star_wars_data["Which of the following Star Wars films
        have you seen? Please select all that apply."].map(eps)
star_wars_data["Attack of the Clones"] = star_wars_data["Unnamed: 4"].map(eps)
star_wars_data["Revenge of the Sith"] = star_wars_data["Unnamed: 5"].map(eps)
star_wars_data["A New Hope"] = star_wars_data["Unnamed: 6"].map(eps)
star_wars_data["The Empire Strikes Back"] = star_wars_data["Unnamed: 7"].map(eps)
```

```
star_wars_data["Return of the Jedi"] = star_wars_data["Unnamed: 8"].map(eps)
yes_no = {'Yes' : 1, 'No' : 0, np.nan : 0}
numbers = {"1":"1", '2':'2', '3':'3', '4':'4', '5':'5', '6':'6', np.nan:'0'}
star_wars_data["Seen_Any"] = star_wars_data["Have you seen any of the 6 films in the Star Wars
        franchise?"].map(yes_no)
star_wars_data["Fan?"] = star_wars_data["Do you consider yourself to be a fan of the Star Wars
        film franchise?"].map(yes_no)
star_wars_data["Ep1_Ranked"] = star_wars_data['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.'].map(numbers)
star_wars_data["Ep2_Ranked"] = star_wars_data["Unnamed: 10"].map(numbers)
star_wars_data["Ep3_Ranked"] = star_wars_data["Unnamed: 11"].map(numbers)
star_wars_data["Ep4_Ranked"] = star_wars_data["Unnamed: 12"].map(numbers)
star_wars_data["Ep5_Ranked"] = star_wars_data["Unnamed: 13"].map(numbers)
star_wars_data["Ep6_Ranked"] = star_wars_data["Unnamed: 14"].map(numbers)
favorable = {"Very favorably":5, "Somewhat favorably":4, "Unfamiliar (N/A)":6, "Neither
        favorably nor unfavorably (neutral)":3, "Very unfavorably":1, "Somewhat
        unfavorably":2, np.nan:0}
star_wars_data["Han_favorable?"] = star_wars_data["Please state whether you view the following
        characters favorably, unfavorably, or are unfamiliar with him/her."].map(favorable)
star_wars_data["Luke_favorable?"] = star_wars_data["Unnamed: 16"].map(favorable)
star_wars_data["Leia_favorable?"] = star_wars_data["Unnamed: 17"].map(favorable)
star_wars_data["Anakin_favorable?"] = star_wars_data["Unnamed: 18"].map(favorable)
star_wars_data["Obi_Wan_favorable?"] = star_wars_data["Unnamed: 19"].map(favorable)
star_wars_data["Palpatine_favorable?"] = star_wars_data["Unnamed: 20"].map(favorable)
star_wars_data["Vader_favorable?"] = star_wars_data["Unnamed: 21"].map(favorable)
star_wars_data["Lando_favorable?"] = star_wars_data["Unnamed: 22"].map(favorable)
star_wars_data["Boba_favorable?"] = star_wars_data["Unnamed: 23"].map(favorable)
star_wars_data["C-3PO_favorable?"] = star_wars_data["Unnamed: 24"].map(favorable)
star_wars_data["R2_D2_favorable?"] = star_wars_data["Unnamed: 25"].map(favorable)
star_wars_data["Jar_Jar_favorable?"] = star_wars_data["Unnamed: 26"].map(favorable)
star_wars_data["Padme_favorable?"] = star_wars_data["Unnamed: 27"].map(favorable)
star_wars_data["Yoda_favorable?"] = star_wars_data["Unnamed: 28"].map(favorable)
shot_first = {"I don't understand this question":1, np.nan:0, "Greedo":2, "Han":3}
star_wars_data["Shot_First"] = star_wars_data["Which character shot first?"].map(shot_first)
star_wars_data["Familiar_Expanded"] = star_wars_data["Are you familiar with the Expanded
        Universe?"].map(yes_no)
star_wars_data["Fan_of_Expanded"] = star_wars_data["Do you consider yourself to be a fan of the
        Expanded Universe?"].map(yes_no)
star_wars_data["Fan_of_Star_Trek"] = star_wars_data["Do you consider yourself to be a fan of
        the Star Trek franchise?"].map(yes_no)
print(tabulate(star_wars_data[["The Phantom Menace", "Attack of the Clones", "Revenge of the
        Sith", "A New Hope"]].head(), headers='keys', tablefmt='fancy_grid'))
```

	The Phantom Menace	Attack of the Clones	Revenge of the Sith	A New Hope
1	1	1	1	1
2	0	0	0	0
3	1	1	1	0

4	1	1	1	1
5	1	1	1	1

As you can see on the table the names have been shortened.

GRAND QUESTION 2

Clean and format the data so that it can be used in a machine learning model. As you format the data, you should complete each item listed below. In your final report provide example(s) of the reformatted data with a short description of the changes made.

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

b. Create a new column that converts the age ranges to a single number. Drop the age range categorical column.

c. Create a new column that converts the education groupings to a single number. Drop the school categorical column

d. Create a new column that converts the income ranges to a single number. Drop the income range categorical column.

e. Create your target (also known as “y” or “label”) column based on the new income range column.

f. One-hot encode all remaining categorical columns.

▼ Read and format data

