In [ ]:
```
%pip install sagemaker --upgrade  #--quiet
```

In [ ]:
```
%pip install xgboost==1.3.1 pandas
```

In [2]:
```
import pandas as pd
import boto3
import sagemaker
import json
import joblib
from sagemaker.xgboost.estimator import XGBoost
from sagemaker.tuner import (
    IntegerParameter,
    ContinuousParameter,
    HyperparameterTuner
)
from sagemaker.inputs import TrainingInput
from sagemaker.image_uris import retrieve
from sagemaker.serializers import CSVSerializer
from sagemaker.deserializers import CSVDeserializer

# Setting SageMaker variables
sess = sagemaker.Session()
write_bucket = sess.default_bucket()
write_prefix = "fraud-detect-demo"

region = sess.boto_region_name
s3_client = boto3.client("s3", region_name=region)

sagemaker_role = sagemaker.get_execution_role()
sagemaker_client = boto3.client("sagemaker")
read_bucket = "sagemaker-sample-files"
read_prefix = "datasets/tabular/synthetic_automobile_claims"


# Setting S3 location for read and write operations
train_data_key = f"{read_prefix}/train.csv"
test_data_key = f"{read_prefix}/test.csv"
validation_data_key = f"{read_prefix}/validation.csv"
model_key = f"{write_prefix}/model"
output_key = f"{write_prefix}/output"


train_data_uri = f"s3://{read_bucket}/{train_data_key}"
test_data_uri = f"s3://{read_bucket}/{test_data_key}"
validation_data_uri = f"s3://{read_bucket}/{validation_data_key}"
model_uri = f"s3://{write_bucket}/{model_key}"
output_uri = f"s3://{write_bucket}/{output_key}"
estimator_output_uri = f"s3://{write_bucket}/{write_prefix}/training_jobs"
bias_report_output_uri = f"s3://{write_bucket}/{write_prefix}/clarify-output/bias"
explainability_report_output_uri = f"s3://{write_bucket}/{write_prefix}/clarify-output/explainability"
```

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

In [3]:
```
tuning_job_name_prefix = "xgbtune"
training_job_name_prefix = "xgbtrain"

xgb_model_name = "fraud-detect-xgb-model"
endpoint_name_prefix = "xgb-fraud-model-dev"
train_instance_count = 1
train_instance_type = "ml.m4.xlarge"
predictor_instance_count = 1
predictor_instance_type = "ml.m4.xlarge"
clarify_instance_count = 1
clarify_instance_type = "ml.m4.xlarge"
```

