Detect data bias with Amazon SageMaker Clarify

Introduction

Bias can be present in your data before any model training occurs. Inspecting the dataset for bias can help detect collection gaps, inform your feature engineering, and understand societal biases the dataset may reflect. In this lab you will analyze bias on the dataset, generate and analyze bias report, and prepare the dataset for the model training.

Table of Contents

First, let's install and import required modules.

In [2]:
```
# please ignore warning messages during the installation
!pip install --disable-pip-version-check -q sagemaker==2.35.0
```

```
WARNING: Running pip as the 'root' user can result in broken permissions
and conflicting behaviour with the system package manager. It is recommen
ded to use a virtual environment instead: https://pip.pypa.io/warnings/ve
nv
```

In [3]:
```
import boto3
import sagemaker
import pandas as pd
import numpy as np
import botocore

config = botocore.config.Config(user_agent_extra='dlai-pds/c1/w2')

# low-level service client of the boto3 session
sm = boto3.client(service_name='sagemaker',
                  config=config)

sess = sagemaker.Session(sagemaker_client=sm)

bucket = sess.default_bucket()
role = sagemaker.get_execution_role()
region = sess.boto_region_name
```

In [4]:
```
import matplotlib.pyplot as plt
%matplotlib inline
%config InlineBackend.figure_format='retina'
```

1. Analyze the dataset

1.1. Create a pandas data frame from the CSV file

Create a pandas dataframe from each of the product categories and concatenate them into one.

In [5]:
```
!aws s3 cp 's3://dlai-practical-data-science/data/transformed/womens_clot
```

```
download: s3://dlai-practical-data-science/data/transformed/womens_clothi
ng_ecommerce_reviews_transformed.csv to ./womens_clothing_ecommerce_revie
ws_transformed.csv
```

In [6]:
```
path = './womens_clothing_ecommerce_reviews_transformed.csv'

df = pd.read_csv(path)
df.head()
```

Out[6]:

	sentiment	review_body	product_category
0	1	If this product was in petite i would get the...	Blouses
1	1	Love this dress! it's sooo pretty. i happene...	Dresses
2	0	I had such high hopes for this dress and reall...	Dresses
3	1	I love love love this jumpsuit. it's fun fl...	Pants
4	1	This shirt is very flattering to all due to th...	Blouses

As you saw in the previous lab, there are way more positive reviews than negative or neutral. Such a dataset is called unbalanced.

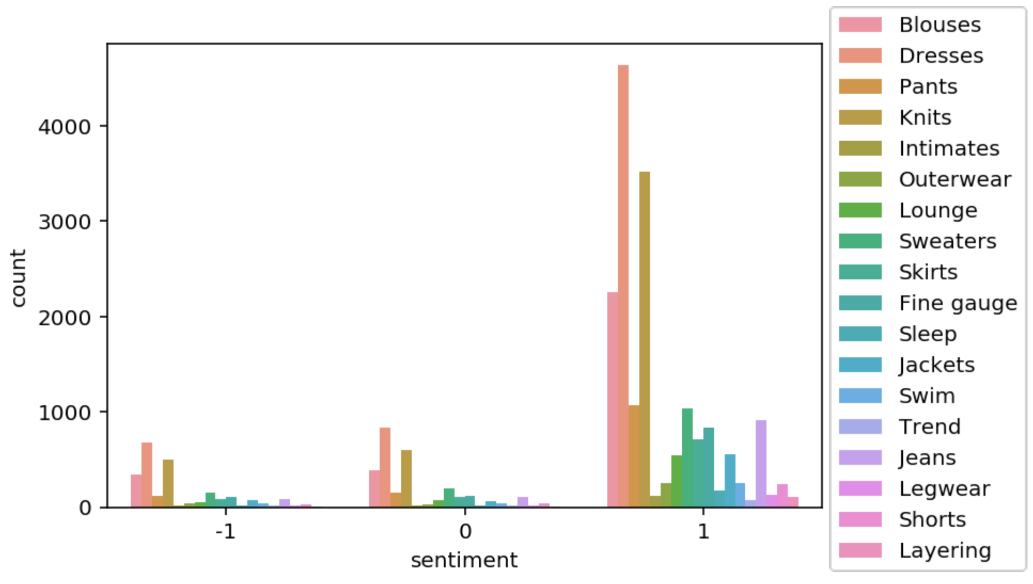
In this case, using a relatively small data subset you could visualize the occurring unbalances. At scale, you would need to perform bias analysis. Let's use this dataset as an example.

