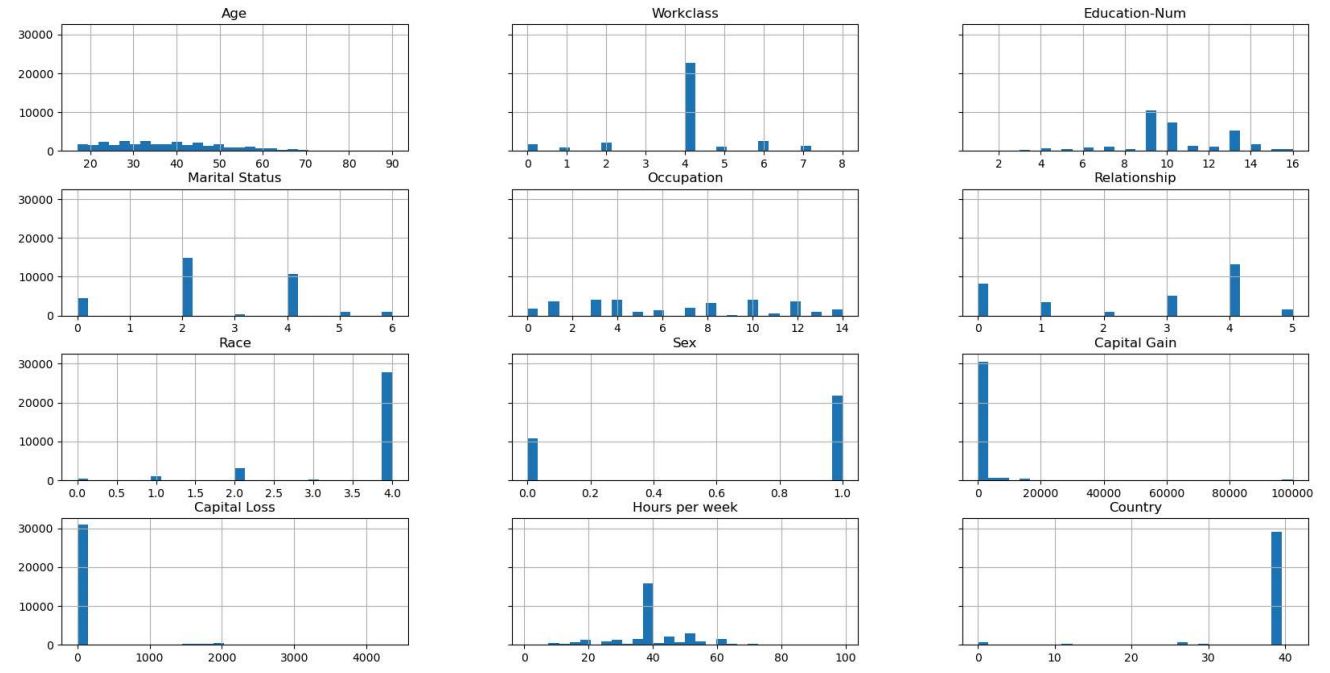
In [1]:
```
import shap
X, y = shap.datasets.adult()
X_display, y_display = shap.datasets.adult(display=True)
feature_names = list(X.columns)
feature_names
```

Out[1]:
```
['Age',
 'Workclass',
 'Education-Num',
 'Marital Status',
 'Occupation',
 'Relationship',
 'Race',
 'Sex',
 'Capital Gain',
 'Capital Loss',
 'Hours per week',
 'Country']
```

In [2]:
```
display(X.describe())
hist = X.hist(bins=30, sharey=True, figsize=(20, 10))
```

	Age	Workclass	Education-Num	Marital Status	Occupation	Relationship	Race	Sex	Capital Gain	Capital Loss	Ho
count	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561
mean	38.581646	3.868892	10.080679	2.611836	6.572740	2.494518	3.665858	0.669205	1077.648804	87.303833	40
std	13.640442	1.455960	2.572562	1.506222	4.228857	1.758232	0.848806	0.470506	7385.911621	403.014771	12
min	17.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1
25%	28.000000	4.000000	9.000000	2.000000	3.000000	0.000000	4.000000	0.000000	0.000000	0.000000	40
50%	37.000000	4.000000	10.000000	2.000000	7.000000	3.000000	4.000000	1.000000	0.000000	0.000000	40
75%	48.000000	4.000000	12.000000	4.000000	10.000000	4.000000	4.000000	1.000000	0.000000	0.000000	45
max	90.000000	8.000000	16.000000	6.000000	14.000000	5.000000	4.000000	1.000000	99999.000000	4356.000000	99