In [4]:
```
%%writefile xgboost_train.py

import argparse
import os
import joblib
import json
import pandas as pd
import xgboost as xgb
from sklearn.metrics import roc_auc_score

if __name__ == "__main__":
    parser = argparse.ArgumentParser()

    # Hyperparameters and algorithm parameters are described here
    parser.add_argument("--num_round", type=int, default=100)
    parser.add_argument("--max_depth", type=int, default=3)
    parser.add_argument("--eta", type=float, default=0.2)
    parser.add_argument("--subsample", type=float, default=0.9)
    parser.add_argument("--colsample_bytree", type=float, default=0.8)
    parser.add_argument("--objective", type=str, default="binary:logistic")
    parser.add_argument("--eval_metric", type=str, default="auc")
    parser.add_argument("--nfold", type=int, default=3)
    parser.add_argument("--early_stopping_rounds", type=int, default=3)


    # SageMaker specific arguments. Defaults are set in the environment variables
    # Location of input training data
    parser.add_argument("--train_data_dir", type=str, default=os.environ.get("SM_CHANNEL_TRAIN"))
    # Location of input validation data
    parser.add_argument("--validation_data_dir", type=str, default=os.environ.get("SM_CHANNEL_VALIDATION"))
    # Location where trained model will be stored. Default set by SageMaker, /opt/ml/model
```

```
    parser.add_argument("--model_dir", type=str, default=os.environ.get("SM_MODEL_DIR"))
    # Location where model artifacts will be stored. Default set by SageMaker, /opt/ml/output/data
    parser.add_argument("--output_data_dir", type=str, default=os.environ.get("SM_OUTPUT_DATA_DIR"))

    args = parser.parse_args()

    data_train = pd.read_csv(f"{args.train_data_dir}/train.csv")
    train = data_train.drop("fraud", axis=1)
    label_train = pd.DataFrame(data_train["fraud"])
    dtrain = xgb.DMatrix(train, label=label_train)


    data_validation = pd.read_csv(f"{args.validation_data_dir}/validation.csv")
    validation = data_validation.drop("fraud", axis=1)
    label_validation = pd.DataFrame(data_validation["fraud"])
    dvalidation = xgb.DMatrix(validation, label=label_validation)

    params = {"max_depth": args.max_depth,
              "eta": args.eta,
              "objective": args.objective,
              "subsample" : args.subsample,
              "colsample_bytree":args.colsample_bytree
             }

    num_boost_round = args.num_round
    nfold = args.nfold
    early_stopping_rounds = args.early_stopping_rounds

    cv_results = xgb.cv(
        params=params,
        dtrain=dtrain,
        num_boost_round=num_boost_round,
        nfold=nfold,
        early_stopping_rounds=early_stopping_rounds,
        metrics=["auc"],
        seed=42,
    )

    model = xgb.train(params=params, dtrain=dtrain, num_boost_round=len(cv_results))

    train_pred = model.predict(dtrain)
    validation_pred = model.predict(dvalidation)

    train_auc = roc_auc_score(label_train, train_pred)
    validation_auc = roc_auc_score(label_validation, validation_pred)

    print(f"[0]#011train-auc:{train_auc:.2f}")
    print(f"[0]#011validation-auc:{validation_auc:.2f}")

    metrics_data = {"hyperparameters" : params,
                    "binary_classification_metrics": {"validation:auc": {"value": validation_auc},
                                                      "train:auc": {"value": train_auc}
                                                     }
                   }

    # Save the evaluation metrics to the location specified by output_data_dir
    metrics_location = args.output_data_dir + "/metrics.json"

    # Save the model to the location specified by model_dir
    model_location = args.model_dir + "/xgboost-model"

    with open(metrics_location, "w") as f:
        json.dump(metrics_data, f)

    with open(model_location, "wb") as f:
        joblib.dump(model, f)
```

```
Overwriting xgboost_train.py
```

In [5]:
```
# SageMaker estimator

# Set static hyperparameters that will not be tuned
static_hyperparams = {
                      "eval_metric" : "auc",
                      "objective": "binary:logistic",
                      "num_round": "5"
                     }

xgb_estimator = XGBoost(
                      entry_point="xgboost_train.py",
                      output_path=estimator_output_uri,
                      code_location=estimator_output_uri,
                      hyperparameters=static_hyperparams,
                      role=sagemaker_role,
                      instance_count=train_instance_count,
                      instance_type=train_instance_type,
                      framework_version="1.3-1",
                      base_job_name=training_job_name_prefix
                    )
```

In [6]:
```
# Setting ranges of hyperparameters to be tuned
hyperparameter_ranges = {
    "eta": ContinuousParameter(0, 1),
    "subsample": ContinuousParameter(0.7, 0.95),
    "colsample_bytree": ContinuousParameter(0.7, 0.95),
    "max_depth": IntegerParameter(1, 5)
}
```

In [7]:
```
objective_metric_name = "validation:auc"
```

```
# Setting up tuner object
tuner_config_dict = {
                     "estimator" : xgb_estimator,
                     "max_jobs" : 5,
                     "max_parallel_jobs" : 2,
                     "objective_metric_name" : objective_metric_name,
                     "hyperparameter_ranges" : hyperparameter_ranges,
                     "base_tuning_job_name" : tuning_job_name_prefix,
                     "strategy" : "Random"
                    }
tuner = HyperparameterTuner(**tuner_config_dict)
```