In [7]:
```
import seaborn as sns

sns.countplot(data=df, x='sentiment', hue='product_category')

plt.legend(loc='upper right',bbox_to_anchor=(1.3, 1.1))
```

Out[7]:
```
<matplotlib.legend.Legend at 0x7fc9b93cfc50>
```



1.2. Upload the dataset to S3 bucket

Upload the dataset to a private S3 bucket in a folder called `bias/unbalanced`.

In [8]:
```
data_s3_uri_unbalanced = sess.upload_data(bucket=bucket,
                                          key_prefix='bias/unbalanced',
                                          path='./womens_clothing_ecommerce_reviews_
data_s3_uri_unbalanced
```

Out[8]:
```
's3://sagemaker-us-east-1-449615851382/bias/unbalanced/womens_clothing_ec
ommerce_reviews_transformed.csv'
```

You can review the uploaded CSV file in the S3 bucket.

Instructions:

- open the link
- click on the S3 bucket name `sagemaker-us-east-1-ACCOUNT`
- go to the folder `bias/unbalanced`
- check the existence of the file `womens_clothing_ecommerce_reviews_transformed.csv`

In [9]:
```
from IPython.core.display import display, HTML

display(HTML('<b>Review <a target="top" href="https://s3.console.aws.amaz
```

Review Amazon S3 bucket

2. Analyze class imbalance on the dataset with Amazon SageMaker Clarify

Let's analyze bias in `sentiment` with respect to the `product_category` facet on the dataset.

2.1. Configure a `DataConfig`

Information about the input data needs to be provided to the processor. This can be done with the `DataConfig` of the Clarify container. It stores information about the dataset to be analyzed, for example the dataset file, its format, headers and labels.

Exercise 1

Configure a `DataConfig` for Clarify.

Instructions: Use `DataConfig` to configure the target column (`'sentiment'` label), data input (`data_s3_uri_unbalanced`) and output paths (`bias_report_unbalanced_output_path`) with their formats (header names and the dataset type):

```
data_config_unbalanced = clarify.DataConfig(
    s3_data_input_path=..., # S3 object path containing the
unbalanced dataset
    s3_output_path=..., # path to store the output
    label='...', # target column
    headers=df_unbalanced.columns.to_list(),
    dataset_type='text/csv'
)
```

In [10]:
```
from sagemaker import clarify

bias_report_unbalanced_output_path = 's3://{}/bias/generated_bias_report/

data_config_unbalanced = clarify.DataConfig(
    ### BEGIN SOLUTION - DO NOT delete this comment for grading purposes
    s3_data_input_path=data_s3_uri_unbalanced, # Replace None
    s3_output_path=bias_report_unbalanced_output_path, # Replace None
    label='sentiment', # Replace None
    ### END SOLUTION - DO NOT delete this comment for grading purposes
    headers=df.columns.to_list(),
    dataset_type='text/csv'
)
```

2.2. Configure `BiasConfig`

Bias is measured by calculating a metric and comparing it across groups. To compute it, you will specify the required information in the `BiasConfig` API. SageMaker Clarify needs the sensitive columns (`facet_name`) and the desirable outcomes (`label_values_or_threshold`). Here `product_category` is the sensitive facet and the desired outcome is with the `sentiment==1`.

