Training machine learning models on tabular data: Resume

It covers the following steps:

- Visualize the data using Seaborn and matplotlib
- Run a parallel hyperparameter sweep to train machine learning models on the dataset
- Explore the results of the hyperparameter sweep with MLflow

In this example, I build a model to predict whether the resume owner will receive call back or not based on the resume properties.

The example uses a dataset from openintro https://www.openintro.org/data/index.php?data=resume

Requirements

This notebook requires Databricks Runtime for Machine Learning.
If you are using Databricks Runtime 7.3 LTS ML, you must update the CloudPickle library.
To do that, uncomment and run the `%pip install` command in Cmd 2.

In [ ]:
```
# This command is only required if you are using a cluster running DBR 7.3 L
#!pip install --upgrade cloudpickle
!pip install mlflow
```

```
Collecting mlflow
  Downloading mlflow-2.8.1-py3-none-any.whl (19.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 19.0/19.0 MB 48.6 MB/s eta 0:0
0:00
Requirement already satisfied: matplotlib<4 in /databricks/python3/lib/pytho
n3.10/site-packages (from mlflow) (3.5.2)
Requirement already satisfied: packaging<24 in /databricks/python3/lib/pytho
n3.10/site-packages (from mlflow) (21.3)
Collecting databricks-cli<1,>=0.8.7
  Downloading databricks_cli-0.18.0-py2.py3-none-any.whl (150 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 150.3/150.3 kB 12.5 MB/s eta 0:0
0:00
Requirement already satisfied: click<9,>=7.0 in /databricks/python3/lib/pyth
on3.10/site-packages (from mlflow) (8.0.4)
Collecting pyyaml<7,>=5.1
  Downloading PyYAML-6.0.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x
86_64.whl (705 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 705.5/705.5 kB 45.3 MB/s eta 0:0
0:00
Collecting gunicorn<22
  Downloading gunicorn-21.2.0-py3-none-any.whl (80 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 80.2/80.2 kB 10.7 MB/s eta 0:0
0:00
Requirement already satisfied: pytz<2024 in /databricks/python3/lib/python3.
10/site-packages (from mlflow) (2022.1)
Requirement already satisfied: pyarrow<15,>=4.0.0 in /databricks/python3/li
b/python3.10/site-packages (from mlflow) (8.0.0)
Collecting cloudpickle<3
  Downloading cloudpickle-2.2.1-py3-none-any.whl (25 kB)
Collecting Flask<4
  Downloading flask-3.0.0-py3-none-any.whl (99 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.7/99.7 kB 12.2 MB/s eta 0:0
0:00
Requirement already satisfied: requests<3,>=2.17.3 in /databricks/python3/li
b/python3.10/site-packages (from mlflow) (2.28.1)
Requirement already satisfied: importlib-metadata!=4.7.0,<7,>=3.7.0 in /usr/
lib/python3/dist-packages (from mlflow) (4.6.4)
Collecting docker<7,>=4.0.0
  Downloading docker-6.1.3-py3-none-any.whl (148 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 148.1/148.1 kB 17.1 MB/s eta 0:0
0:00
Requirement already satisfied: entrypoints<1 in /databricks/python3/lib/pyth
on3.10/site-packages (from mlflow) (0.4)
Collecting querystring-parser<2
  Downloading querystring_parser-1.2.4-py2.py3-none-any.whl (7.9 kB)
Requirement already satisfied: protobuf<5,>=3.12.0 in /databricks/python3/li
b/python3.10/site-packages (from mlflow) (3.19.4)
Requirement already satisfied: numpy<2 in /databricks/python3/lib/python3.1
0/site-packages (from mlflow) (1.21.5)
Requirement already satisfied: Jinja2<4,>=2.11 in /databricks/python3/lib/py
thon3.10/site-packages (from mlflow) (2.11.3)
Collecting sqlalchemy<3,>=1.4.0
  Downloading SQLAlchemy-2.0.23-cp310-cp310-manylinux_2_17_x86_64.manylinux2
