{"cells": [{"cell_type": "markdown", "metadata": {}, "source": "![svg image](data:image/svg+xml,%3Csvg%20version%3D%221.1%22%20id%3D%22Layer_1%22%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsvg%22%20xmlns%3Axlink%3D%22http%3A%2F%2Fwww.w3.org%2F1999%2Fxlink%22%20x%3D%220px%22%20y%3D%220px%22%0A%09%20viewBox%3D%220%200%201796%20100%22%20style%3D%22enable-background%3Anew%200%200%201796%20100%3B%22%20xml%3Aspace%3D%22preserve%22%3E%0A%3Cstyle%20type%3D%22text%2Fcss%22%3E%0A%09.st0%7Bfill-rule%3Aevenodd%3Bclip-rule%3Aevenodd%3Bfill%3Aurl%28%23SVGID_1_%29%3B%7D%0A%09.st1%7Bclip-path%3Aurl%28%23SVGID_00000055697549078573754210000016361062423753190026_%29%3B%7D%0A%09.st2%7Bfill%3Aurl%28%23SVGID_00000035492356454411262960000012525147512689374345_%29%3B%7D%0A%09.st3%7Bclip-path%3Aurl%28%23SVGID_00000044865169841396490120000008248100562427721118_%29%3B%7D%0A%09.st4%7Bfill%3Aurl%28%23SVGID_00000075884984259626911740000005845437875848140460_%29%3B%7D%0A%09.st5%7Bfill%3Anone%3Bstroke%3A%23FFFFFF%3Bstroke-width%3A2%3Bstroke-miterlimit%3A10%3B%7D%0A%09.st6%7Bfill%3Anone%3Bstroke%3A%23FFFFFF%3Bstroke-width%3A1.5%3Bstroke-miterlimit%3A10%3B%7D%0A%09.st7%7Bopacity%3A0.2%3Bfill%3Aurl%28%23SVGID_000000412537076412841479800000043741000267420215 12_%29%3Benable-background%3Anew%20%20%20%20%3B%7D%0A%09.st8%7Bfill%3A%23FFFFFF%3B%7D%0A%09.st9%7Bfont-family%3A%27IBMPlexSans-Medium%27%3B%7D%0A%09.st10%7Bfont-size%3A32px%3B%7D%0A%09.st11%7Bfont-family%3A%27IBMPlexSans%27%3B%7D%0A%09.st12%7Bfill%3A%233D3D3D%3B%7D%0A%09.st13%7Bfill%3A%23939598%3B%7D%0A%3C%2Fstyle%3E%0A%3Crect%20width%3D%221796%22%20height%3D%22100%22%2F%3E%0A%3ClinearGradient%20id%3D%22SVGID_1_%22%20gradientUnits%3D%22userSpaceOnUse%22%20x1%3D%2242.8625%22%20y1%3D%22-925.998%22%20x2%3D%2279.71%22%20y2%3D%22-925.998%22%20gradientTransform%3D%22matrix%281%200%200%20-1%200%20-876%29%22%3E%0A%09%3Cstop%20%20offset%3D%220%22%20style%3D%22stop-color%3A%23AB74FF%22%2F%3E%0A%09%3Cstop%20%20offset%3D%220.21%22%20style%3D%22stop-color%3A%23866AFF%22%2F%3E%0A%09%3Cstop%20%20offset%3D%220.75%22%20style%3D%22stop-color%3A%232A4FFF%22%2F%3E%0A%09%3Cstop%20%20offset%3D%221%22%20style%3D%22stop-color%3A%230645FF%22%2F%3E%0A%3C%2FlinearGradient%3E%0A%3Cpath%20class%3D%22st0%22%20d%3D%22M52.4%2C45.9c0-2.3%2C1.8-4.1%2C4.1-4.1s4.1%2C1.8%2C4.1%2C4.1S58.8%2C50%2C56.5%2C50l0%2C0c-2.2%2C0.1-4-1.7-4.1-3.9%0A%09C52.4%2C46%2C52.4%2C46%2C52.4%2C45.9z%20M77.5%2C52.5c-0.8-1.1-1.4-2.3-1.9-3.5c1.2-4.5%2C0.7-8.6-1.8-11.9c-2.9-3.8-8.2-6-14.5-6.1%0A%09c-4.5-0.1-8.8%2C1.7-12%2C4.8c-3%2C3-4.6%2C7.2-4.5%2C11.5c-0.1%2C2.9%2C0.9%2C5.8%2C2.7%2C8.1c0.8%2C0.8%2C1.3%2C1.9%2C1.