```
# Part a
star_wars_data = star_wars_data.query("Seen_Any == 1")
```

▼ Read and format data

```
# Part b
age = {"18-29":18, "30-44":30, ">60":61, "45-60":45, np.nan:0}
star_wars_data["Age"] = star_wars_data["Age"].map(age)
```

I decided to choose the bottom number since the last number was a top inclusive.

▼ Read and format data

```
# Part c
education = {"High school degree":1, "Bachelor degree": 3, "Some college or Associate degree":
         2, np.nan:0, "Graduate degree":4}
star_wars_data["Education"] = star_wars_data["Education"].map(education)
```

I decided to go on a scale starting from 1 and going forward from that based on amount of education until 4.

▼ Read and format data

```
# Part d
income = {np.nan:-1, "$0 - $24,999":0, "$100,000 - $149,999":100000, "$25,000 - $49,999":25000,
         "$50,000 - $99,999":50000, "$150,000+":150000}
star_wars_data["Income"] = star_wars_data["Household Income"].map(income)
```

I decided to choose the bottom number since the last number was a top inclusive, like with the age.

▼ Read and format data

```
# Part e
star_wars_data["target"] = star_wars_data["Income"] >= 50000
```

I decided to include 50,000 since I made the income the bottom number so that it would include all of the $50,000 - $99,999 range.

▼ Read and format data

```
# Part f
regions = {"South Atlantic":4, "West South Central":8, "Middle Atlantic":5, "East North
         Central":9, "Pacific":1, "Mountain":2, "New England":3, "West North Central":6, "East
         South Central":7, np.nan:0}
star_wars_data["Census_Region"] = star_wars_data["Location (Census Region)"].map(regions)
gender = {"Male":1, "Female":2, np.nan:0}
star_wars_data["Gender"] = star_wars_data["Gender"].map(gender)
```

The other columns from the first question I also did integer encoding with.

GRAND QUESTION 3

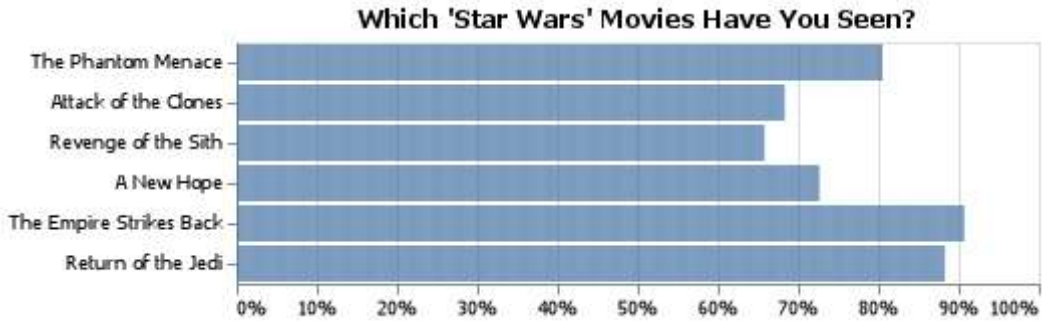
Validate that the data provided on GitHub lines up with the article by recreating 2 of the visuals from the article.

I was able to recreate 2 of the visuals from the article so the data seems to line up with the article.

▼ Chart 1