SageMaker Clarify can handle both categorical and continuous data for `label_values_or_threshold`. In this case you are using categorical data.

In [11]:
```
bias_config_unbalanced = clarify.BiasConfig(
    label_values_or_threshold=[1], # desired sentiment
    facet_name='product_category' # sensitive column (facet)
)
```

2.3. Configure Amazon SageMaker Clarify as a processing job

Now you need to construct an object called `SageMakerClarifyProcessor`. This allows you to scale the process of data bias detection using two parameters, `instance_count` and `instance_type`. `Instance_count` represents how many nodes you want in the distributor cluster during the data detection. `Instance_type` specifies the processing capability (compute capacity, memory capacity) available for each one of those nodes. For the purposes of this lab, you will use a relatively small instance type. Please refer to this link for additional instance types that may work for your use case outside of this lab.

In [12]:
```
clarify_processor_unbalanced = clarify.SageMakerClarifyProcessor(role=rol
                                                                  instance_count=1,
                                                                  instance_type='ml.m
                                                                  sagemaker_session=s
```

2.4. Run the Amazon SageMaker Clarify processing job

Exercise 2

Run the configured processing job to compute the requested bias `methods` of the input data

Instructions: Apply the `run_pre_training_bias` method to the configured Clarify processor, passing the configured input/output data (`data_config_unbalanced`), configuration of sensitive groups (`bias_config_unbalanced`) with the other job setup parameters:

```
clarify_processor_unbalanced.run_pre_training_bias(
    data_config=..., # configured input/output data
    data_bias_config=..., # configured sensitive groups
    methods=["CI", "DPL", "KL", "JS", "LP", "TVD", "KS"], #
selector of a subset of potential metrics
    wait=False, # whether the call should wait until the job
completes (default: True)
    logs=False # whether to show the logs produced by the job.
Only meaningful when wait is True (default: True)
)
```

In [13]:
```
clarify_processor_unbalanced.run_pre_training_bias(
    ### BEGIN SOLUTION - DO NOT delete this comment for grading purposes
    data_config=data_config_unbalanced, # Replace None
    data_bias_config=bias_config_unbalanced, # Replace None
    ### END SOLUTION - DO NOT delete this comment for grading purposes
    methods=["CI", "DPL", "KL", "JS", "LP", "TVD", "KS"],
    wait=False,
    logs=False
)
```

```
Job Name:  Clarify-Pretraining-Bias-2022-08-22-12-51-10-265
Inputs:  [{'InputName': 'dataset', 'AppManaged': False, 'S3Input': {'S3Ur
i': 's3://sagemaker-us-east-1-449615851382/bias/unbalanced/womens_clothin
g_ecommerce_reviews_transformed.csv', 'LocalPath': '/opt/ml/processing/in
put/data', 'S3DataType': 'S3Prefix', 'S3InputMode': 'File', 'S3DataDistri
butionType': 'FullyReplicated', 'S3CompressionType': 'None'}}, {'InputNam
e': 'analysis_config', 'AppManaged': False, 'S3Input': {'S3Uri': 's3://sa
gemaker-us-east-1-449615851382/bias/generated_bias_report/unbalanced/anal
ysis_config.json', 'LocalPath': '/opt/ml/processing/input/config', 'S3Dat
aType': 'S3Prefix', 'S3InputMode': 'File', 'S3DataDistributionType': 'Ful
lyReplicated', 'S3CompressionType': 'None'}}]
Outputs:  [{'OutputName': 'analysis_result', 'AppManaged': False, 'S3Outp
ut': {'S3Uri': 's3://sagemaker-us-east-1-449615851382/bias/generated_bias
_report/unbalanced', 'LocalPath': '/opt/ml/processing/output', 'S3UploadM
ode': 'EndOfJob'}}]
```

In [14]:
```
run_unbalanced_bias_processing_job_name = clarify_processor_unbalanced.la
print(run_unbalanced_bias_processing_job_name)
```

```
Clarify-Pretraining-Bias-2022-08-22-12-51-10-265
```

2.5. Run and review the Amazon SageMaker Clarify processing job on the unbalanced dataset

Review the created Amazon SageMaker Clarify processing job and the Cloud Watch logs.