4%2C3v4.5c-0.8%2C0.5-1.4%2C1.3-1.4%2C2.3%0A%09c0.2%2C1.5%2C1.5%2C2.6%2C3%2C2.4c1.2-0.2%2C2.2-1.1%2C2.4-2.4c0-1-0.5-1.9-1.4-2.3v-4.5c0-2-1-3.3-1.9-4.6c-1.5-1.9-2.2-4.2-2.1-6.5%0A%09c0-3.5%2C1.4-6.9%2C3.8-9.4c2.7-2.7%2C6.3-4.1%2C10-4.1c5.5%2C0%2C9.8%2C1.9%2C12.1%2C5c2%2C2.8%2C2.5%2C6.3%2C1.4%2C9.6c-0.4%2C1.2%2C0.6%2C2.7%2C2.3%2C5.6%0A%09c0.6%2C0.9%2C1.2%2C1.9%2C1.6%2C2.9c-0.9%2C0.7-2%2C1.2-3.1%2C1.5c-0.5%2C0.4-0.7%2C0.9-0.8%2C1.5V65c0%2C0.4-0.1%2C0.8-0.4%2C1.1c-0.3%2C0.2-0.7%2C0.3-1.1%2C0.3%0A%09c-1.6-0.3-3.4-0.7-5.2-1.1v-4.8c0.8-0.5%2C1.4-1.4%2C1.4-2.3c0-1.5-1.2-2.7-2.7-2.7s-2.7%2C1.2-2.7%2C2.7c0%2C1%2C0.5%2C1.9%2C1.4%2C2.3v4.1%0A%09c-0.4-0.1-0.7-0.1-1.1-0.3c-4.5-1.1-4.5-2.6-4.5-3.4v-8.3c3.2-0.7%2C5.4-3.5%2C5.5-6.7c-0.1-3.8-3.3-6.7-7.1-6.6c-3.6%2C0.1-6.4%2C3-6.6%2C6.6%0A%09c0%2C3.2%2C2.3%2C6%2C5.5%2C6.7v8.3c0%2C2%2C0.7%2C4.6%2C6.6%2C6.1c3%2C0.8%2C6%2C1.5%2C9.1%2C1.9c0.3%2C0%2C0.6%2C0.1%2C0.8%2C0.1c1%2C0%2C1.9-0.3%2C2.6-1%0A%09c0.9-0.8%2C1.4-1.9%2C1.4-3.1v-4.5c2-0.8%2C4.1-

2%2C4.1-
3.7C79.7%2C55.9%2C79%2C54.6%2C77.5%2C52.5z%22%2F%3E%0A%3Cg%3E%0A%09%3Cg%3
E%0A%09%09%3Cg%3E%0A%09%09%09%3Cdefs%3E%0A%09%09%09%09%3Cpath%20id%3D%22S
VGID_0000002398698412522435272000000908613204010133673 7_%22%20d%3D%22M52.
4%2C45.9c0-2.3%2C1.8-4.1%2C4.1-
4.1s4.1%2C1.8%2C4.1%2C4.1%0A%09%09%09%09%09S58.8%2C50%2C56.5%2C50l0%2C0c-
2.2%2C0.1-4-1.7-4.1-
3.9C52.4%2C46%2C52.4%2C46%2C52.4%2C45.9z%20M77.5%2C52.5c-0.8-1.1-1.4-2.3-
1.9-3.5%0A%09%09%09%09%09c1.2-4.5%2C0.7-8.6-1.8-11.9c-2.9-3.8-8.2-6-14.5-
6.1c-4.5-0.1-8.8%2C1.7-12%2C4.8c-3%2C3-4.6%2C7.2-4.5%2C11.5c-
0.1%2C2.9%2C0.9%2C5.8%2C2.7%2C8.1%0A%09%09%09%09%09c0.8%2C0.8%2C1.3%2C1.9
%2C1.4%2C3v4.5c-0.8%2C0.5-1.4%2C1.3-
1.4%2C2.3c0.2%2C1.5%2C1.5%2C2.6%2C3%2C2.4c1.2-0.2%2C2.2-1.1%2C2.4-2.4c0-
1-0.5-1.9-1.4-2.3v-4.5%0A%09%09%09%09%09c0-2-1-3.3-1.9-4.6c-1.5-1.9-2.2-
4.2-2.1-6.5c0-3.5%2C1.4-6.9%2C3.8-9.4c2.7-2.7%2C6.3-4.1%2C10-
4.1c5.5%2C0%2C9.8%2C1.9%2C12.1%2C5%0A%09%09%09%09%09c2%2C2.8%2C2.5%2C6.3%
2C1.4%2C9.6c-
0.4%2C1.2%2C0.6%2C2.7%2C2.3%2C5.6c0.6%2C0.9%2C1.2%2C1.9%2C1.6%2C2.9c-
0.9%2C0.7-2%2C1.2-3.1%2C1.5c-0.5%2C0.4-0.7%2C0.9-
0.8%2C1.5V65%0A%09%09%09%09%09c0%2C0.4-0.1%2C0.8-0.4%2C1.1c-0.3%2C0.2-
0.7%2C0.3-1.1%2C0.3c-1.6-0.3-3.4-0.7-5.2-1.1v-4.8c0.8-0.5%2C1.4-
1.4%2C1.4-2.3c0-1.5-1.2-2.7-2.7-2.7%0A%09%09%09%09%09s-2.7%2C1.2-
2.7%2C2.7c0%2C1%2C0.5%2C1.9%2C1.4%2C2.3v4.1c-0.4-0.1-0.7-0.1-1.1-0.3c-
4.5-1.1-4.5-2.6-4.5-3.4v-8.3c3.2-0.7%2C5.4-3.5%2C5.5-
6.7%0A%09%09%09%09%09c-0.1-3.8-3.3-6.7-7.1-6.6c-3.6%2C0.1-6.4%2C3-
6.6%2C6.6c0%2C3.2%2C2.3%2C6%2C5.5%2C6.7v8.3c0%2C2%2C0.7%2C4.6%2C6.6%2C6.1
c3%2C0.8%2C6%2C1.5%2C9.1%2C1.9%0A%09%09%09%09%09c0.3%2C0%2C0.6%2C0.1%2C0.