In [8]:
```
# Setting the input channels for tuning job
s3_input_train = TrainingInput(s3_data="s3://{}/{}".format(read_bucket, train_data_key), content_type="csv", s3_data_type="S3Prefix")
s3_input_validation = (TrainingInput(s3_data="s3://{}/{}".format(read_bucket, validation_data_key),
                                     content_type="csv", s3_data_type="S3Prefix")
                       )

tuner.fit(inputs={"train": s3_input_train, "validation": s3_input_validation}, include_cls_metadata=False)
tuner.wait()
```

```
No finished training job found associated with this estimator. Please make sure this estimator is only used for building workflow config
.....................................................!
!
```

In [9]:
```
# Summary of tuning results ordered in descending order of performance
df_tuner = sagemaker.HyperparameterTuningJobAnalytics(tuner.latest_tuning_job.job_name).dataframe()
df_tuner = df_tuner[df_tuner["FinalObjectiveValue"]>-float('inf')].sort_values("FinalObjectiveValue", ascending=False)
df_tuner
```

Out[9]:

	colsample_bytree	eta	max_depth	subsample	TrainingJobName	TrainingJobStatus	FinalObjectiveValue	TrainingStartTime	TrainingEndTime	TrainingElapsedTimeSeconds
1	0.930159	0.960635	2.0	0.725262	xgbtune-240428-2158-004-34517db2	Completed	0.80	2024-04-28 22:01:19+00:00	2024-04-28 22:02:02+00:00	43.0
2	0.715721	0.831568	5.0	0.829370	xgbtune-240428-2158-003-036d93ca	Completed	0.70	2024-04-28 22:01:18+00:00	2024-04-28 22:01:59+00:00	41.0
3	0.879741	0.927272	1.0	0.700078	xgbtune-240428-2158-002-1ad6b716	Completed	0.68	2024-04-28 21:59:08+00:00	2024-04-28 22:01:05+00:00	117.0
4	0.855023	0.755502	1.0	0.777568	xgbtune-240428-2158-001-2018362d	Completed	0.68	2024-04-28 21:59:10+00:00	2024-04-28 22:01:08+00:00	118.0
0	0.843129	0.853527	1.0	0.771150	xgbtune-240428-2158-005-30e5cac6	Completed	0.64	2024-04-28 22:02:08+00:00	2024-04-28 22:02:50+00:00	42.0

In [10]:
```
tuner_job_info = sagemaker_client.describe_hyper_parameter_tuning_job(HyperParameterTuningJobName=tuner.latest_tuning_job.job_name)

model_matches = sagemaker_client.list_models(NameContains=xgb_model_name)["Models"]

if not model_matches:
    _ = sess.create_model_from_job(
            name=xgb_model_name,
            training_job_name=tuner_job_info['BestTrainingJob']["TrainingJobName"],
            role=sagemaker_role,
            image_uri=tuner_job_info['TrainingJobDefinition']["AlgorithmSpecification"]["TrainingImage"]
            )
else:

    print(f"Model {xgb_model_name} already exists.")
```

In [11]:
```
train_df = pd.read_csv(train_data_uri)
train_df_cols = train_df.columns.to_list()

clarify_processor = sagemaker.clarify.SageMakerClarifyProcessor(
    role=sagemaker_role,
    instance_count=clarify_instance_count,
    instance_type=clarify_instance_type,
    sagemaker_session=sess,
)

# Data config
bias_data_config = sagemaker.clarify.DataConfig(
    s3_data_input_path=train_data_uri,
    s3_output_path=bias_report_output_uri,
    label="fraud",
    headers=train_df_cols,
    dataset_type="text/csv",
)

# Model config
model_config = sagemaker.clarify.ModelConfig(
    model_name=xgb_model_name,
    instance_type=train_instance_type,
    instance_count=1,
    accept_type="text/csv",
)

# Model predictions config to get binary labels from probabilities
predictions_config = sagemaker.clarify.ModelPredictedLabelConfig(probability_threshold=0.5)

# Bias config
bias_config = sagemaker.clarify.BiasConfig(
    label_values_or_threshold=[0],
    facet_name="customer_gender_female",
    facet_values_or_threshold=[1],
)
```