014_x86_64.whl (3.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.0/3.0 MB 77.0 MB/s eta 0:00:
00
```

```
Collecting sqlparse<1,>=0.4.0
  Downloading sqlparse-0.4.4-py3-none-any.whl (41 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.2/41.2 kB 1.5 MB/s eta 0:0
0:00
Collecting markdown<4,>=3.3
  Downloading Markdown-3.5.1-py3-none-any.whl (102 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 102.2/102.2 kB 12.7 MB/s eta 0:0
0:00
Collecting gitpython<4,>=2.1.0
  Downloading GitPython-3.1.40-py3-none-any.whl (190 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 190.6/190.6 kB 23.6 MB/s eta 0:0
0:00
Requirement already satisfied: scipy<2 in /databricks/python3/lib/python3.1
0/site-packages (from mlflow) (1.9.1)
Requirement already satisfied: pandas<3 in /databricks/python3/lib/python3.1
0/site-packages (from mlflow) (1.4.4)
Requirement already satisfied: scikit-learn<2 in /databricks/python3/lib/pyt
hon3.10/site-packages (from mlflow) (1.1.1)
Collecting alembic!=1.10.0,<2
  Downloading alembic-1.12.1-py3-none-any.whl (226 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 226.8/226.8 kB 10.4 MB/s eta 0:0
0:00
Requirement already satisfied: typing-extensions>=4 in /databricks/python3/l
ib/python3.10/site-packages (from alembic!=1.10.0,<2->mlflow) (4.3.0)
Collecting Mako
  Downloading Mako-1.3.0-py3-none-any.whl (78 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.6/78.6 kB 7.6 MB/s eta 0:0
0:00
Requirement already satisfied: urllib3<3,>=1.26.7 in /databricks/python3/li
b/python3.10/site-packages (from databricks-cli<1,>=0.8.7->mlflow) (1.26.11)
Collecting tabulate>=0.7.7
  Downloading tabulate-0.9.0-py3-none-any.whl (35 kB)
Requirement already satisfied: six>=1.10.0 in /usr/lib/python3/dist-packages
(from databricks-cli<1,>=0.8.7->mlflow) (1.16.0)
Requirement already satisfied: oauthlib>=3.1.0 in /usr/lib/python3/dist-pack
ages (from databricks-cli<1,>=0.8.7->mlflow) (3.2.0)
Requirement already satisfied: pyjwt>=1.7.0 in /usr/lib/python3/dist-package
s (from databricks-cli<1,>=0.8.7->mlflow) (2.3.0)
Collecting websocket-client>=0.32.0
  Downloading websocket_client-1.6.4-py3-none-any.whl (57 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 57.3/57.3 kB 7.7 MB/s eta 0:0
0:00
Collecting blinker>=1.6.2
  Downloading blinker-1.7.0-py3-none-any.whl (13 kB)
Collecting itsdangerous>=2.1.2
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting Jinja2<4,>=2.11
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 133.1/133.1 kB 12.0 MB/s eta 0:0
0:00
Collecting click<9,>=7.0
  Downloading click-8.1.7-py3-none-any.whl (97 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 97.9/97.9 kB 8.8 MB/s eta 0:0
0:00
Collecting Werkzeug>=3.0.0
  Downloading werkzeug-3.0.1-py3-none-any.whl (226 kB)
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 226.7/226.7 kB 24.8 MB/s eta 0:0
0:00
Collecting gitdb<5,>=4.0.1
  Downloading gitdb-4.0.11-py3-none-any.whl (62 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.7/62.7 kB 8.2 MB/s eta 0:0
0:00
Requirement already satisfied: MarkupSafe>=2.0 in /databricks/python3/lib/py
thon3.10/site-packages (from Jinja2<4,>=2.11->mlflow) (2.0.1)
Requirement already satisfied: cycler>=0.10 in /databricks/python3/lib/pytho
n3.10/site-packages (from matplotlib<4->mlflow) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in /databricks/python3/lib/pyth
on3.10/site-packages (from matplotlib<4->mlflow) (9.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /databricks/python3/l
ib/python3.10/site-packages (from matplotlib<4->mlflow) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /databricks/python3/lib/