8%2C0.1c1%2C0%2C1.9-0.3%2C2.6-1c0.9-0.8%2C1.4-1.9%2C1.4-3.1v-4.5c2-
0.8%2C4.1-2%2C4.1-
3.7C79.7%2C55.9%2C79%2C54.6%2C77.5%2C52.5z%22%2F%3E%0A%09%09%09%3C%2Fdefs
%3E%0A%09%09%09%3CclipPath%20id%3D%22SVGID_0000000669954212366935514000001
17780821947155755945_%22%3E%0A%09%09%09%09%3Cuse%20xlink%3Ahref%3D%22%23S
VGID_0000002398698412522435272000000908613204010133673 7_%22%20%20style%3D
%22overflow%3Avisible%3B%22%2F%3E%0A%09%09%09%3C%2FclipPath%3E%0A%09%09%0
9%3Cg%20style%3D%22clip-
path%3Aurl%28%23SVGID_0000000669954212366935514000001778082194715575594 5_
%29%3B%22%3E%0A%09%09%09%09%0A%09%09%09%09%09%3ClinearGradient%20id%3D%22
SVGID_0000002612252313662683596000000247855362260891794 6_%22%20gradientUn
its%3D%22userSpaceOnUse%22%20x1%3D%22-1341.4%22%20y1%3D%22-
703.5%22%20x2%3D%22449.7%22%20y2%3D%22-
703.5%22%20gradientTransform%3D%22matrix%281%200%200%20-1%200%20-
876%29%22%3E%0A%09%09%09%09%09%3Cstop%20%20offset%3D%220%22%20style%3D%22
stop-
color%3A%23AB74FF%22%2F%3E%0A%09%09%09%09%09%3Cstop%20%20offset%3D%220.21
%22%20style%3D%22stop-
color%3A%23866AFF%22%2F%3E%0A%09%09%09%09%09%3Cstop%20%20offset%3D%220.75
%22%20style%3D%22stop-
color%3A%232A4FFF%22%2F%3E%0A%09%09%09%09%09%3Cstop%20%20offset%3D%221%22
%20style%3D%22stop-
color%3A%230645FF%22%2F%3E%0A%09%09%09%09%3C%2FlinearGradient%3E%0A%09%09
%09%09%0A%09%09%09%09%09%3Crect%20x%3D%22-1341.4%22%20y%3D%22-
876.9%22%20style%3D%22fill%3Aurl%28%23SVGID_0000002612252313662683596000000
2478553622608917946_%29%3B%22%20width%3D%221791.1%22%20height%3D%221408
.8%22%2F%3E%0A%09%09%09%09%3Cg%3E%0A%09%09%09%09%09%3Cg%3E%0A%09%09%09%09
%09%09%3Cg%3E%0A%09%09%09%09%09%09%09%3Cdefs%3E%0A%09%09%09%09%09%09%09%0
9%0A%09%09%09%09%09%09%09%09%09%3Crect%20id%3D%22SVGID_000000924198284396
690741400000182828989766341410 93_%22%20x%3D%22-1341.4%22%20y%3D%22-
876.9%22%20width%3D%221791.1%22%20height%3D%221408.8%22%2F%3E%0A%09%09%09

%09%09%09%09%3C%2Fdefs%3E%0A%09%09%09%09%09%09%09%3CclipPath%20id%3D%22SV
GID_000001695395194405695737700000147408919242473143 26_%22%3E%0A%09%09%09
%09%09%09%09%09%3Cuse%20xlink%3Ahref%3D%22%23SVGID_000000924198284396690741
4140000018282898976634141093_%22%20%20style%3D%22overflow%3Avisible%3B%22
%2F%3E%0A%09%09%09%09%09%09%09%3C%2FclipPath%3E%0A%09%09%09%09%09%09%09%3
Cg%20style%3D%22clip-
path%3Aurl%28%23SVGID_000001695395194405695737700000147408919242473143 26_
%29%3B%22%3E%0A%09%09%09%09%09%09%09%0A%09%09%09%09%09%09%09%09%09%3Cl
inearGradient%20id%3D%22SVGID_000001068441345182106961500000016829231296 71
1618217_%22%20gradientUnits%3D%22userSpaceOnUse%22%20x1%3D%2236%22%20y1%3
D%22-926.05%22%20x2%3D%2286.5%22%20y2%3D%22-
926.05%22%20gradientTransform%3D%22matrix%281%200%200%20-1%200%20-
876%29%22%3E%0A%09%09%09%09%09%09%09%09%09%3Cstop%20%20offset%3D%220%22%2
0style%3D%22stop-
color%3A%23AB74FF%22%2F%3E%0A%09%09%09%09%09%09%09%09%09%3Cstop%20%20offs
et%3D%220.21%22%20style%3D%22stop-
color%3A%23866AFF%22%2F%3E%0A%09%09%09%09%09%09%09%09%09%3Cstop%20%20offs
et%3D%220.75%22%20style%3D%22stop-
color%3A%232A4FFF%22%2F%3E%0A%09%09%09%09%09%09%09%09%09%3Cstop%20%20offs
et%3D%221%22%20style%3D%22stop-
color%3A%230645FF%22%2F%3E%0A%09%09%09%09%09%09%09%09%3C%2FlinearGradient
%3E%0A%09%09%09%09%09%09%09%09%0A%09%09%09%09%09%09%09%09%09%3Crect%20x%3
D%2236%22%20y%3D%2224.1%22%20style%3D%22fill%3Aurl%28%23SVGID_00000106844
13451821069615000000168292312967 1618217_%29%3B%22%20width%3D%2250.5%22%20
height%3D%2251.9%22%2F%3E%0A%09%09%09%09%09%09%09%3C%2Fg%3E%0A%09%09%09%0
9%09%09%3C%2Fg%3E%0A%09%09%09%09%09%3C%2Fg%3E%0A%09%09%09%09%3C%2Fg%3E%0A
%09%09%09%3C%2Fg%3E%0A%09%09%3C%2Fg%3E%0A%09%3C%2Fg%3E%0A%3C%2Fg%3E%0A%3C
circle%20class%3D%22st5%22%20cx%3D%2256.5%22%20cy%3D%2245.9%22%20r%3D%225
.4%22%2F%3E%0A%3Ccircle%20class%3D%22st6%22%20cx%3D%2248.3%22%20cy%3D%226
5%22%20r%3D%221.6%22%2F%3E%0A%3Ccircle%20class%3D%22st6%22%20cx%3D%2264.8
