automl-dont-overfit

May 5, 2020

[46]:
```
from azureml.core import Workspace, Experiment, Dataset, Datastore
from azureml.train.automl import AutoMLConfig
import logging
from azureml.widgets import RunDetails
```

[9]:
```
from sklearn.model_selection import train_test_split
```

[3]:
```
ws = Workspace.from_config()
print(ws)
```

```
Performing interactive authentication. Please follow the instructions on the
terminal.
To sign in, use a web browser to open the page https://microsoft.com/devicelogin
and enter the code AHZUAT2P3 to authenticate.
Interactive authentication successfully completed.
Workspace.create(name='aml-ws',
subscription_id='[illegible]', resource_group='aml-rg')
```

[20]:
```
train_data = Dataset.get_by_name(ws, name='train_data')


test_data = Dataset.get_by_name(ws, name='test_data')


submission_df = Dataset.get_by_name(ws, name='submission_file')
```

[24]:
```
train_data.to_pandas_dataframe().head()
```

[24]:

	target	0	1	2	3	4	5	6	7	8	...	290	291	292	293	294	295	296	297	298	299
0	1.00	-1.07	-1.11	-0.62	0.38	1.09	0.47	-0.42	0.46	-0.44	...	0.22	-0.34	0.25	-0.18	0.35	0.12	0.35	0.44	0.96	-0.82
1	0.00	-0.83	0.27	1.72	1.10	1.73	-0.20	1.90	-0.27	0.56	...	-0.77	-0.73	-1.16	2.55	0.86	-1.51	0.46	-0.03	-1.93	-0.34
2	0.00	0.10	1.39	-0.73	-1.06	0.01	-0.08	-1.45	0.32	-0.62	...	-1.31	0.80	-1.00	1.54	0.57	-0.31	-0.34	-0.15	-0.65	0.72
3	1.00	-0.99	-0.92	-1.34	0.14	0.54	0.64	1.13	0.19	-0.12	...	-1.37									
4	0.00	0.81	-1.51	0.52	-0.36	-0.22	-0.96	0.33	-0.57	-0.66	...	-0.18									

```
3  1.09  0.60 -0.59 -0.65 -0.16 -0.96 -1.08  0.81  3.40
4  0.72 -1.02  1.25 -0.60 -0.45  1.75  1.44 -0.39 -0.64

[5 rows x 301 columns]
```

[13]:
```
from azureml.core.compute import ComputeTarget, AmlCompute
from azureml.core.compute_target import ComputeTargetException

# Choose a name for your CPU cluster
cpu_cluster_name = "cpu-cluster-1"

# Verify that cluster does not exist already
try:
    compute_target = ComputeTarget(workspace=ws, name=cpu_cluster_name)
    print('Found existing cluster, use it.')
except ComputeTargetException:
    compute_config = AmlCompute.
 ↪provisioning_configuration(vm_size='STANDARD_DS12_V2',
                                                            max_nodes=6)
    compute_target = ComputeTarget.create(ws, cpu_cluster_name, compute_config)

compute_target.wait_for_completion(show_output=True)
```

```
Creating
Succeeded
AmlCompute wait for completion finished

Minimum number of nodes requested have been provisioned
```

[42]:
```
label_column_name = 'target'

automl_settings = {
   "n_cross_validations": 5,
    "primary_metric": 'average_precision_score_weighted',
    "enable_early_stopping": True,
    "max_concurrent_iterations": 2,
    "experiment_timeout_hours": 0.25,
    "verbosity": logging.INFO,
}

automl_config = AutoMLConfig(task = 'classification',
                             debug_log = 'automl_errors.log',
                             compute_target = compute_target,
                             training_data = train_data,
                             label_column_name = label_column_name,
                             **automl_settings
                            )
```

[43]:
```
# choose a name for experiment
experiment_name = 'automl-dont-overfit-classification2'

experiment=Experiment(ws, experiment_name)
```

[44]:
```
remote_run = experiment.submit(automl_config, show_output = True)
```

```
Running on remote or ADB.
```

[53]:
```
remote_run
```

[53]:
```
Run(Experiment: automl-dont-overfit-classification2,
Id: AutoML_a715bb03-9b58-4d32-888e-5c9dbfede7f0,
Type: automl,
Status: Running)
```

[52]:
```
RunDetails(remote_run).show()
```

```
_AutoMLWidget(widget_settings={'childWidgetDisplay': 'popup', 'send_telemetry': False, 'log_leve
```

[54]:
```
best_run, fitted_model = remote_run.get_output()
fitted_model
```

[54]:
```
PipelineWithYTransformations(Pipeline={'memory': None, 'steps':
[('datatransformer', DataTransformer(enable_dnn=None,
enable_feature_sweeping=None,
        feature_sweeping_config=None, feature_sweeping_timeout=None,
        featurization_config=None, force_text_dnn=None,
        is_cross_validation=None, is_onnx_compatible=Non...   silent=True,
subsample=1.0, subsample_for_bin=200000,
          subsample_freq=0, verbose=-10))]},
               y_transformer={}, y_transformer_name='LabelEncoder')
```

[56]:
```
test_data.head()
```

[56]:

	0	1	2	3	4	5	6	7	8	9	...	290	\
0	-0.68	1.72	-0.74	-0.84	0.15	-1.14	0.24	0.50	-1.83	-1.38	...	-1.18	
1	-0.73	-0.25	0.06	0.05	1.15	2.46	0.84	0.72	-2.27	0.58	...	1.30	
2	1.12	1.04	1.22	1.52	0.27	-0.09	0.24	-0.53	-0.92	0.71	...	-0.86	
3	-0.93	0.21	-0.05	0.57	-1.54	-1.11	0.46	1.02	-0.21	-0.20	...	0.06	
4	-0.21	-0.56	2.64	0.85	-0.38	0.31	0.51	0.48	-1.93	-0.40	...	-0.69	

	291	292	293	294	295	296	297	298	299
0	-0.40	0.76	-0.60	0.95	-0.35	0.45	-0.82	-0.28	1.30
1	1.11	0.66	0.76	0.90	-1.61	-1.70	1.11	-0.31	-0.64
2	-0.74	0.37	0.15	0.83	-1.35	0.91	0.38	0.59	-0.91
3	-0.96	0.76	-0.21	-2.17	0.83	1.44	0.12	2.78	0.62
4	0.21	0.57	-0.94	-0.01	0.27	0.74	1.34	-0.18	1.01

```
[5 rows x 300 columns]
```

[61]:
```
submission_df['target'] = fitted_model.predict_proba(test_data)
```

[62]:
```
submission_df.to_csv("submission.csv", index=None)
```

[ ]:
```

```