python3.10/site-packages (from matplotlib<4->mlflow) (1.4.2)
Requirement already satisfied: fonttools>=4.22.0 in /databricks/python3/lib/
python3.10/site-packages (from matplotlib<4->mlflow) (4.25.0)
Requirement already satisfied: pyparsing>=2.2.1 in /databricks/python3/lib/p
ython3.10/site-packages (from matplotlib<4->mlflow) (3.0.9)
Requirement already satisfied: charset-normalizer<3,>=2 in /databricks/pytho
n3/lib/python3.10/site-packages (from requests<3,>=2.17.3->mlflow) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /databricks/python3/li
b/python3.10/site-packages (from requests<3,>=2.17.3->mlflow) (2022.9.14)
Requirement already satisfied: idna<4,>=2.5 in /databricks/python3/lib/pytho
n3.10/site-packages (from requests<3,>=2.17.3->mlflow) (3.3)
Requirement already satisfied: joblib>=1.0.0 in /databricks/python3/lib/pyth
on3.10/site-packages (from scikit-learn<2->mlflow) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /databricks/python3/l
ib/python3.10/site-packages (from scikit-learn<2->mlflow) (2.2.0)
Collecting greenlet!=0.4.17
  Downloading greenlet-3.0.1-cp310-cp310-manylinux_2_24_x86_64.manylinux_2_2
8_x86_64.whl (613 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 613.2/613.2 kB 22.1 MB/s eta 0:0
0:00
Collecting smmap<6,>=3.0.1
  Downloading smmap-5.0.1-py3-none-any.whl (24 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.3-cp310-cp310-manylinux_2_17_x86_64.manylinux20
14_x86_64.whl (25 kB)
Installing collected packages: websocket-client, tabulate, sqlparse, smmap,
querystring-parser, pyyaml, MarkupSafe, markdown, itsdangerous, greenlet, cl
oudpickle, click, blinker, Werkzeug, sqlalchemy, Mako, Jinja2, gunicorn, git
db, docker, databricks-cli, gitpython, Flask, alembic, mlflow
  Attempting uninstall: MarkupSafe
    Found existing installation: MarkupSafe 2.0.1
    Not uninstalling markupsafe at /databricks/python3/lib/python3.10/site-p
ackages, outside environment /local_disk0/.ephemeral_nfs/envs/pythonEnv-2df7
3871-3175-4be4-8c95-414efb8964ca
    Can't uninstall 'MarkupSafe'. No files were found to uninstall.
  Attempting uninstall: click
    Found existing installation: click 8.0.4
    Not uninstalling click at /databricks/python3/lib/python3.10/site-packag
es, outside environment /local_disk0/.ephemeral_nfs/envs/pythonEnv-2df73871-
3175-4be4-8c95-414efb8964ca
    Can't uninstall 'click'. No files were found to uninstall.
```

```
  Attempting uninstall: blinker
    Found existing installation: blinker 1.4
    Not uninstalling blinker at /usr/lib/python3/dist-packages, outside envi
ronment /local_disk0/.ephemeral_nfs/envs/pythonEnv-2df73871-3175-4be4-8c95-4
14efb8964ca
    Can't uninstall 'blinker'. No files were found to uninstall.
  Attempting uninstall: Jinja2
    Found existing installation: Jinja2 2.11.3
    Not uninstalling jinja2 at /databricks/python3/lib/python3.10/site-packa
ges, outside environment /local_disk0/.ephemeral_nfs/envs/pythonEnv-2df73871
-3175-4be4-8c95-414efb8964ca
    Can't uninstall 'Jinja2'. No files were found to uninstall.
Successfully installed Flask-3.0.0 Jinja2-3.1.2 Mako-1.3.0 MarkupSafe-2.1.3
Werkzeug-3.0.1 alembic-1.12.1 blinker-1.7.0 click-8.1.7 cloudpickle-2.2.1 da
tabricks-cli-0.18.0 docker-6.1.3 gitdb-4.0.11 gitpython-3.1.40 greenlet-3.0.
1 gunicorn-21.2.0 itsdangerous-2.1.2 markdown-3.5.1 mlflow-2.8.1 pyyaml-6.0.
1 querystring-parser-1.2.4 smmap-5.0.1 sqlalchemy-2.0.23 sqlparse-0.4.4 tabu
late-0.9.0 websocket-client-1.6.4

[notice] A new release of pip available: 22.2.2 -> 23.3.1
[notice] To update, run: pip install --upgrade pip
```