%22%20cy%3D%2258.2%22%20r%3D%221.6%22%2F%3E%0A%3ClinearGradient%20id%3D%2
2SVGID_000001212603448131351913900000153474335398992562 16_%22%20gradientU
nits%3D%22userSpaceOnUse%22%20x1%3D%22773.8%22%20y1%3D%22-
926%22%20x2%3D%221796%22%20y2%3D%22-
926%22%20gradientTransform%3D%22matrix%281%200%200%20-1%200%20-
876%29%22%3E%0A%09%3Cstop%20%20offset%3D%220%22%20style%3D%22stop-
color%3A%23161616%22%2F%3E%0A%09%3Cstop%20%20offset%3D%220.52%22%20style%
3D%22stop-
color%3A%23AB74FF%22%2F%3E%0A%09%3Cstop%20%20offset%3D%220.62%22%20style%
3D%22stop-
color%3A%23866AFF%22%2F%3E%0A%09%3Cstop%20%20offset%3D%220.88%22%20style%
3D%22stop-
color%3A%232A4FFF%22%2F%3E%0A%09%3Cstop%20%20offset%3D%221%22%20style%3D%
22stop-
color%3A%230645FF%22%2F%3E%0A%3C%2FlinearGradient%3E%0A%3Crect%20x%3D%227
73.8%22%20style%3D%22opacity%3A0.2%3Bfill%3Aurl%28%23SVGID_00000121260344
8131351913900000153474335398992562 16_%29%3Benable-
background%3Anew%20%20%20%20%3B%22%20width%3D%221022.2%22%20height%3D%221
00%22%2F%3E%0A%3Ctext%20transform%3D%22matrix%281%200%200%201%201309.5752
%2059.46%29%22%20class%3D%22st8%20st9%20st10%22%3EIncremental%20learning%
20notebook%3C%2Ftext%3E%0A%3Ctext%20transform%3D%22matrix%281%200%200%201
%20101.02%2059.33%29%22%20class%3D%22st8%20st11%20st10%22%3EAutoAI%3C%2Ft
ext%3E%0A%3Crect%20x%3D%22231.1%22%20y%3D%2234%22%20class%3D%22st12%22%20
width%3D%221%22%20height%3D%2232%22%2F%3E%0A%3Ctext%20transform%3D%22matr
ix%281%200%200%201%20256.29%2059.66%29%22%20class%3D%22st13%20st11%20st10
%22%3EPart%20of%20IBM%20Watson%C2%AE%20Studio%3C%2Ftext%3E%0A%3C%2Fsvg%3E
)\n# Pipeline 9 Notebook for training continuation - AutoAI Notebook
v2.1.7\n\nConsider these tips for working with an auto-generated

notebook:\n- Notebook code generated using AutoAI will execute successfully. If you modify the notebook, we cannot guarantee it will run successfully.\n- This pipeline is optimized for the original data set. The pipeline might fail or produce sub-optimal results if used with different data. If you want to use a different data set, consider retraining the AutoAI experiment to generate a new pipeline. For more information, see Cloud Platform. \n- Before modifying the pipeline or trying to re-fit the pipeline, consider that the code converts dataframes to numpy arrays before fitting the pipeline (a current restriction of the preprocessor pipeline).\n"}, {"cell_type": "markdown", "metadata": {}, "source": "\n## Notebook content\n\nThis notebook contains code to resume and continue training an AutoAI pipeline partially trained in an AutoAI experiment. If there is additional training data, the notebook retrieves the data in batches and incrementally trains the model, then tests the model.\n\nSome familiarity with Python is helpful. This notebook uses python 3.11 and scikit-learn 1.3."}, {"cell_type": "markdown", "metadata": {"pycharm": {"name": "#%% md\n"}}, "source": "## Notebook goals\n\nThis notebook introduces new commands and demonstrates techniques to support incremental learning, including: \n\n- Data reader (read data in batches)\n- Incremental learning (`partial_fit`)\n- Pipeline evaluation\n\n## Contents\n\nThis notebook contains the following parts:\n\n**[Setup](#Setup)**
\n [Package installation](#Package-installation)
\n [AutoAI experiment metadata](#AutoAI-experiment-metadata)
\n [watsonx.ai connection](#watsonx.ai-connection)
\n**[Incremental learning](#Incremental-learning)**
\n [Get pipeline](#Get-pipeline)
\n [Read training data (DataLoader)](#Data-loader)
\n [Incrementally train pipeline model](#Continue-model-training)
\n [Test pipeline model](#Test-pipeline-model)
\n**[Store the model](#Store-the-model)**
\n**[Create online deployment](#Create-online-deployment)**
\n [Working with spaces](#Working-with-spaces)
\n**[Summary and next steps](#Summary-and-next-steps)**