In [ ]:
```
clarify_processor.run_bias(
    data_config=bias_data_config,
    bias_config=bias_config,
    model_config=model_config,
    model_predicted_label_config=predictions_config,
    pre_training_methods=["CI"],
    post_training_methods=["DPPL"]
    )

clarify_bias_job_name = clarify_processor.latest_job.name
```

```
INFO:sagemaker:Creating processing-job with name Clarify-Bias-2024-04-28-22-09-12-032
.........................................
```

In [13]:
```
# Copy bias report and view locally
!aws s3 cp s3://{write_bucket}/{write_prefix}/clarify-output/bias/report.pdf ./clarify_bias_output.pdf
```

```
download: s3://sagemaker-us-east-1-711638914386/fraud-detect-demo/clarify-output/bias/report.pdf to ./clarify_bias_output.pdf
```

In [ ]:
```
explainability_data_config = sagemaker.clarify.DataConfig(
    s3_data_input_path=train_data_uri,
    s3_output_path=explainability_report_output_uri,
    label="fraud",
    headers=train_df_cols,
    dataset_type="text/csv",
)

# Use mean of train dataset as baseline data point
shap_baseline = [list(train_df.drop(["fraud"], axis=1).mean())]

shap_config = sagemaker.clarify.SHAPConfig(
    baseline=shap_baseline,
    num_samples=500,
    agg_method="mean_abs",
    save_local_shap_values=True,
)

clarify_processor.run_explainability(
    data_config=explainability_data_config,
    model_config=model_config,
    explainability_config=shap_config
)
```

```
INFO:sagemaker.clarify:Analysis Config: {'dataset_type': 'text/csv', 'headers': ['fraud', 'num_vehicles_involved', 'num_injuries', 'num_witnesses', 'police
_report_available', 'injury_claim', 'vehicle_claim', 'total_claim_amount', 'incident_month', 'incident_day', 'incident_dow', 'incident_hour', 'customer_ag
e', 'months_as_customer', 'num_claims_past_year', 'num_insurers_past_5_years', 'policy_deductable', 'policy_annual_premium', 'policy_liability', 'customer_
education', 'auto_year', 'driver_relationship_other', 'driver_relationship_child', 'driver_relationship_spouse', 'driver_relationship_na', 'driver_relation
ship_self', 'incident_type_collision', 'incident_type_break-in', 'incident_type_theft', 'collision_type_rear', 'collision_type_side', 'collision_type_na',
'collision_type_front', 'incident_severity_totaled', 'incident_severity_major', 'incident_severity_minor', 'authorities_contacted_fire', 'authorities_conta
cted_none', 'authorities_contacted_police', 'authorities_contacted_ambulance', 'policy_state_ca', 'policy_state_az', 'policy_state_nv', 'policy_state_id',
'policy_state_wa', 'policy_state_or', 'customer_gender_other', 'customer_gender_male', 'customer_gender_female'], 'label': 'fraud', 'predictor': {'model_na
me': 'fraud-detect-xgb-model', 'instance_type': 'ml.m4.xlarge', 'initial_instance_count': 1, 'accept_type': 'text/csv'}, 'methods': {'report': {'name': 're
port', 'title': 'Analysis Report'}, 'shap': {'use_logit': False, 'save_local_shap_values': True, 'baseline': [[2.1085058618109254, 0.5584933898727862, 0.86
85457720129708, 0.4220503866300823, 24257.121476677476, 17169.351123437555, 41426.472600115034, 6.726365677226241, 15.585682215016213, 2.645048640558743, 1
1.722624095784484, 44.15714642055375, 98.60688450985283, 0.08730356697430781, 1.4130705911698678, 751.0725866799701, 2925.3305063606886, 1.118233973559491,
2.531304564729359, 2015.7251184834124, 0.04065851833374907, 0.04489897730107259, 0.08505861810925418, 0.14342728860064854, 0.6859565976552756, 0.8565727113