In [ ]:

```
import pandas as pd

data = pd.read_csv("/dbfs/FileStore/shared_uploads/mz246@duke.edu/resume.csv
```

In [ ]:

```
data = data.iloc[:, -16:].drop(["firstname"], axis=1)
data.head(5)
```

Out[ ]:

	received_callback	race	gender	years_college	college_degree	honors	worked_d
0	0	white	f	4	1	0	
1	0	white	f	3	0	0	
2	0	black	f	4	1	0	
3	0	black	f	3	0	0	
4	0	white	f	3	0	0	

In [ ]:

```
data["race"] = data["race"].apply(lambda x: 1 if x == "white" else 0)
data["resume_quality"] = data["resume_quality"].apply(lambda x: 1 if x == "h
data["gender"] = data["gender"].apply(lambda x: 1 if x == "f" else 0)
data.head(10)
```

Out[ ]:

	received_callback	race	gender	years_college	college_degree	honors	worked_du
0	0	1	1	4	1	0	
1	0	1	1	3	0	0	
2	0	0	1	4	1	0	
3	0	0	1	3	0	0	
4	0	1	1	3	0	0	
5	0	1	0	4	1	1	
6	0	1	1	4	1	0	
7	0	0	1	3	0	0	
8	0	0	1	4	1	0	
9	0	0	0	4	1	0	

Preprocess data

Prior to training a model, check for missing values and split the data into training and validation sets.

In [ ]:
```
data.isna().any()
```

There are no missing values.

Prepare dataset for training baseline model

Split the input data into 3 sets:

- Train (60% of the dataset used to train the model)
- Validation (20% of the dataset used to tune the hyperparameters)
- Test (20% of the dataset used to report the true performance of the model on an unseen dataset)

In [ ]:
```
from sklearn.model_selection import train_test_split

X = data.drop(["received_callback"], axis=1)
y = data.received_callback

# Split out the training data
X_train, X_rem, y_train, y_rem = train_test_split(
    X, y, train_size=0.6, random_state=123
)

# Split the remaining data equally into validation and test
X_val, X_test, y_val, y_test = train_test_split(
```

```
    X_rem, y_rem, test_size=0.5, random_state=123
)
```

Build a baseline model

This task seems well suited to a random forest classifier, since the output is binary and there may be interactions between multiple variables.

The following code builds a simple classifier using scikit-learn. It uses MLflow to keep track of the model accuracy, and to save the model for later use.

In [ ]:
```
import os

os.environ["MLFLOW_TRACKING_URI"] = "http://localhost:5000"
```