In [3]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)
X_train_display = X_display.loc[X_train.index]
```

In [6]:
```
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.25, random_state=1)
X_train_display = X_display.loc[X_train.index]
X_val_display = X_display.loc[X_val.index]
```

In [7]:
```
import pandas as pd
train = pd.concat([pd.Series(y_train, index=X_train.index,
                             name='Income>50K', dtype=int), X_train], axis=1)
validation = pd.concat([pd.Series(y_val, index=X_val.index,
                             name='Income>50K', dtype=int), X_val], axis=1)
test = pd.concat([pd.Series(y_test, index=X_test.index,
                             name='Income>50K', dtype=int), X_test], axis=1)
```

In [8]:
```
train
```

Out[8]:

	Income>50K	Age	Workclass	Education-Num	Marital Status	Occupation	Relationship	Race	Sex	Capital Gain	Capital Loss	Hours per week	Country
13825	0	54.0	6	6.0	2	3	4	4	1	0.0	0.0	36.0	39
2843	1	41.0	2	10.0	2	8	4	4	1	0.0	1485.0	40.0	39
3112	0	24.0	4	9.0	4	1	3	4	1	0.0	0.0	40.0	39
10886	0	33.0	4	12.0	0	7	0	4	0	0.0	0.0	42.0	39
12148	1	33.0	4	9.0	2	1	5	4	0	0.0	1887.0	20.0	39
...	...	...	...	...	...	...	...	...	...	...	...	...	...
245	0	56.0	4	9.0	2	1	4	4	1	0.0	0.0	35.0	0
10156	0	28.0	4	9.0	4	6	3	4	1	0.0	0.0	40.0	39
21991	0	35.0	4	9.0	2	6	4	4	1	0.0	0.0	40.0	26
342	1	36.0	7	9.0	2	11	4	4	1	7298.0	0.0	40.0	39
25283	1	56.0	4	10.0	2	4	4	4	1	0.0	0.0	40.0	39

14652 rows × 13 columns

In [9]:
```
validation
```

Out[9]:

	Income>50K	Age	Workclass	Education-Num	Marital Status	Occupation	Relationship	Race	Sex	Capital Gain	Capital Loss	Hours per week	Country
22308	0	24.0	4	10.0	4	6	3	4	1	0.0	0.0	40.0	39
8499	0	66.0	0	10.0	2	0	4	4	1	0.0	0.0	40.0	39
27309	0	38.0	4	8.0	4	7	0	2	0	0.0	0.0	50.0	39
18937	0	21.0	4	8.0	4	6	3	4	1	0.0	0.0	32.0	39
30262	0	30.0	4	10.0	4	4	0	4	1	0.0	0.0	52.0	39
...	...	...	...	...	...	...	...	...	...	...	...	...	...
21639	0	33.0	4	11.0	2	1	5	2	0	0.0	0.0	40.0	39
28968	0	29.0	4	4.0	0	3	1	4	0	0.0	0.0	55.0	39
21714	0	28.0	4	5.0	4	8	2	4	1	0.0	0.0	52.0	39
12412	1	39.0	2	8.0	2	14	4	4	1	0.0	1848.0	40.0	27
11419	1	39.0	7	13.0	2	10	4	4	1	0.0	0.0	45.0	39

4884 rows × 13 columns

In [10]:
```
test
```

Out[10]:

	Income>50K	Age	Workclass	Education-Num	Marital Status	Occupation	Relationship	Race	Sex	Capital Gain	Capital Loss	Hours per week	Country
9646	0	62.0	6	4.0	6	8	0	4	0	0.0	0.0	66.0	39
709	0	18.0	4	7.0	4	8	2	4	1	0.0	0.0	25.0	39
7385	1	25.0	4	13.0	4	5	3	4	1	27828.0	0.0	50.0	39
16671	0	33.0	4	9.0	2	10	4	4	1	0.0	0.0	40.0	39
21932	0	36.0	4	7.0	4	7	1	4	0	0.0	0.0	40.0	39
...	...	...	...	...	...	...	...	...	...	...	...	...	...
5889	1	39.0	4	13.0	2	10	5	4	0	0.0	0.0	20.0	39
25723	0	17.0	4	6.0	4	12	3	4	0	0.0	0.0	20.0	39
29514	0	35.0	4	9.0	4	14	3	4	1	0.0	0.0	40.0	39
1600	0	30.0	4	7.0	2	3	4	4	1	0.0	0.0	45.0	39
639	1	52.0	6	16.0	2	10	4	4	1	0.0	0.0	60.0	39

6513 rows × 13 columns

In [11]:
```
# Use 'csv' format to store the data
# The first column is expected to be the output column
train.to_csv('train.csv', index=False, header=False)
validation.to_csv('validation.csv', index=False, header=False)
```

In [12]:
```
import sagemaker, boto3, os
bucket = sagemaker.Session().default_bucket()
prefix = "demo-sagemaker-xgboost-adult-income-prediction"