Instructions:

- open the link
- note that you are in the section Amazon SageMaker -> Processing jobs
- check the processing job name
- note which other properties of the processing job you can see in the console

In [15]:
```
from IPython.core.display import display, HTML

display(HTML('<b>Review <a target="blank" href="https://console.aws.amazo
```

Review processing job

Instructions:

- open the link
- open the log stream with the name, which starts from the processing job name
- have a quick look at the log messages

In [16]:
```
from IPython.core.display import display, HTML

display(HTML('<b>Review <a target="blank" href="https://console.aws.amazo
```

Review CloudWatch logs after about 5 minutes

In [17]:
```
running_processor = sagemaker.processing.ProcessingJob.from_processing_na
```

This cell will take approximately 5-10 minutes to run.

In [18]:
```
%%time

running_processor.wait(logs=False)
```

```
.................................................................!CPU time
s: user 244 ms, sys: 53.7 ms, total: 298 ms
Wall time: 5min 27s
```

2.6. Analyze unbalanced bias report

In this run, you analyzed bias for `sentiment` relative to the `product_category` for the unbalanced data. Let's have a look at the bias report.

List the files in the output path `bias_report_unbalanced_output_path`:

In [19]:
```
!aws s3 ls $bias_report_unbalanced_output_path/
```

```
2022-08-22 12:57:24      31732 analysis.json
2022-08-22 12:51:11        346 analysis_config.json
2022-08-22 12:57:24     390923 report.html
2022-08-22 12:57:24     131783 report.ipynb
2022-08-22 12:57:24     150178 report.pdf
```

Download generated bias report from S3 bucket:

In [20]:
```
!aws s3 cp --recursive $bias_report_unbalanced_output_path ./generated_bi
```

```
download: s3://sagemaker-us-east-1-449615851382/bias/generated_bias_repor
t/unbalanced/analysis_config.json to generated_bias_report/unbalanced/ana
lysis_config.json
download: s3://sagemaker-us-east-1-449615851382/bias/generated_bias_repor
t/unbalanced/analysis.json to generated_bias_report/unbalanced/analysis.j
son
download: s3://sagemaker-us-east-1-449615851382/bias/generated_bias_repor
t/unbalanced/report.html to generated_bias_report/unbalanced/report.html
download: s3://sagemaker-us-east-1-449615851382/bias/generated_bias_repor
t/unbalanced/report.pdf to generated_bias_report/unbalanced/report.pdf
download: s3://sagemaker-us-east-1-449615851382/bias/generated_bias_repor
t/unbalanced/report.ipynb to generated_bias_report/unbalanced/report.ipyn
b
```

Review the downloaded bias report (in HTML format):

In [21]:
```
from IPython.core.display import display, HTML

display(HTML('<b>Review <a target="blank" href="./generated_bias_report/u
```

Review unbalanced bias report

The bias report shows a number of metrics, but here you can focus on just two of them:

- Class Imbalance (CI). Measures the imbalance in the number of members between different facet values. Answers the question, does a `product_category` have disproportionately more reviews than others? Values of CI will become equal for even distribution between facets. Here, different CI values show the existence of imbalance.
- Difference in Positive Proportions in Labels (DPL). Measures the imbalance of positive outcomes between different facet values. Answers the question, does a `product_category` have disproportionately higher ratings than others? With the range over the interval from -1 to 1, if there is no bias, you want to see this value as close as possible to zero. Here, non-zero values indicate the imbalances.