\n**[Copyrights](#Copyrights)**"}, {"cell_type": "markdown", "metadata": {}, "source": "\n# Setup"}, {"cell_type": "markdown", "metadata": {}, "source": "\n## Package installation\n\nBefore you use the sample code in this notebook, install the following packages:\n - ibm-watsonx-ai,\n - autoai-libs,\n - scikit-learn,\n - snapml\n"}, {"cell_type": "code", "execution_count": null, "metadata": {"execution": {"iopub.execute_input": "2020-10-12T14:00:45.009458Z", "iopub.status.busy": "2020-10-12T14:00:45.007968Z", "iopub.status.idle": "2020-10-12T14:00:46.037702Z", "shell.execute_reply": "2020-10-12T14:00:46.038270Z"}, "pycharm": {"name": "#%%\n"}, "scrolled": true}, "outputs": [], "source": "!pip install ibm-watsonx-ai | tail -n 1\n!pip install autoai-libs~=2.0 | tail -n 1\n!pip install scikit-learn==1.3.* | tail -n 1\n!pip install -U lale~=0.8.3 | tail -n 1\n!pip install snapml==1.14.* | tail -n 1"}, {"cell_type": "markdown", "metadata": {}, "source": "\n## AutoAI experiment metadata\n\nThe following cell contains the training data connection details. \n**Note**: The connection might contain authorization credentials, so be careful when sharing the notebook."}, {"cell_type": "code", "execution_count": null, "metadata": {"execution": {"iopub.execute_input": "2020-10-12T14:00:49.797633Z", "iopub.status.busy": "2020-10-12T14:00:49.796778Z", "iopub.status.idle": "2020-10-12T14:00:57.182715Z", "shell.execute_reply": "2020-10-

12T14:00:57.183132Z"}, "pycharm": {"is_executing": true}}, "outputs": [], "source": "from ibm_watsonx_ai.helpers import DataConnection\nfrom ibm_watsonx_ai.helpers import ContainerLocation\n\ntraining_data_references = [\n   DataConnection(\n data_asset_id='a2ee2f7a-afa6-4306-8ed6-8b741b2f3842'\n),\n]\ntraining_result_reference = DataConnection(\n location=ContainerLocation(\n   path='auto_ml/023b00e1-0a46-4441-89bb-28fa4dd88a65/wml_data/94f47c52-82df-4cec-b71e-3e13c17bddf2/data/automl',\n   model_location='auto_ml/023b00e1-0a46-4441-89bb-28fa4dd88a65/wml_data/94f47c52-82df-4cec-b71e-3e13c17bddf2/data/automl/model.zip',\n training_status='auto_ml/023b00e1-0a46-4441-89bb-28fa4dd88a65/wml_data/94f47c52-82df-4cec-b71e-3e13c17bddf2/training-status.json'\n   )\n)"}, {"cell_type": "markdown", "metadata": {}, "source": "The following cell contains input parameters provided to run the AutoAI experiment in Watson Studio."}, {"cell_type": "code", "execution_count": null, "metadata": {"execution": {"iopub.execute_input": "2020-10-12T14:00:57.187305Z", "iopub.status.busy": "2020-10-12T14:00:57.186602Z", "iopub.status.idle": "2020-10-12T14:00:57.188392Z", "shell.execute_reply": "2020-10-12T14:00:57.188878Z"}, "pycharm": {"name": "#%%\n"}}, "outputs": [], "source": "experiment_metadata = dict(\n prediction_type='multiclass',\n   prediction_column='Fault Type',\n holdout_size=0.1,\n   scoring='accuracy',\n   csv_separator=',',\n random_state=33,\n   max_number_of_estimators=2,\n training_data_references=training_data_references,\n training_result_reference=training_result_reference,\n deployment_url='https://au-syd.ml.cloud.ibm.com',\n project_id='091af446-afc8-42a1-a29d-8a4f8e20845b',\n drop_duplicates=True,\n include_batched_ensemble_estimators=['BatchedTreeEnsembleClassifier(ExtraTreesClassifier)', 'BatchedTreeEnsembleClassifier(LGBMClassifier)', 'BatchedTreeEnsembleClassifier(RandomForestClassifier)', 'BatchedTreeEnsembleClassifier(SnapBoostingMachineClassifier)', 'BatchedTreeEnsembleClassifier(SnapRandomForestClassifier)', 'BatchedTreeEnsembleClassifier(XGBClassifier)'],\n   classes=['Line Breakage', 'Overheating', 'Transformer Failure'],\n feature_selector_mode='auto'\n)"}, {"cell_type": "markdown", "metadata": {}, "source": "## Set `n_jobs` parameter to the number of available CPUs"}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "import os, ast\nCPU_NUMBER = 1\nif 'RUNTIME_HARDWARE_SPEC' in os.environ:\n   CPU_NUMBER = int(ast.literal_eval(os.environ['RUNTIME_HARDWARE_SPEC'])['num_cpu'])"}, {"cell_type": "markdown", "metadata": {}, "source": "\n## watsonx.ai connection\n\nThis cell defines the credentials required to work with the watsonx.ai Runtime.