boto3.Session().resource('s3').Bucket(bucket).Object(
    os.path.join(prefix, 'data/train.csv')).upload_file('train.csv')
boto3.Session().resource('s3').Bucket(bucket).Object(
    os.path.join(prefix, 'data/validation.csv')).upload_file('validation.csv')
```

```
sagemaker.config INFO - Not applying SDK defaults from location: /etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /home/ec2-user/.config/sagemaker/config.yaml
```

In [13]:
```
! aws s3 ls {bucket}/{prefix}/data --recursive
```

```
2024-04-18 02:48:00     589692 demo-sagemaker-xgboost-adult-income-prediction/data/train.csv
2024-04-18 02:48:00     196593 demo-sagemaker-xgboost-adult-income-prediction/data/validation.csv
```

In [14]:
```
import sagemaker

region = sagemaker.Session().boto_region_name
print("AWS Region: {}".format(region))

role = sagemaker.get_execution_role()
print("RoleArn: {}".format(role))
```

```
AWS Region: ap-south-1
RoleArn: arn:aws:iam::992382676037:role/service-role/AmazonSageMakerServiceCatalogProductsUseRole
```

In [15]:
```
from sagemaker.debugger import Rule, ProfilerRule, rule_configs
from sagemaker.session import TrainingInput

s3_output_location='s3://{}/{}/{}'.format(bucket, prefix, 'xgboost_model')

container=sagemaker.image_uris.retrieve("xgboost", region, "1.2-1")
print(container)

xgb_model=sagemaker.estimator.Estimator(
    image_uri=container,
    role=role,
    instance_count=1,
    instance_type='ml.m4.xlarge',
    volume_size=5,
    output_path=s3_output_location,
    sagemaker_session=sagemaker.Session(),
    rules=[
        Rule.sagemaker(rule_configs.create_xgboost_report()),
        ProfilerRule.sagemaker(rule_configs.ProfilerReport())
    ]
)
```

```
720646828776.dkr.ecr.ap-south-1.amazonaws.com/sagemaker-xgboost:1.2-1
```

In [16]:
```
xgb_model.set_hyperparameters(
    max_depth = 5,
    eta = 0.2,
    gamma = 4,
    min_child_weight = 6,
    subsample = 0.7,
    objective = "binary:logistic",
    num_round = 1000
)
```

In [17]:
```
from sagemaker.session import TrainingInput

train_input = TrainingInput(
    "s3://{}/{}/{}".format(bucket, prefix, "data/train.csv"), content_type="csv"
)
validation_input = TrainingInput(
    "s3://{}/{}/{}".format(bucket, prefix, "data/validation.csv"), content_type="csv"
)
```

In [18]:
```
xgb_model.fit({"train": train_input, "validation": validation_input}, wait=True)
```

```
[310]#011train-error:0.11309#011validation-error:0.12920
[311]#011train-error:0.11309#011validation-error:0.12920
[312]#011train-error:0.11302#011validation-error:0.12920
[313]#011train-error:0.11295#011validation-error:0.12879
[314]#011train-error:0.11302#011validation-error:0.12920
[315]#011train-error:0.11295#011validation-error:0.12899
[316]#011train-error:0.11330#011validation-error:0.12879
[317]#011train-error:0.11330#011validation-error:0.12879
[318]#011train-error:0.11323#011validation-error:0.12858
[319]#011train-error:0.11323#011validation-error:0.12899
[320]#011train-error:0.11316#011validation-error:0.12899
[321]#011train-error:0.11302#011validation-error:0.12899
[322]#011train-error:0.11302#011validation-error:0.12899
[323]#011train-error:0.11343#011validation-error:0.12879
[324]#011train-error:0.11302#011validation-error:0.12899
[325]#011train-error:0.11261#011validation-error:0.12858
[326]#011train-error:0.11248#011validation-error:0.12858
[327]#011train-error:0.11289#011validation-error:0.12899
[328]#011train-error:0.11268#011validation-error:0.12899
[329]#011train-error:0.11282#011validation-error:0.12899
```

In [19]:
```
rule_output_path = xgb_model.output_path + "/" + xgb_model.latest_training_job.job_name + "/rule-output"
! aws s3 ls {rule_output_path} --recursive
```