```
chart1 = al.Chart(star_wars_data).properties(title="Which 'Star Wars' Movies Have You
         Seen?").transform_fold(
    ["The Phantom Menace", "Attack of the Clones", "Revenge of the Sith", "A New Hope", "The
         Empire Strikes Back", "Return of the Jedi"],
    as_=["key", "value"]
).transform_calculate(
    PercentageOfTotal = "datum.value / 836"
).mark_bar().encode(
    al.X('PercentageOfTotal:Q', axis=al.Axis(format='.0%'), title=""),
```

```
    al.X('PercentageOfTotal:Q', axis=al.Axis(format='.0%'), title=''),
    al.Y('key:N', title="", sort=["The Phantom Menace", "Attack of the Clones", "Revenge of the
        Sith", "A New Hope", "The Empire Strikes Back", "Return of the Jedi"]),
)
```

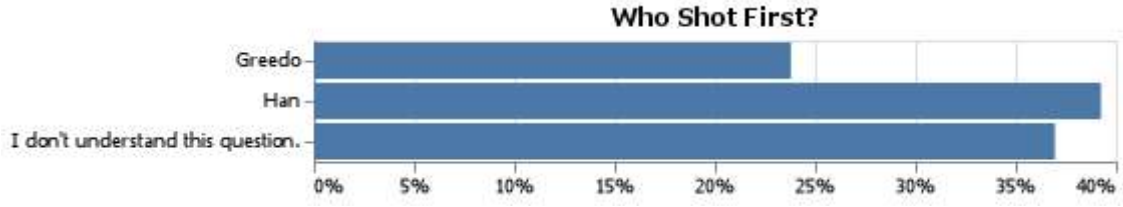


"Which 'Star Wars' Movies Have You Seen?"

This is the "Which 'Star Wars' Movies Have You Seen?" chart and if you compare the two you can see the semblance.

▼ Chart 2

```
other_data1 = star_wars_data.query("Shot_First == 1")
other_data2 = star_wars_data.query("Shot_First == 2")
other_data3 = star_wars_data.query("Shot_First == 3")
count1 = other_data1["Shot_First"].count()
count2 = other_data2["Shot_First"].count()
count3 = other_data3["Shot_First"].count()
the_data = {"Names":["I don't understand this question.", "Greedo", "Han"], "Count":[count1,
        count2, count3]}
who_shot_first = pd.DataFrame(data=the_data)
chart2 = al.Chart(who_shot_first).properties(title="Who Shot First?").transform_joinaggregate(
    Total_Count = "sum(Count)",
).transform_calculate(
    PercentOfTotal = "datum.Count / datum.Total_Count"
).mark_bar().encode(
    al.X("PercentOfTotal:Q", axis=al.Axis(format='.0%'), title=""),
    al.Y('Names:N', title="")
)
```



"Who Shot First?"

This is the "Who Shot First?" chart and if you compare the two you can see the semblance.

GRAND QUESTION 4

GRAND QUESTION 1

Build a machine learning model that predicts whether a person makes more than $50k. Describe your model and report the accuracy.

I wasn't able to raise the accuracy above 70%, so I do not believe that it would be thebest option to use this data to predict the interviewing job candidate's current income.

▼ Machine Learning Algorithm

```
X = star_wars_data[["Census_Region", "Age", "Education", "Seen_Any", "Fan?",
         "Fan_of_Star_Trek", "Fan_of_Expanded"]]

y = star_wars_data['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3, random_state=24,
         stratify=y)

rf = RandomForestClassifier(random_state=24)
```

Accuracy: 0.676

Precision: 0.644

Recall: 0.659

As you can see from the accuracy, precision, and recall reports the data does not predict the target that accurately or at least not accurate enough for it to be a viable source.