3. Balance the dataset by `product_category` and `sentiment`

Let's balance the dataset by `product_category` and `sentiment`. Then you can configure and run SageMaker Clarify processing job to analyze the bias of it. Which metrics values do you expect to see in the bias report?

In [22]:
```
df_grouped_by = df.groupby(['product_category', 'sentiment'])
df_balanced = df_grouped_by.apply(lambda x: x.sample(df_grouped_by.size()
```

In [23]:
```
df_balanced
```

Out[23]:

			sentiment	review_body	product_category
product_category	**sentiment**				
Blouses	**-1**	**0**	-1	I loved the star pattern of this top but the c...	Blouses
		1	-1	The first time i wore it felt wonderful. had ...	Blouses
		2	-1	The print is very pretty but the fabric is sti...	Blouses
		3	-1	The shape is awkward too loose not flatterin...	Blouses
		4	-1	Cute top but started disintegrating after two ...	Blouses
...	**...**	**...**	...	...	...
Trend	**1**	**4**	1	Medium is equivalent to eu size 40. the pants ...	Trend
		5	1	I love his cardigan! i knew the small would be...	Trend
		6	1	I love the style and look oft this blouse but ...	Trend
		7	1	This easel caftan is simply amazing! the silho...	Trend
		8	1	I was on the fence about ordering this jacket ...	Trend

486 rows × 3 columns

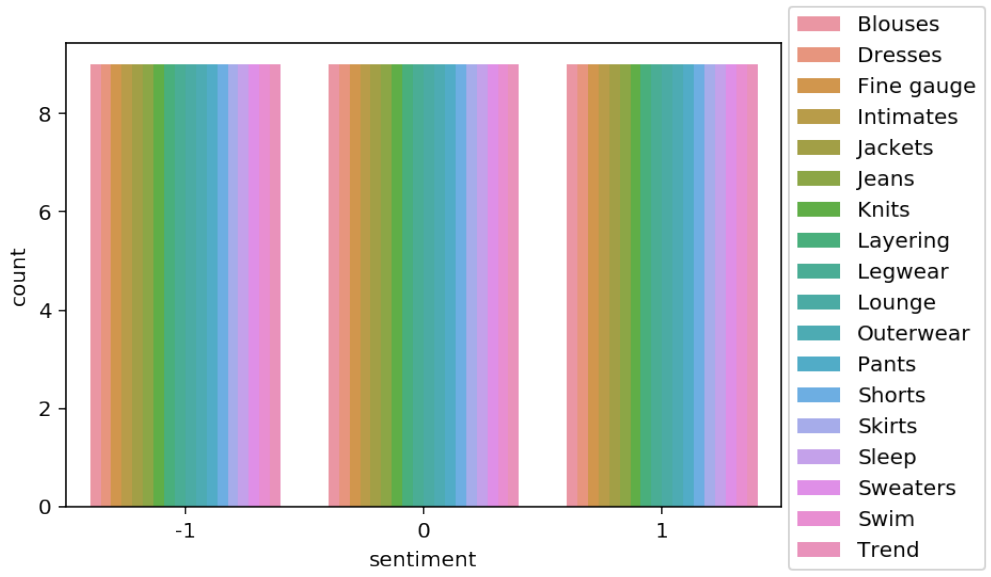
Visualize the distribution of review sentiment in the balanced dataset.

In [24]:
```
import seaborn as sns

sns.countplot(data=df_balanced, x='sentiment', hue='product_category')

plt.legend(loc='upper right',bbox_to_anchor=(1.3, 1.1))
```

Out[24]:
```
<matplotlib.legend.Legend at 0x7fc9b279e850>
```



4. Analyze bias on balanced dataset with Amazon SageMaker Clarify

Let's analyze bias in `sentiment` with respect to the `product_category` facet on your balanced dataset.

Save and upload balanced data to S3 bucket.