993514, 0.09553504614617112, 0.04789224245447742, 0.21900723372412073, 0.21052631578947367, 0.14342728860064854, 0.427039161885757, 0.2354701920678473 3, 0.
34846595160888, 0.41606385632327264, 0.024195559990022448, 0.2432027937141432, 0.7031678722873534, 0.029433774008480917, 0.6303317535545023, 0.107009229234
223, 0.04190571214766775, 0.027937141431778497, 0.12297331005238214, 0.06984285357944625, 0.0177101521576453, 0.4622100274382639, 0.43576951858318785]], 'n
um_samples': 500, 'agg_method': 'mean_abs'}}}
INFO:sagemaker:Creating processing-job with name Clarify-Explainability-2024-04-28-22-29-57-366
```

```
.............................................WARNING:root:logging.conf not found when configuring logging, using default logging configuration.
INFO:sagemaker-clarify-processing:Starting SageMaker Clarify Processing job
INFO:analyzer.data_loading.data_loader_util:Analysis config path: /opt/ml/processing/input/config/analysis_config.json
INFO:analyzer.data_loading.data_loader_util:Analysis result path: /opt/ml/processing/output
INFO:analyzer.data_loading.data_loader_util:This host is algo-1.
INFO:analyzer.data_loading.data_loader_util:This host is the leader.
INFO:analyzer.data_loading.data_loader_util:Number of hosts in the cluster is 1.
INFO:sagemaker-clarify-processing:Running Python / Pandas based analyzer.
INFO:analyzer.data_loading.data_loader_factory:Dataset type: text/csv uri: /opt/ml/processing/input/data
/usr/local/lib/python3.9/site-packages/analyzer/data_loading/data_readers/csv_data_reader.py:58: FutureWarning: The frame.append method is deprecated and w
ill be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(df_tmp, ignore_index=True)
/usr/local/lib/python3.9/site-packages/analyzer/data_loading/data_readers/csv_data_reader.py:58: FutureWarning: The frame.append method is deprecated and w
ill be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(df_tmp, ignore_index=True)
INFO:sagemaker-clarify-processing:Loading dataset...
/usr/local/lib/python3.9/site-packages/analyzer/data_loading/data_readers/csv_data_reader.py:58: FutureWarning: The frame.append method is deprecated and w
ill be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(df_tmp, ignore_index=True)
INFO:sagemaker-clarify-processing:Loaded dataset. Dataset info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4009 entries, 0 to 4008
Data columns (total 48 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   num_vehicles_involved           4009 non-null   int64
 1   num_injuries                    4009 non-null   int64
 2   num_witnesses                   4009 non-null   int64
 3   police_report_available         4009 non-null   int64
 4   injury_claim                    4009 non-null   int64
 5   vehicle_claim                   4009 non-null   float64
 6   total_claim_amount              4009 non-null   float64
 7   incident_month                  4009 non-null   int64
 8   incident_day                    4009 non-null   int64
 9   incident_dow                    4009 non-null   int64
 10  incident_hour                   4009 non-null   int64
 11  customer_age                    4009 non-null   int64
 12  months_as_customer              4009 non-null   int64
 13  num_claims_past_year            4009 non-null   int64
 14  num_insurers_past_5_years       4009 non-null   int64
 15  policy_deductable               4009 non-null   int64
 16  policy_annual_premium           4009 non-null   int64
 17  policy_liability                4009 non-null   int64