\n\n**Action**: Provide the IBM Cloud apikey, For details, see [documentation](https://cloud.ibm.com/docs/account?topic=account-userapikey)."}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "import getpass\n\napi_key = getpass.getpass(\"Please enter your api key (press enter): \")"}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "from ibm_watsonx_ai import Credentials\n\ncredentials = Credentials(\n   api_key=api_key,\n url=experiment_metadata['deployment_url']\n)"}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "from ibm_watsonx_ai import APIClient\n\nclient = APIClient(credentials)\n\nif 'space_id' in experiment_metadata:\n

client.set.default_space(experiment_metadata['space_id'])\nelse:\n client.set.default_project(experiment_metadata['project_id'])\n\ntraining _data_references[0].set_client(client)"}, {"cell_type": "markdown", "metadata": {}, "source": "\n# Incremental learning"}, {"cell_type": "markdown", "metadata": {}, "source": "\n## Get pipeline\n\nDownload and save a pipeline model object from the AutoAI training job (`lale` pipeline type is used for inspection and `partial_fit` capabilities)."}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "from ibm_watsonx_ai.experiment import AutoAI\n\npipeline_optimizer = AutoAI(credentials, project_id=experiment_metadata['project_id']).runs.get_optimizer(metadata =experiment_metadata)\npipeline_model = pipeline_optimizer.get_pipeline(pipeline_name='Pipeline_9', astype='lale')"}, {"cell_type": "markdown", "metadata": {}, "source": "\n## Data loader\n\nCreate DataLoader iterator to retrieve training dataset in batches. DataLoader is `Torch` compatible (`torch.utils.data`), returning Pandas DataFrames.\n\n**Note**: If reading data results in an error, provide data as iterable reader (e.g. `read_csv()` method from Pandas with chunks). It may be necessary to use methods for initial data pre-processing like: e.g. `DataFrame.dropna()`, `DataFrame.drop_duplicates()`, `DataFrame.sample()`.\n\n```\nreader_full_data = pd.read_csv(DATA_PATH, chunksize=CHUNK_SIZE)\n```"}, {"cell_type": "markdown", "metadata": {}, "source": "Batch size in rows."}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "number_of_batch_rows = 506"}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "from ibm_watsonx_ai.data_loaders import experiment as data_loaders\nfrom ibm_watsonx_ai.data_loaders.datasets import experiment as datasets\n\ndataset = datasets.ExperimentIterableDataset(\n connection=training_data_references[0],\n   enable_sampling=False,\n experiment_metadata=experiment_metadata,\n number_of_batch_rows=number_of_batch_rows\n   )\n   \ndata_loader = data_loaders.ExperimentDataLoader(dataset=dataset)"}, {"cell_type": "markdown", "metadata": {}, "source": "\n## Continue model training\n\nIn this cell, the pipeline is incrementally fitted using data batches (via `partial_fit` calls).\n\n**Note**: If you need, you can evaluate the pipeline using custom holdout data. Provide the `X_test`, `y_test` and call `scorer` on them.\n"}, {"cell_type": "markdown", "metadata": {}, "source": "### Define scorer from the optimization metric\nThis cell constructs the cell scorer based on the experiment metadata."}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "from sklearn.metrics import get_scorer\n\nscorer = get_scorer(experiment_metadata['scoring'])"}, {"cell_type": "markdown", "metadata": {}, "source": "### Tuning the incremental learner\n\nFor the best training performance set:\n\n- `n_jobs` - to available number of CPUs."}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "pipeline_model.steps[-1][1].impl.base_ensemble.set_params(n_jobs=CPU_NUMBER)"}, {"cell_type": "markdown", "metadata": {}, "source": "### Set up a learning curve plot"}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "import matplotlib.pyplot as plt\nfrom ibm_watsonx_ai.utils.autoai.incremental import plot_learning_curve\nimport time\n\npartial_fit_scores = []\nfit_times = []"}, {"cell_type": "markdown", "metadata": {}, "source": "<a