In [25]:
```
path_balanced = './womens_clothing_ecommerce_reviews_balanced.csv'
df_balanced.to_csv(path_balanced, index=False, header=True)

data_s3_uri_balanced = sess.upload_data(bucket=bucket, key_prefix='bias/b
data_s3_uri_balanced
```

Out[25]:
```
's3://sagemaker-us-east-1-449615851382/bias/balanced/womens_clothing_ecom
merce_reviews_balanced.csv'
```

You can review the uploaded CSV file in the S3 bucket and prefix `bias/balanced`.

In [26]:
```
from IPython.core.display import display, HTML

display(HTML('<b>Review <a target="top" href="https://s3.console.aws.amaz
```

Review Amazon S3 bucket

4.1. Configure a `DataConfig`

Exercise 3

Configure a `DataConfig` for Clarify to analyze bias on the balanced dataset.

Instructions: Pass the S3 object path containing the balanced dataset, the path to store the output (`bias_report_balanced_output_path`) and the target column. You can use exercise 1 as an example.

In [27]:
```
from sagemaker import clarify

bias_report_balanced_output_path = 's3://{}/bias/generated_bias_report/ba

data_config_balanced = clarify.DataConfig(
    ### BEGIN SOLUTION - DO NOT delete this comment for grading purposes
    s3_data_input_path=data_s3_uri_balanced, # Replace None
    s3_output_path=bias_report_balanced_output_path, # Replace None
    label='sentiment', # Replace None
    ### END SOLUTION - DO NOT delete this comment for grading purposes
    headers=df_balanced.columns.to_list(),
    dataset_type='text/csv'
)
```

4.2. Configure `BiasConfig`

`BiasConfig` for the balanced dataset will have the same settings as before.

In [28]:
```
bias_config_balanced = clarify.BiasConfig(
    label_values_or_threshold=[1], # desired sentiment
    facet_name='product_category' # sensitive column (facet)
)
```

4.3. Configure SageMaker Clarify as a processing job

`SageMakerClarifyProcessor` object will also have the same parameters.

In [29]:
```
clarify_processor_balanced = clarify.SageMakerClarifyProcessor(role=role,
                                                               instance_count=1,
                                                               instance_type='ml.m
                                                               sagemaker_session=s
```

4.4. Run the Amazon SageMaker Clarify processing job

Exercise 4

Run the configured processing job for the balanced dataset.

Instructions: Apply the `run_pre_training_bias` method to the configured Clarify processor, passing the input/output data, configuration of sensitive groups with the other job setup parameters. You can use exercise 2 as an example.

In [30]:
```
clarify_processor_balanced.run_pre_training_bias(
    ### BEGIN SOLUTION - DO NOT delete this comment for grading purposes
    data_config=data_config_balanced, # Replace None
    data_bias_config=bias_config_balanced, # Replace None
    ### END SOLUTION - DO NOT delete this comment for grading purposes
    methods=["CI", "DPL", "KL", "JS", "LP", "TVD", "KS"],
    wait=False,
    logs=False
)
```

```
Job Name:  Clarify-Pretraining-Bias-2022-08-22-13-01-08-932
Inputs:  [{'InputName': 'dataset', 'AppManaged': False, 'S3Input': {'S3Ur
i': 's3://sagemaker-us-east-1-449615851382/bias/balanced/womens_clothing_
ecommerce_reviews_balanced.csv', 'LocalPath': '/opt/ml/processing/input/d
ata', 'S3DataType': 'S3Prefix', 'S3InputMode': 'File', 'S3DataDistributio
nType': 'FullyReplicated', 'S3CompressionType': 'None'}}, {'InputName': '
analysis_config', 'AppManaged': False, 'S3Input': {'S3Uri': 's3://sagemak
er-us-east-1-449615851382/bias/generated_bias_report/balanced/analysis_co
nfig.json', 'LocalPath': '/opt/ml/processing/input/config', 'S3DataType':
'S3Prefix', 'S3InputMode': 'File', 'S3DataDistributionType': 'FullyReplic
ated', 'S3CompressionType': 'None'}}]
Outputs:  [{'OutputName': 'analysis_result', 'AppManaged': False, 'S3Outp
ut': {'S3Uri': 's3://sagemaker-us-east-1-449615851382/bias/generated_bias
_report/balanced', 'LocalPath': '/opt/ml/processing/output', 'S3UploadMod
e': 'EndOfJob'}}]
```

