Project 02 Written Work

● Graded

Group

Scott Kim
Sangwon Ji

Total Points

12 / 12 pts

Question 1

Q1.2.2 **2** / 2 pts

✔ **+ 1 pt** Correlation between words greater than 0.2 or smaller than -0.2

✔ **+ 1 pt** Correct graph (correct r, slope, and intercept equations)

+ 0 pts Incorrect/Blank

Question 2

Q1.3.1 **2** / 2 pts

✔ **+ 2 pts** Correct graph (vertical bar chart is fine!)

+ 0 pts Incorrect/Blank

Question 3

Q3.1.7 **2** / 2 pts

✔ **+ 2 pts** Any logical explanation. i.e. The staff features don't work very well. It's a good idea to pick words that are common in thriller movies but uncommon in comedy movies and vice-versa. These are points that are far away from the diagonal in the above plot.

+ 0 pts Incorrect/Blank

Question 4

Q3.3.3 **2** / 2 pts

✔ **+ 2 pts** Correct answer: The classifier tends to misclassify movies that have both comedy and thriller elements in them (or any logical explanation similar to this)

+ 0 pts Incorrect/Blank

Question 5

Q4.2 **2** / 2 pts

✔ **+ 2 pts** Any reasonable answer that relates to the student's classifier.

+ 0 pts Incorrect/Blank

Question 6

Q4.3 **2** / 2 pts

✔ **+ 2 pts** Answer that shows any amount of effort.

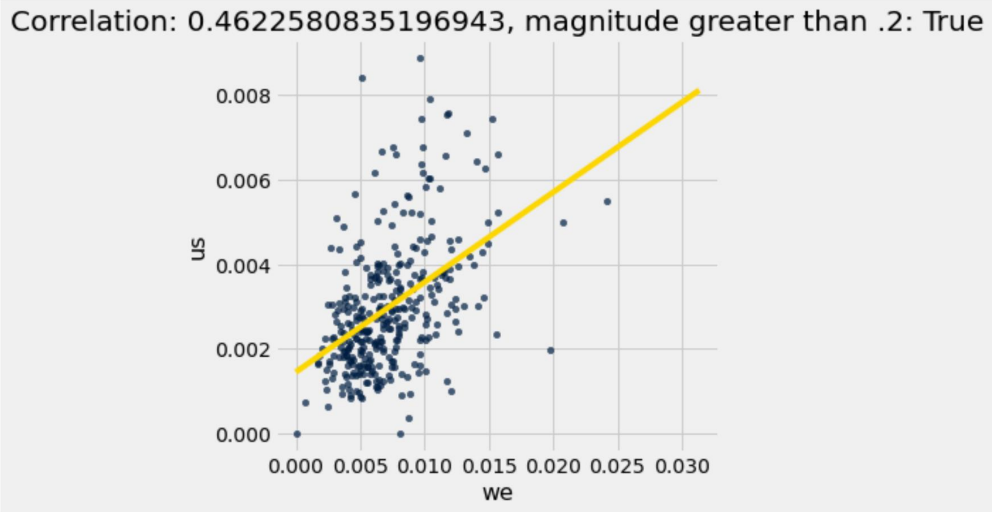
+ 0 pts Incorrect/Blank

Question 1.2.2 Choose two *different* words in the dataset with a magnitude (absolute value) of correlation higher than 0.2 and plot a scatter plot with a line of best fit for them. Please do not pick "outer" and "space" or "san" and "francisco". The code to plot the scatter plot and line of best fit is given for you, you just need to calculate the correct values to `r`, `slope` and `intercept`.

Hint 1: It's easier to think of words with a positive correlation, i.e. words that are often mentioned together*. Try to think of common phrases or idioms.

Hint 2: Refer to Section 15.2 of the textbook for the formulas.