```
id=\"test_model\"></a>\n### Fit pipeline model in batches\n\n**Tip**: If the data passed to `partial_fit` is highly imbalanced (>1:10), please consider applying the `sample_weight` parameter:\n\n```\nfrom sklearn.utils.class_weight import compute_sample_weight\n\npipeline_model.partial_fit(X_train, y_train, freeze_trained_prefix=True,\n sample_weight=compute_sample_weight('balanced', y_train))\n```
"}, {"cell_type": "markdown", "metadata": {}, "source": "**Note**: If you have a holdout/test set please provide it for better pipeline evaluation and replace X_test and y_test in the following cell.\n\n```\nfrom pandas import read_csv\ntest_df = read_csv('DATA_PATH')\n\nX_test = test_df.drop([experiment_metadata['prediction_column']], axis=1).values\ny_test = test_df[experiment_metadata['prediction_column']].values\n```\n\nIf holdout set was not provided, 30% of first training batch would be used as holdout.\n\n"}, {"cell_type": "markdown", "metadata": {}, "source": "Filter warnings for incremental training."}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "import warnings\n\nwarnings.filterwarnings('ignore')"}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "from sklearn.model_selection import train_test_split\n\nfig, axes = plt.subplots(1, 3, figsize=(18, 4))\n\nfor i, batch_df in enumerate(data_loader):\n    batch_df.dropna(subset=experiment_metadata[\"prediction_column\"], inplace=True)\n    X_train = batch_df.drop([experiment_metadata['prediction_column']], axis=1).values\n    y_train = batch_df[experiment_metadata['prediction_column']].values\n    if i==0:\n        X_train, X_test, y_train, y_test = train_test_split(X_train, y_train, test_size=0.3)\n        start_time = time.time()\n        pipeline_model = pipeline_model.partial_fit(X_train, y_train, freeze_trained_prefix=True)\n        fit_times.append(time.time() - start_time)\n        partial_fit_scores.append(scorer(pipeline_model, X_test, y_test))\n        plot_learning_curve(fig=fig, axes=axes, scores=partial_fit_scores, fit_times=fit_times)"}, {"cell_type": "markdown", "metadata": {}, "source": "<a id=\"test_model\"></a>\n## Test pipeline model"}, {"cell_type": "markdown", "metadata": {"pycharm": {"name": "#%% md\n"}}, "source": "Test the fitted pipeline (`predict`)."}, {"cell_type": "code", "execution_count": null, "metadata": {"pycharm": {"name": "#%%\n"}}, "outputs": [], "source": "pipeline_model.predict(X_test[:10])"}, {"cell_type": "markdown", "metadata": {}, "source": "<a id=\"saving\"></a>\n## Store the model\n\nIn this section you will learn how to store the incrementally trained model."}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "model_metadata = {\n client.repository.ModelMetaNames.NAME: 'P9 - Pretrained AutoAI pipeline'\n}\n\nstored_model_details = client.repository.store_model(model=pipeline_model, meta_props=model_metadata, experiment_metadata=experiment_metadata)"}, {"cell_type": "markdown", "metadata": {}, "source": "Inspect the stored model details."}, {"cell_type": "code", "execution_count": null, "metadata": {}, "outputs": [], "source": "stored_model_details"}, {"cell_type": "markdown", "metadata": {}, "source": "<a id=\"deployment\"></a>\n## Create online deployment"}, {"cell_type": "markdown", "metadata": {}, "source": "You can use the commands below to promote the model to space and create online deployment (web service).\n\n<a id=\"working_spaces\"></a>\n### Working with spaces\n\nIn this section you will specify a deployment space for organizing the