In [ ]:
```
import mlflow
import mlflow.pyfunc
import mlflow.sklearn
import numpy as np
import sklearn
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_auc_score
from mlflow.models.signature import infer_signature
from mlflow.utils.environment import _mlflow_conda_env
import cloudpickle
import time

mlflow.sklearn.autolog()

# The predict method of sklearn's RandomForestClassifier returns a binary cl
# The following code creates a wrapper function, SklearnModelWrapper, that u
# the predict_proba method to return the probability that the observation be


class SklearnModelWrapper(mlflow.pyfunc.PythonModel):
    def __init__(self, model):
        self.model = model

    def predict(self, context, model_input):
        return self.model.predict_proba(model_input)[:, 1]


# mlflow.start_run creates a new MLflow run to track the performance of this
# Within the context, you call mlflow.log_param to keep track of the paramet
# mlflow.log_metric to record metrics like accuracy.
# mlflow.create_experiment("mzExperiment")

with mlflow.start_run(
    run_name="untuned_random_forest",
    experiment_id=mlflow.get_experiment_by_name("mzExperiment").experiment_i
):
    n_estimators = 10
    model = RandomForestClassifier(
```

```
        n_estimators=n_estimators, random_state=np.random.RandomState(123)
    )
    model.fit(X_train, y_train)

    # predict_proba returns [prob_negative, prob_positive], so slice the out
    predictions_test = model.predict_proba(X_test)[:, 1]
    auc_score = roc_auc_score(y_test, predictions_test)
    mlflow.log_param("n_estimators", n_estimators)
    # Use the area under the ROC curve as a metric.
    mlflow.log_metric("auc", auc_score)
    wrappedModel = SklearnModelWrapper(model)
    # Log the model with a signature that defines the schema of the model's
    # When the model is deployed, this signature will be used to validate in
    signature = infer_signature(X_train, wrappedModel.predict(None, X_train)

    # MLflow contains utilities to create a conda environment used to serve
    # The necessary dependencies are added to a conda.yaml file which is log
    conda_env = _mlflow_conda_env(
        additional_conda_deps=None,
        additional_pip_deps=[
            "cloudpickle=={}".format(cloudpickle.__version__),
            "scikit-learn=={}".format(sklearn.__version__),
        ],
        additional_conda_channels=None,
    )
    mlflow.pyfunc.log_model(
        "random_forest_model",
        python_model=wrappedModel,
        conda_env=conda_env,
        signature=signature,
    )
```

```
2023/11/24 02:17:56 WARNING mlflow.utils.autologging_utils: MLflow autologgi
ng encountered a warning: "/local_disk0/.ephemeral_nfs/envs/pythonEnv-2df738
71-3175-4be4-8c95-414efb8964ca/lib/python3.10/site-packages/mlflow/data/pand
as_dataset.py:134: UserWarning: Hint: Inferred schema contains integer colum
n(s). Integer columns in Python cannot represent missing values. If your inp
ut data contains missing values at inference time, it will be encoded as flo
ats and will cause a schema enforcement error. The best way to avoid this pr
oblem is to infer the model schema based on a realistic data sample (trainin
g dataset) that includes missing values. Alternatively, you can declare inte
ger columns as doubles (float64) whenever these columns may have missing val
ues. See `Handling Integers With Missing Values <https://www.mlflow.org/doc
s/latest/models.html#handling-integers-with-missing-values>`_ for more detai
ls."
2023/11/24 02:17:57 WARNING mlflow.utils.autologging_utils: MLflow autologgi
ng encountered a warning: "/local_disk0/.ephemeral_nfs/envs/pythonEnv-2df738
71-3175-4be4-8c95-414efb8964ca/lib/python3.10/site-packages/mlflow/models/si
gnature.py:212: UserWarning: Hint: Inferred schema contains integer column
(s). Integer columns in Python cannot represent missing values. If your inpu
t data contains missing values at inference time, it will be encoded as floa
ts and will cause a schema enforcement error. The best way to avoid this pro
blem is to infer the model schema based on a realistic data sample (training
dataset) that includes missing values. Alternatively, you can declare intege
r columns as doubles (float64) whenever these columns may have missing value
s. See `Handling Integers With Missing Values <https://www.mlflow.org/docs/l
atest/models.html#handling-integers-with-missing-values>`_ for more detail
s."
2023/11/24 02:17:59 WARNING mlflow.utils.autologging_utils: MLflow autologgi
ng encountered a warning: "/databricks/python/lib/python3.10/site-packages/_
distutils_hack/__init__.py:33: UserWarning: Setuptools is replacing distutil
s."
2023/11/24 02:17:59 WARNING mlflow.utils.autologging_utils: MLflow autologgi
ng encountered a warning: "/local_disk0/.ephemeral_nfs/envs/pythonEnv-2df738
71-3175-4be4-8c95-414efb8964ca/lib/python3.10/site-packages/mlflow/data/pand
as_dataset.py:134: UserWarning: Hint: Inferred schema contains integer colum
n(s). Integer columns in Python cannot represent missing values. If your inp
ut data contains missing values at inference time, it will be encoded as flo
ats and will cause a schema enforcement error. The best way to avoid this pr
oblem is to infer the model schema based on a realistic data sample (trainin
g dataset) that includes missing values. Alternatively, you can declare inte
ger columns as doubles (float64) whenever these columns may have missing val
ues. See `Handling Integers With Missing Values <https://www.mlflow.org/doc
s/latest/models.html#handling-integers-with-missing-values>`_ for more detai
ls."
2023/11/24 02:17:59 WARNING mlflow.utils.autologging_utils: MLflow autologgi
ng encountered a warning: "/local_disk0/.ephemeral_nfs/envs/pythonEnv-2df738
71-3175-4be4-8c95-414efb8964ca/lib/python3.10/site-packages/mlflow/data/pand
as_dataset.py:134: UserWarning: Hint: Inferred schema contains integer colum
n(s). Integer columns in Python cannot represent missing values. If your inp
ut data contains missing values at inference time, it will be encoded as flo
ats and will cause a schema enforcement error. The best way to avoid this pr
oblem is to infer the model schema based on a realistic data sample (trainin
g dataset) that includes missing values. Alternatively, you can declare inte
ger columns as doubles (float64) whenever these columns may have missing val
ues. See `Handling Integers With Missing Values <https://www.mlflow.org/doc
s/latest/models.html#handling-integers-with-missing-values>`_ for more detai
ls."
```

```
/local_disk0/.ephemeral_nfs/envs/pythonEnv-2df73871-3175-4be4-8c95-414efb896
4ca/lib/python3.10/site-packages/mlflow/models/signature.py:212: UserWarnin
g: Hint: Inferred schema contains integer column(s). Integer columns in Pyth
on cannot represent missing values. If your input data contains missing valu
es at inference time, it will be encoded as floats and will cause a schema e
nforcement error. The best way to avoid this problem is to infer the model s
chema based on a realistic data sample (training dataset) that includes miss
ing values. Alternatively, you can declare integer columns as doubles (float
64) whenever these columns may have missing values. See `Handling Integers W
ith Missing Values <https://www.mlflow.org/docs/latest/models.html#handling-
integers-with-missing-values>`_ for more details.
  inputs = _infer_schema(model_input) if model_input is not None else None
```