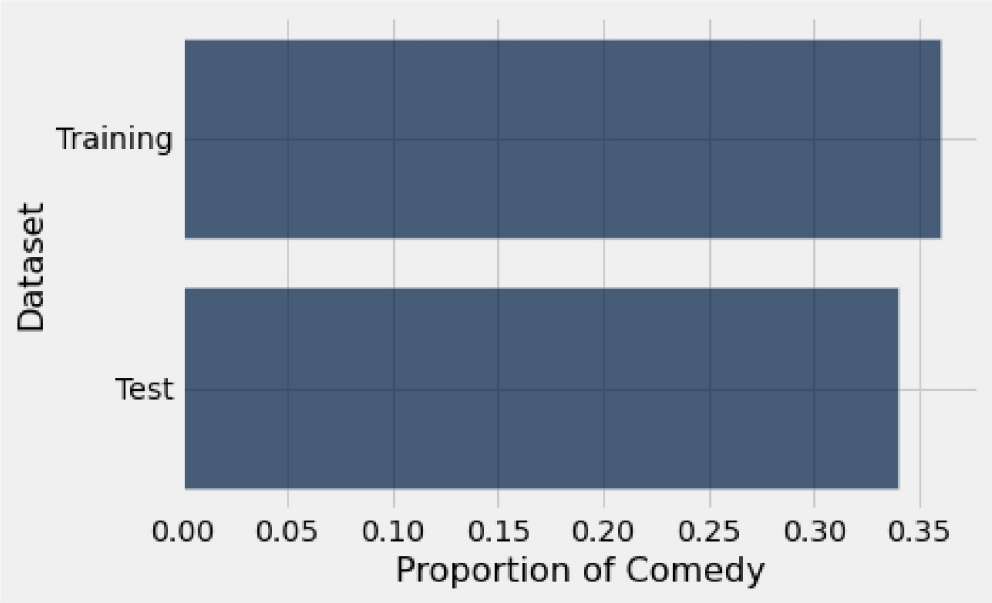
In [22]:
```
word_x = "we"
word_y = "us"

# These arrays should make your code cleaner!
arr_x = movies.column(word_x)
arr_y = movies.column(word_y)

x_su = (arr_x - np.mean(arr_x)) / np.std(arr_x)
y_su = (arr_y - np.mean(arr_y)) / np.std(arr_y)

r = np.mean(x_su * y_su)

slope = r * (np.std(arr_y) / np.std(arr_x))
intercept = slope * (-np.mean(arr_x)) + np.mean(arr_y)

# DON'T CHANGE THESE LINES OF CODE
movies.scatter(word_x, word_y)
max_x = max(movies.column(word_x))
plots.title(f"Correlation: {r}, magnitude greater than .2: {abs(r) >= 0.2}")
plots.plot([0, max_x * 1.3], [intercept, intercept + slope * (max_x*1.3)], color='gold');
```



Question 1.3.1 Draw a horizontal bar chart with two bars that show the proportion of Comedy movies in each dataset (`train_movies` and `test_movies`). The two bars should be labeled "Training" and "Test". Complete the function `comedy_proportion` first; it should help you create the bar chart.

Hint: Refer to Section 7.1 of the textbook if you need a refresher on bar charts.

```
In [27]: def comedy_proportion(table):
             # Return the proportion of movies in a table that have the comedy genre.
             return table.where("Genre", are.equal_to("comedy")).num_rows/table.num_rows

         # The staff solution took multiple lines.  Start by creating a table.
         # If you get stuck, think about what sort of table you need for barh to work

         Table().with_columns("Dataset", make_array("Training", "Test"), "Proportion of Comedy", make_a
```



Question 3.1.7 In two sentences or less, describe how you selected your features.

I looked at the scatter plot at the beginning of 3.1 and chose words that were further away from the diagonal line. I chose 5 words that were far from the line as possible and had a high occurance in comedy movies, and then I chose the other 5 that were also far from the line but had a high occurance in thriller movies.

Question 3.3.3

Do you see a pattern in the types of movies your classifier misclassifies? In two sentences or less, describe any patterns you see in the results or any other interesting findings from the table above. If you need some help, try looking up the movies that your classifier got wrong on Wikipedia.

Looking at the list of movies that my classifier misclassified, many of them are multi-genre meaning they contain elements not from a single genre but from variety of genres. For example, "Spiderman" is certainly an action/thriller movie, but it does have some comedic vibe to it. "Bettle Juice" is another example of a multi-genre movie where its genre is supposed to be fantasy/horror/comedy.

Question 4.2

Do you see a pattern in the mistakes your new classifier makes? How good an accuracy were you able to get with your limited classifier? Did you notice an improvement from your first classifier to the second one? Describe in two sentences or less.

Hint: You may not be able to see a pattern.

In terms of accuracy, it has dropped from which was initially 0.8 to 0.5, as using less words seems to be the reason. There doesn't seem to be a pattern to where the classifiers went wrong, however, it still seems that classifier misclassifies movies that has diverse genres features in one.

Question 4.3

Given the constraint of five words, how did you select those five? Describe in two sentences or less.

For the Five words, as I did it in my previous my features, I tried to select words 2 from the comedy and 2 from the thriller, and since the number is odd I have selected one word that seems to be even as possible.