 18  customer_education              4009 non-null   int64
 19  auto_year                       4009 non-null   int64
 20  driver_relationship_other       4009 non-null   int64
 21  driver_relationship_child       4009 non-null   int64
 22  driver_relationship_spouse      4009 non-null   int64
 23  driver_relationship_na          4009 non-null   int64
 24  driver_relationship_self        4009 non-null   int64
 25  incident_type_collision         4009 non-null   int64
 26  incident_type_break-in          4009 non-null   int64
 27  incident_type_theft             4009 non-null   int64
 28  collision_type_rear             4009 non-null   int64
 29  collision_type_side             4009 non-null   int64
 30  collision_type_na               4009 non-null   int64
 31  collision_type_front            4009 non-null   int64
 32  incident_severity_totaled       4009 non-null   int64
 33  incident_severity_major         4009 non-null   int64
 34  incident_severity_minor         4009 non-null   int64
 35  authorities_contacted_fire      4009 non-null   int64
 36  authorities_contacted_none      4009 non-null   int64
 37  authorities_contacted_police    4009 non-null   int64
 38  authorities_contacted_ambulance 4009 non-null   int64
 39  policy_state_ca                 4009 non-null   int64
 40  policy_state_az                 4009 non-null   int64
 41  policy_state_nv                 4009 non-null   int64
 42  policy_state_id                 4009 non-null   int64
 43  policy_state_wa                 4009 non-null   int64
 44  policy_state_or                 4009 non-null   int64
 45  customer_gender_other           4009 non-null   int64
 46  customer_gender_male            4009 non-null   int64
 47  customer_gender_female          4009 non-null   int64
dtypes: float64(2), int64(46)
memory usage: 1.5 MB
INFO:analyzer.predictor.managed_endpoint:Spinning up shadow endpoint
INFO:sagemaker:Creating endpoint-config with name sm-clarify-config-1714343875-2d05
INFO:analyzer.predictor.managed_endpoint:Creating endpoint: 'sm-clarify-fraud-detect-xgb-model-1714343875-30c5'
INFO:botocore.client:No endpoints ruleset found for service sagemaker-internal, falling back to legacy endpoint routing.
INFO:sagemaker-clarify-processing:Using endpoint name: sm-clarify-fraud-detect-xgb-model-1714343875-30c5
INFO:sagemaker-clarify-processing:Waiting for endpoint ...
INFO:analyzer.predictor.managed_endpoint:Checking endpoint status:
Legend:
(OutOfService: x, Creating: -, Updating: -, InService: !, RollingBack: <, Deleting: o, Failed: *)
INFO:analyzer.predictor.managed_endpoint:Endpoint is in service after 181 seconds
INFO:sagemaker-clarify-processing:Endpoint ready.
INFO:explainers.shap.kernel_shap:Clarify Kernel SHAP n_coalitions: 500, n_instances: 1, n_features_to_explain: 48, model_output_size: 1
INFO:analyzer.shap.shap_analyzer:====================================================
INFO:analyzer.shap.shap_analyzer:Shap analyzer: explaining 4009 rows, 48 columns...
INFO:analyzer.shap.shap_analyzer:====================================================
  0% (0 of 4009) |                       | Elapsed Time: 0:00:00 ETA:  --:--:--
  3% (158 of 4009) |                     | Elapsed Time: 0:00:30 ETA:   0:12:15
  7% (319 of 4009) |#                    | Elapsed Time: 0:01:00 ETA:   0:11:37
 11% (479 of 4009) |##                   | Elapsed Time: 0:01:30 ETA:   0:11:05
 15% (641 of 4009) |###                  | Elapsed Time: 0:02:00 ETA:   0:10:33