Examine the learned feature importances output by the model as a sanity-check.

In [ ]:

```
feature_importances = pd.DataFrame(
    model.feature_importances_, index=X_train.columns.tolist(), columns=["im
)
feature_importances.sort_values("importance", ascending=False)
```

Out[ ]:

	importance
years_experience	0.424431
race	0.085594
worked_during_school	0.063321
special_skills	0.062062
gender	0.057090
volunteer	0.051227
computer_skills	0.050512
employment_holes	0.045791
years_college	0.039595
honors	0.032277
has_email_address	0.025542
college_degree	0.024520
military	0.021261
resume_quality	0.016777

In [ ]:

```
import mlflow
from mlflow.tracking import MlflowClient

# Set the value of 'run_name'
run_name = "untuned_random_forest"

# Retrieve the run ID for the run
# Assumes that you have already set the experiment ID and 'run_name' paramet
search_results = mlflow.search_runs(
```

```
    experiment_ids=mlflow.get_experiment_by_name("mzExperiment").experiment_
    run_view_type=ViewType.ACTIVE_ONLY,
    filter_string=f"tags.mlflow.runName = '{run_name}'",
)
run_id = search_results.loc[search_results["tags.mlflow.runName"] == run_nam
    "run_id"
].iloc[0]

# Retrieve the AUC metric value from the run
run = mlflow.get_run(run_id)
auc = run.data.metrics["auc"]
print(auc)
```

```
0.5325833586703153
```

Register the model in MLflow Model Registry

By registering this model in Model Registry, you can easily reference the model from anywhere within Databricks.

The following section shows how to do this programmatically, but you can also register a model using the UI. See "Create or register a model using the UI" (AWS|Azure|GCP).

In [ ]:
```
model_name = "resume"
model_version = mlflow.register_model(f"runs:/{run_id}/random_forest_model",

# Registering the model takes a few seconds, so add a small delay
time.sleep(15)
```

```
Successfully registered model 'resume'.
2023/11/24 02:29:06 INFO mlflow.store.model_registry.abstract_store: Waiting
up to 300 seconds for model version to finish creation. Model name: resume,
version 1
Created version '1' of model 'resume'.
```

In [ ]:
```
from mlflow.tracking import MlflowClient

client = MlflowClient()
client.transition_model_version_stage(
    name=model_name,
    version=model_version.version,
    stage="Production",
)
```

Out[ ]:
```
<ModelVersion: aliases=[], creation_timestamp=1700792946796, current_stage
='Production', description='', last_updated_timestamp=1700793013666, name
='resume', run_id='243bb223af0c4bc682493a54ab9eecec', run_link='', source
='mlflow-artifacts:/438529569510626367/243bb223af0c4bc682493a54ab9eecec/art
ifacts/random_forest_model', status='READY', status_message='', tags={}, us
er_id='', version='1'>
```

The Models page now shows the model version in stage "Production".