```
2024-04-18 02:53:09     322351 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-report.html
2024-04-18 02:53:09     168681 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-report.ipynb
2024-04-18 02:53:05        191 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/BatchSize.json
2024-04-18 02:53:05        199 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/CPUBottleneck.json
2024-04-18 02:53:05        126 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/Dataloader.json
2024-04-18 02:53:05        127 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/GPUMemoryIncrease.json
2024-04-18 02:53:05        198 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/IOBottleneck.json
2024-04-18 02:53:05        119 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/LoadBalancing.json
2024-04-18 02:53:05        151 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/LowGPUUtilization.json
2024-04-18 02:53:05        179 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/MaxInitializationTime.json
2024-04-18 02:53:05        133 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/OverallFrameworkMetrics.json
2024-04-18 02:53:05        469 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/OverallSystemUsage.json
2024-04-18 02:53:05        156 demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/StepOutlier.json
```

In [20]:
```
! aws s3 cp {rule_output_path} ./ --recursive
```

```
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/BatchSize.json to ProfilerReport/profiler-output/profiler-reports/BatchSize.json
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-report.html to ProfilerReport/profiler-output/profiler-report.html
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/CPUBottleneck.json to ProfilerReport/profiler-output/profiler-reports/CPUBottleneck.json
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-report.ipynb to ProfilerReport/profiler-output/profiler-report.ipynb
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/MaxInitializationTime.json to ProfilerReport/profiler-output/profiler-reports/MaxInitializationTime.json
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/OverallSystemUsage.json to ProfilerReport/profiler-output/profiler-reports/OverallSystemUsage.json
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/StepOutlier.json to ProfilerReport/profiler-output/profiler-reports/StepOutlier.json
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/LowGPUUtilization.json to ProfilerReport/profiler-output/profiler-reports/LowGPUUtilization.json
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/Dataloader.json to ProfilerReport/profiler-output/profiler-reports/Dataloader.json
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/GPUMemoryIncrease.json to ProfilerReport/profiler-output/profiler-reports/GPUMemoryIncrease.json
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/IOBottleneck.json to ProfilerReport/profiler-output/profiler-reports/IOBottleneck.json
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/LoadBalancing.json to ProfilerReport/profiler-output/profiler-reports/LoadBalancing.json
download: s3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/rule-output/ProfilerReport/profiler-output/profiler-reports/OverallFrameworkMetrics.json to ProfilerReport/profiler-output/profiler-reports/OverallFrameworkMetrics.json
```

In [29]:
```
from IPython.display import FileLink, FileLinks
display("Click link below to view the XGBoost Training report", FileLink("CreateXgboostReport/xgboost_report.html"))
```

```
'Click link below to view the XGBoost Training report'
```

Path (`CreateXgboostReport/xgboost_report.html`) doesn't exist. It may still be in the process of being generated, or you may have the incorrect path.

In [31]:
```
profiler_report_name = [rule["RuleConfigurationName"]
                        for rule in xgb_model.latest_training_job.rule_job_summary()
                        if "Profiler" in rule["RuleConfigurationName"]][0]
profiler_report_name
display("Click link below to view the profiler report", FileLink(profiler_report_name+"/profiler-output/profiler-report.html"))
```

```
'Click link below to view the profiler report'
```

ProfilerReport/profiler-output/profiler-report.html (ProfilerReport/profiler-output/profiler-report.html)

In [33]:
```
xgb_model.model_data
```

Out[33]:
```
's3://sagemaker-ap-south-1-992382676037/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-04-18-02-49-21-263/output/model.tar.gz'
```

In [41]:
```
import sagemaker
from sagemaker.serializers import CSVSerializer
xgb_predictor=xgb_model.deploy(
    initial_instance_count=1,
    instance_type='ml.t2.medium',
    serializer=CSVSerializer()
)
```