 20% (802 of 4009) |####                 | Elapsed Time: 0:02:30 ETA:   0:10:02
 24% (965 of 4009) |#####                | Elapsed Time: 0:03:00 ETA:   0:09:30
 28% (1126 of 4009) |#####               | Elapsed Time: 0:03:30 ETA:   0:08:59
 32% (1287 of 4009) |######              | Elapsed Time: 0:04:00 ETA:   0:08:29
 36% (1448 of 4009) |#######             | Elapsed Time: 0:04:30 ETA:   0:07:59
```

```
 40% (1610 of 4009) |########            | Elapsed Time: 0:05:01 ETA:   0:07:28
 44% (1771 of 4009) |########            | Elapsed Time: 0:05:31 ETA:   0:06:58
 48% (1932 of 4009) |#########           | Elapsed Time: 0:06:01 ETA:   0:06:28
 52% (2094 of 4009) |##########          | Elapsed Time: 0:06:31 ETA:   0:05:57
 56% (2255 of 4009) |###########         | Elapsed Time: 0:07:01 ETA:   0:05:27
 60% (2415 of 4009) |############        | Elapsed Time: 0:07:31 ETA:   0:04:57
 64% (2577 of 4009) |############        | Elapsed Time: 0:08:01 ETA:   0:04:27
 68% (2739 of 4009) |#############       | Elapsed Time: 0:08:31 ETA:   0:03:57
 72% (2902 of 4009) |##############      | Elapsed Time: 0:09:01 ETA:   0:03:26
 76% (3065 of 4009) |###############     | Elapsed Time: 0:09:31 ETA:   0:02:56
 80% (3228 of 4009) |################    | Elapsed Time: 0:10:01 ETA:   0:02:25
 84% (3391 of 4009) |################    | Elapsed Time: 0:10:31 ETA:   0:01:55
 88% (3554 of 4009) |#################   | Elapsed Time: 0:11:01 ETA:   0:01:24
 92% (3718 of 4009) |##################  | Elapsed Time: 0:11:31 ETA:   0:00:54
 96% (3882 of 4009) |################### | Elapsed Time: 0:12:02 ETA:   0:00:23
100% (4009 of 4009) |####################| Elapsed Time: 0:12:25 Time:  0:12:25
INFO:analyzer.shap.shap_analyzer:getting explanations took 745.73 seconds.
INFO:analyzer.shap.shap_analyzer:==================================================
WARNING:analyzer.shap.shap_util:Falling back to generic labels: label0, label1, ...
INFO:analyzer.shap.shap_analyzer:converting explanations to tabular took 0.63 seconds.
INFO:analyzer.shap.shap_analyzer:==================================================
INFO:analyzer.shap.shap_analyzer:Wrote baseline used to compute explanations to: /opt/ml/processing/output/explanations_shap/baseline.csv
INFO:analyzer.shap.shap_analyzer:Wrote 4009 local explanations to: /opt/ml/processing/output/explanations_shap/out.csv
INFO:analyzer.shap.shap_analyzer:writing local explanations took 0.17 seconds.
INFO:analyzer.shap.shap_analyzer:==================================================
/usr/local/lib/python3.9/site-packages/numpy/core/fromnumeric.py:3430: FutureWarning: In a future version, DataFrame.mean(axis=None) will return a scalar m
ean over the entire DataFrame. To retain the old behavior, use 'frame.mean(axis=0)' or just 'frame.mean()'
  return mean(axis=axis, dtype=dtype, out=out, **kwargs)
INFO:analyzer.shap.shap_analyzer:aggregating local explanations took 0.00 seconds.
INFO:analyzer.shap.shap_analyzer:==================================================
INFO:analyzer.shap.shap_analyzer:Shap analysis finished.
INFO:sagemaker-clarify-processing:Calculated global analysis with predictor
INFO:analyzer.predictor.predictor:Stop using endpoint: sm-clarify-fraud-detect-xgb-model-1714343875-30c5
INFO:sagemaker:Deleting endpoint configuration with name: sm-clarify-config-1714343875-2d05
INFO:sagemaker:Deleting endpoint with name: sm-clarify-fraud-detect-xgb-model-1714343875-30c5
INFO:analyzer.predictor.managed_endpoint:Model endpoint delivered 5.37148 requests per second and a total of 4011 requests over 747 seconds