In [ ]:
```
model = mlflow.pyfunc.load_model(f"models:/{model_name}/production")
```

```
# Sanity-check: This should match the AUC logged by MLflow
print(f"AUC: {roc_auc_score(y_test, model.predict(X_test))}")
```

```
2023/11/24 02:30:23 WARNING mlflow.utils.autologging_utils: MLflow autologgi
ng encountered a warning: "/local_disk0/.ephemeral_nfs/envs/pythonEnv-2df738
71-3175-4be4-8c95-414efb8964ca/lib/python3.10/site-packages/mlflow/data/pand
as_dataset.py:134: UserWarning: Hint: Inferred schema contains integer colum
n(s). Integer columns in Python cannot represent missing values. If your inp
ut data contains missing values at inference time, it will be encoded as flo
ats and will cause a schema enforcement error. The best way to avoid this pr
oblem is to infer the model schema based on a realistic data sample (trainin
g dataset) that includes missing values. Alternatively, you can declare inte
ger columns as doubles (float64) whenever these columns may have missing val
ues. See `Handling Integers With Missing Values <https://www.mlflow.org/doc
s/latest/models.html#handling-integers-with-missing-values>`_ for more detai
ls."
AUC: 0.5325833586703153
```

Experiment with a new model

The random forest model performed well even without hyperparameter tuning.

The following code uses the xgboost library to train a more accurate model. It runs a parallel hyperparameter sweep to train multiple models in parallel, using Hyperopt and SparkTrials. As before, the code tracks the performance of each parameter configuration with MLflow.

In [ ]:
```
from hyperopt import fmin, tpe, hp, SparkTrials, Trials, STATUS_OK
from hyperopt.pyll import scope
from math import exp
import mlflow.xgboost
import numpy as np
import xgboost as xgb

search_space = {
    "max_depth": scope.int(hp.quniform("max_depth", 4, 100, 1)),
    "learning_rate": hp.loguniform("learning_rate", -3, 0),
    "reg_alpha": hp.loguniform("reg_alpha", -5, -1),
    "reg_lambda": hp.loguniform("reg_lambda", -6, -1),
    "min_child_weight": hp.loguniform("min_child_weight", -1, 3),
    "objective": "binary:logistic",
    "seed": 123,  # Set a seed for deterministic training
}


def train_model(params):
    # With MLflow autologging, hyperparameters and the trained model are aut
    mlflow.xgboost.autolog()
    with mlflow.start_run(nested=True):
        train = xgb.DMatrix(data=X_train, label=y_train)
        validation = xgb.DMatrix(data=X_val, label=y_val)
        # Pass in the validation set so xgb can track an evaluation metric.
        # is no longer improving.
```

```
        booster = xgb.train(
            params=params,
            dtrain=train,
            num_boost_round=1000,
            evals=[(validation, "validation")],
            early_stopping_rounds=50,
        )
        validation_predictions = booster.predict(validation)
        auc_score = roc_auc_score(y_val, validation_predictions)
        mlflow.log_metric("auc", auc_score)

        signature = infer_signature(X_train, booster.predict(train))
        mlflow.xgboost.log_model(booster, "model", signature=signature)

        # Set the loss to -1*auc_score so fmin maximizes the auc_score
        return {
            "status": STATUS_OK,
            "loss": -1 * auc_score,
            "booster": booster.attributes(),
        }


# Greater parallelism will lead to speedups, but a less optimal hyperparamet
# A reasonable value for parallelism is the square root of max_evals.
spark_trials = SparkTrials(parallelism=10)

# Run fmin within an MLflow run context so that each hyperparameter configur
# run called "xgboost_models" .
with mlflow.start_run(
    run_name="xgboost_models",
    experiment_id=mlflow.get_experiment_by_name("mzExperiment").experiment_i
):
    best_params = fmin(
        fn=train_model,
        space=search_space,
        algo=tpe.suggest,
        max_evals=96,
        trials=spark_trials,
    )
```

```
100%|██████████| 96/96 [05:54<00:00,  3.69s/trial, best loss: -0.64433172919
17314]
```

```
INFO:hyperopt-spark:Total Trials: 96: 96 succeeded, 0 failed, 0 cancelled.
```