In [31]:
```
run_balanced_bias_processing_job_name = clarify_processor_balanced.latest
print(run_balanced_bias_processing_job_name)
```

```
Clarify-Pretraining-Bias-2022-08-22-13-01-08-932
```

4.5. Run and review the Clarify processing job on the balanced dataset

Review the results of the run following the links:

In [32]:
```
from IPython.core.display import display, HTML

display(HTML('<b>Review <a target="blank" href="https://console.aws.amazo
```

Review processing job

In [33]:
```
from IPython.core.display import display, HTML

display(HTML('<b>Review <a target="blank" href="https://console.aws.amazo
```

Review CloudWatch logs after about 5 minutes

```
In [34]: running_processor = sagemaker.processing.ProcessingJob.from_processing_na
```

This cell will take approximately 5-10 minutes to run.

```
In [35]: %%time

         running_processor.wait(logs=False)
```

```
..........................................................................
!CPU times: user 292 ms, sys: 47.8 ms, total: 339 ms
Wall time: 6min 12s
```

4.6. Analyze balanced bias report

List the files in the output path `bias_report_balanced_output_path`:

```
In [36]: !aws s3 ls $bias_report_balanced_output_path/
```

```
2022-08-22 13:07:32      29889 analysis.json
2022-08-22 13:01:09        346 analysis_config.json
2022-08-22 13:07:32     398677 report.html
2022-08-22 13:07:32     139537 report.ipynb
2022-08-22 13:07:32     152258 report.pdf
```

Download generated bias report from S3 bucket:

```
In [37]: !aws s3 cp --recursive $bias_report_balanced_output_path ./generated_bias
```

```
download: s3://sagemaker-us-east-1-449615851382/bias/generated_bias_repor
t/balanced/analysis_config.json to generated_bias_report/balanced/analysi
s_config.json
download: s3://sagemaker-us-east-1-449615851382/bias/generated_bias_repor
t/balanced/analysis.json to generated_bias_report/balanced/analysis.json
download: s3://sagemaker-us-east-1-449615851382/bias/generated_bias_repor
t/balanced/report.html to generated_bias_report/balanced/report.html
download: s3://sagemaker-us-east-1-449615851382/bias/generated_bias_repor
t/balanced/report.ipynb to generated_bias_report/balanced/report.ipynb
download: s3://sagemaker-us-east-1-449615851382/bias/generated_bias_repor
t/balanced/report.pdf to generated_bias_report/balanced/report.pdf
```

Review the downloaded bias report (in HTML format):

```
In [38]: from IPython.core.display import display, HTML

         display(HTML('<b>Review <a target="blank" href="./generated_bias_report/b
```

Review balanced bias report

In this run, you analyzed bias for `sentiment` relative to the `product_category` for the balanced data. Note that the Class Imbalance (CI) metric is equal across all product categories for the target label, `sentiment`. And Difference in Positive Proportions in Labels (DPL) metric values are zero.

Upload the notebook into S3 bucket for grading purposes.

Note: you may need to click on "Save" button before the upload.

```
In [39]: !aws s3 cp ./C1_W2_Assignment.ipynb s3://$bucket/C1_W2_Assignment_Learner
```

```
upload: ./C1_W2_Assignment.ipynb to s3://sagemaker-us-east-1-449615851382
/C1_W2_Assignment_Learner.ipynb
```

Please go to the main lab window and click on `Submit` button (see the `Finish the lab` section of the instructions).

```
In [ ]:
```