INFO:sagemaker-clarify-processing:Calculated global analysis without predictor
INFO:sagemaker-clarify-processing:Collected analyses:
{'version': '1.0', 'explanations': {'kernel_shap': defaultdict(<function <lambda> at 0x7f5b9d8e0040>, {'label0': defaultdict(<function <lambda> at 0x7f5b9d
8e0040>, {'global_shap_values': defaultdict(<function <lambda> at 0x7f5b9d8e0040>, {'num_vehicles_involved': 0.00392150879118399, 'num_injuries': 0.0048490
90625918709, 'num_witnesses': 0.001572701003359833, 'police_report_available': 0.0001168588056336879, 'injury_claim': 0.00011393384680107584, 'vehicle_clai
m': 0.0005993315744167583, 'total_claim_amount': 0.00011528601649159885, 'incident_month': 0.00034458016277888964, 'incident_day': 0.002776792784586135, 'i
ncident_dow': 0.0008216149474696238, 'incident_hour': 0.00011397507411251623, 'customer_age': 0.00012119205549435232, 'months_as_customer': 0.0005467563211
479715, 'num_claims_past_year': 0.0016771030030709188, 'num_insurers_past_5_years': 0.0024645670782113688, 'policy_deductable': 0.00011174387007239662, 'po
licy_annual_premium': 0.00010972635901540262, 'policy_liability': 0.0001172352096988818, 'customer_education': 0.00011264988965524587, 'auto_year': 0.00011
817545214182857, 'driver_relationship_other': 0.00011611715764970212, 'driver_relationship_child': 0.00010827797140714944, 'driver_relationship_spouse': 0.
00012296217870248 71, 'driver_relationship_na': 0.00011279658035210293, 'driver_relationship_self': 0.0001190177546178634, 'incident_type_collision': 0.0001
142548856740222, 'incident_type_break-in': 0.00012001881822030 43, 'incident_type_theft': 0.00011280024719295129, 'collision_type_rear': 0.00011587114863936
616, 'collision_type_side': 0.00011590375056344209, 'collision_type_na': 0.000114891336639244, 'collision_type_front': 0.00011323096304079115, 'incident_se
verity_totaled': 0.00011695715083163813, 'incident_severity_major': 0.00011593635495433597, 'incident_severity_minor': 0.00010995571124886854, 'authorities
_contacted_fire': 0.00012023245894904406, 'authorities_contacted_none': 0.00012034575106473229, 'authorities_contacted_police': 0.00010594299652162646, 'au
thorities_contacted_ambulance': 0.00011749300092872615, 'policy_state_ca': 0.00010501633793380241, 'policy_state_az': 0.00010803150988872866, 'policy_state
_nv': 0.00011816003209981373, 'policy_state_id': 0.0001166100935781299, 'policy_state_wa': 0.00011384770144059727, 'policy_state_or': 0.00011717222589240942, 'customer_gender_other': 0.00011458753780493927, 'customer_gender_male': 0.00886072662464185, 'customer_gender_female': 0.00012275815033980318}), 'expec
ted_value': 0.009246491827070713})})}}
```

In [ ]:
```
# Copy explainability report and view
!aws s3 cp s3://{write_bucket}/{write_prefix}/clarify-output/explainability/report.pdf ./clarify_explainability_output.pdf
```

In [ ]:
```
import matplotlib.pyplot as plt
import matplotlib
%matplotlib inline
local_explanations_out = pd.read_csv(explainability_report_output_uri + "/explanations_shap/out.csv")
feature_names = [str.replace(c, "_label0", "") for c in
local_explanations_out.columns.to_series()]
local_explanations_out.columns = feature_names

selected_example = 100
print("Example number:", selected_example)

local_explanations_out.iloc[selected_example].plot(
    kind="bar", title="Local explanation for the example number " + str(selected_example), rot=60, figsize=(20, 8)
);
```

In [ ]:
```
```