```
INFO:sagemaker:Creating model with name: sagemaker-xgboost-2024-04-18-03-04-14-511
INFO:sagemaker:Creating endpoint-config with name sagemaker-xgboost-2024-04-18-03-04-14-511
INFO:sagemaker:Creating endpoint with name sagemaker-xgboost-2024-04-18-03-04-14-511

-------!
```

In [42]:
```
xgb_predictor.endpoint_name
```

Out[42]:
```
'sagemaker-xgboost-2024-04-18-03-04-14-511'
```

In [43]:
```
import sagemaker
xgb_predictor_reuse=sagemaker.predictor.Predictor(
    endpoint_name="sagemaker-xgboost-YYYY-MM-DD-HH-MM-SS-SSS",
    sagemaker_session=sagemaker.Session(),
    serializer=sagemaker.serializers.CSVSerializer()
)
```

In [44]:
```
X_test.to_csv('test.csv', index=False, header=False)

boto3.Session().resource('s3').Bucket(bucket).Object(
os.path.join(prefix, 'test/test.csv')).upload_file('test.csv')
```

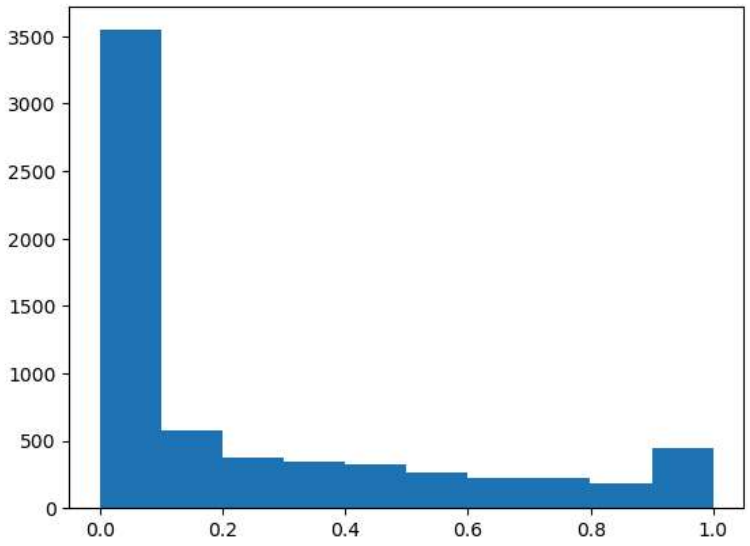
```
INFO:botocore.credentials:Found credentials from IAM Role: BaseNotebookInstanceEc2InstanceRole
```

In [45]:
```
# The location of the test dataset
batch_input = 's3://{}/{}/test'.format(bucket, prefix)