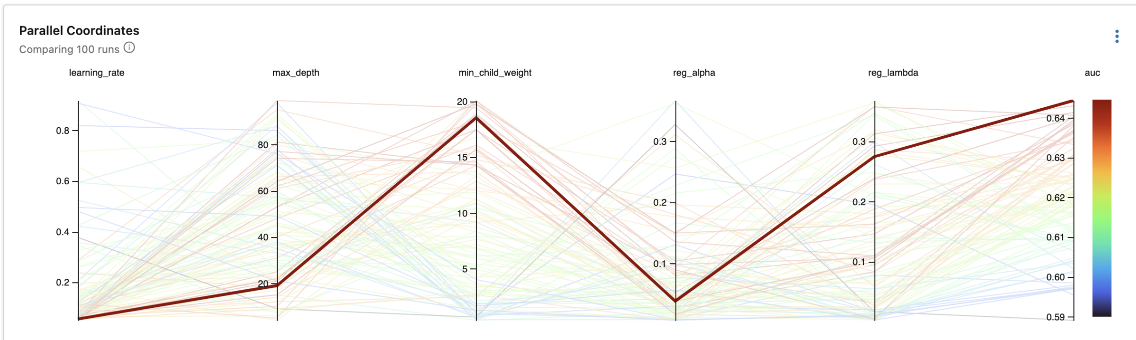
Use MLflow to view the results

Open up the Experiment Runs sidebar to see the MLflow runs. Click on Date next to the down arrow to display a menu, and select 'auc' to display the runs sorted by the auc metric. The highest auc value is 0.64.

MLflow tracks the parameters and performance metrics of each run. Click the External Link icon at the top of the Experiment Runs sidebar to navigate to the MLflow Runs Table.

Now investigate how the hyperparameter choice correlates with AUC. Click the "+" icon to expand the parent run, then select all runs except the parent, and click "Compare". Select the Parallel Coordinates Plot.

The Parallel Coordinates Plot is useful in understanding the impact of parameters on a metric. You can drag the pink slider bar at the upper right corner of the plot to highlight a subset of AUC values and the corresponding parameter values. The plot below highlights the highest AUC values:



Notice that all of the top performing runs have a low value for reg_lambda and learning_rate.

You could run another hyperparameter sweep to explore even lower values for these parameters. For simplicity, that step is not included in this example.

Update the production `resume` model in MLflow Model Registry

Earlier, you saved the baseline model to Model Registry with the name `resume`. Now that you have a created a more accurate model, update `resume`.

In [ ]:
```
new_model_version = mlflow.register_model(f"runs:/{best_run.run_id}/model",

# Registering the model takes a few seconds, so add a small delay
time.sleep(15)
```

```
Registered model 'resume' already exists. Creating a new version of this mod
el...
2023/11/24 03:28:12 INFO mlflow.store.model_registry.abstract_store: Waiting
up to 300 seconds for model version to finish creation. Model name: resume,
version 2
Created version '2' of model 'resume'.
```

Click **Models** in the left sidebar to see that the `resume` model now has two versions.

The following code promotes the new version to production.

In [ ]:
```
# Archive the old model version
client.transition_model_version_stage(
    name=model_name, version=model_version.version, stage="Archived"
```

```
)

# Promote the new model version to Production
client.transition_model_version_stage(
    name=model_name, version=new_model_version.version, stage="Production"
)
```

Out[ ]:

```
<ModelVersion: aliases=[], creation_timestamp=1700796492421, current_stage
='Production', description='', last_updated_timestamp=1700796514375, name
='resume', run_id='243bb223af0c4bc682493a54ab9eecec', run_link='', source
='mlflow-artifacts:/438529569510626367/243bb223af0c4bc682493a54ab9eecec/art
ifacts/model', status='READY', status_message='', tags={}, user_id='', vers
ion='2'>
```

Clients that call load_model now receive the new model.

The auc value on the test set for the new model is 0.64. It beat the baseline!