```

assets for deploying and scoring the model. If you do not have an existing space, you can use Deployment Spaces Dashboard to create a new space, following these steps:\n\n- Click **New Deployment Space**.\n- Create an empty space.\n- Select Cloud Object Storage.\n- Select watsonx.ai Runtime and press **Create**.\n- Copy `space_id` and paste it below.\n\n**Tip**: You can also use the API to prepare the space for your work. Learn more [here](https://github.com/IBM/watson-machine-learning-samples/blob/master/cloud/notebooks/python_sdk/instance-management/Space%20management.ipynb).\n\n**Info**: Below cells are `raw` type - in order to run them, change their type to `code` and run them (no need to restart the notebook). You may need to add some additional info (see the **action** below).\n\n**Action**: Assign or update space ID below.\n"}, {"cell_type": "raw", "metadata": {}, "source": "space_id = \"PUT_YOUR_SPACE_ID_HERE\"\n\nmodel_id = client.spaces.promote(asset_id=stored_model_details[\"metadata\"][\"id\"], source_project_id=experiment_metadata[\"project_id\"], target_space_id=space_id)"}, {"cell_type": "markdown", "metadata": {}, "source": "#### Prepare online deployment"}, {"cell_type": "raw", "metadata": {}, "source": "client.set.default_space(space_id)\n\ndeploy_meta = {\n client.deployments.ConfigurationMetaNames.NAME: \"Incrementally trained AutoAI pipeline\",\n client.deployments.ConfigurationMetaNames.ONLINE: {},\n }\n\ndeployment_details = client.deployments.create(artifact_uid=model_id, meta_props=deploy_meta)\ndeployment_id = client.deployments.get_id(deployment_details)"}, {"cell_type": "markdown", "metadata": {}, "source": "#### Test online deployment"}, {"cell_type": "raw", "metadata": {}, "source": "scoring_payload = {\n \"input_data\": [{\n   'values': X_test[:5]\n }]\n}\n\nclient.deployments.score(deployment_id, scoring_payload)"}, {"cell_type": "markdown", "metadata": {}, "source": "\n### Deleting deployment\nYou can delete the existing deployment by calling the `client.deployments.delete(deployment_id)` command.\nTo list the existing web services, use `client.deployments.list()`."}, {"cell_type": "markdown", "metadata": {}, "source": "\n# Summary and next steps\nYou've successfully completed this notebook!\nYou've learned how to use AutoAI pipeline definition to train the model.\nCheck out the official [AutoAI site](https://www.ibm.com/cloud/watson-studio/autoai) for more samples, tutorials, documentation, how-tos, and blog posts."}, {"cell_type": "markdown", "metadata": {}, "source": "\n### Copyrights\n\nLicensed Materials - Copyright \u00a9 2025 IBM. This notebook and its source code are released under the terms of the ILAN License. Use, duplication disclosure restricted by GSA ADP Schedule Contract with IBM Corp.\n\n**Note:** The auto-generated notebooks are subject to the International License Agreement for Non-Warranted Programs (or equivalent) and License Information document for Watson Studio Auto-generated Notebook (License Terms), such agreements located in the link below. Specifically, the Source Components and Sample Materials clause included in the License Information document for Watson Studio Auto-generated Notebook applies to the auto-generated notebooks. \n\nBy downloading, copying, accessing, or otherwise using the materials, you agree to the License Terms\n\n___"}], "metadata": {"kernelspec": {"display_name": "Python 3 (ipykernel)",

"language": "python", "name": "python3"}, "language_info":
{"codemirror_mode": {"name": "ipython", "version": 3}, "file_extension":
".py", "mimetype": "text/x-python", "name": "python",
"nbconvert_exporter": "python", "pygments_lexer": "ipython3", "version":
"3.11.5"}}, "nbformat": 4, "nbformat_minor": 4}