# The location to store the results of the batch transform job
batch_output = 's3://{}/{}/batch-prediction'.format(bucket, prefix)
```

In [48]:
```
! aws s3 cp {batch_output} ./ --recursive
```

In [49]:
```
import numpy as np
def predict(data, rows=1000):
    split_array = np.array_split(data, int(data.shape[0] / float(rows) + 1))
    predictions = ''
    for array in split_array:
        predictions = ','.join([predictions, xgb_predictor.predict(array).decode('utf-8')])
    return np.fromstring(predictions[1:], sep=',')
```

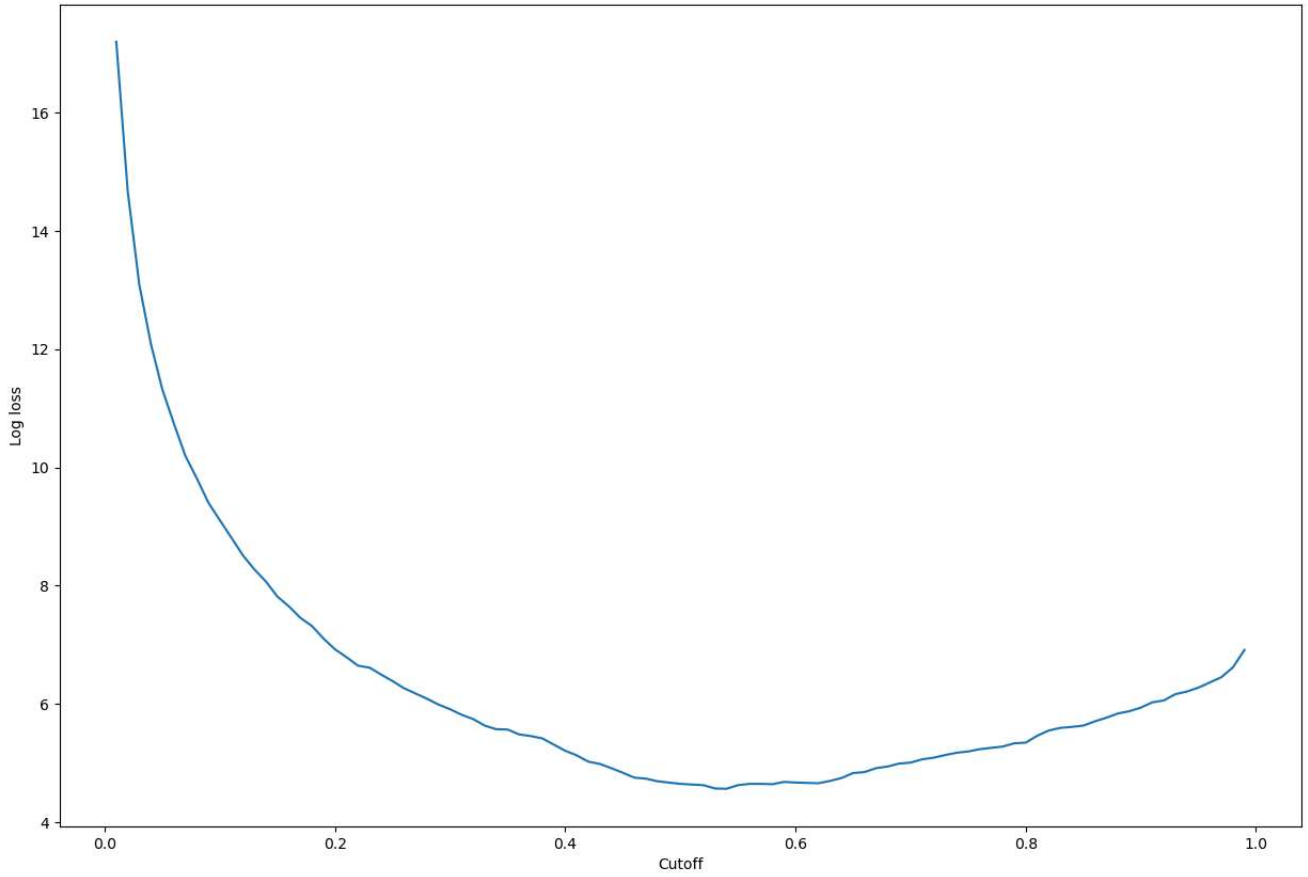
In [50]:
```
import matplotlib.pyplot as plt

predictions=predict(test.to_numpy()[:,1:])
plt.hist(predictions)
plt.show()
```



In [51]:
```
import sklearn

cutoff=0.5
print(sklearn.metrics.confusion_matrix(test.iloc[:, 0], np.where(predictions > cutoff, 1, 0)))
print(sklearn.metrics.classification_report(test.iloc[:, 0], np.where(predictions > cutoff, 1, 0)))
```

```
[[4679  347]
 [ 493  994]]
              precision    recall  f1-score   support

           0       0.90      0.93      0.92      5026
           1       0.74      0.67      0.70      1487

    accuracy                           0.87      6513
   macro avg       0.82      0.80      0.81      6513
weighted avg       0.87      0.87      0.87      6513
```

In [52]:
```
import matplotlib.pyplot as plt

cutoffs = np.arange(0.01, 1, 0.01)
log_loss = []
for c in cutoffs:
    log_loss.append(
        sklearn.metrics.log_loss(test.iloc[:, 0], np.where(predictions > c, 1, 0))
    )

plt.figure(figsize=(15,10))
plt.plot(cutoffs, log_loss)
plt.xlabel("Cutoff")
plt.ylabel("Log loss")
plt.show()
```



In [53]:
```
print(
    'Log loss is minimized at a cutoff of ', cutoffs[np.argmin(log_loss)],
    ', and the log loss value at the minimum is ', np.min(log_loss)
)
```

```
Log loss is minimized at a cutoff of  0.54 , and the log loss value at the minimum is  4.565640111472693
```

In [ ]:

In [ ]: