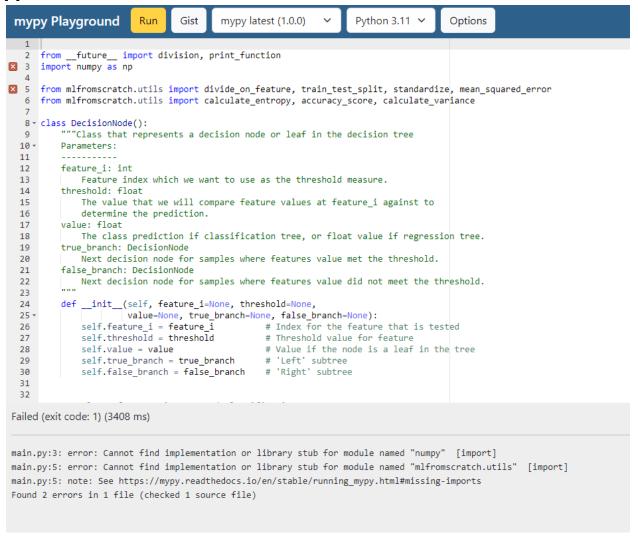**Lab 5**

Student ID: 202001132

Static Analysis of a git repository using mypy tool

Git Repositories:

https://github.com/eriklindernoren/ML-From-Scratch
https://github.com/googleapis/python-error-reporting

[1]


```
     1
     2  from __future__ import division, print_function
❌   3  import numpy as np
     4
❌   5  from mlfromscratch.utils import divide_on_feature, train_test_split, standardize, mean_squared_error
     6  from mlfromscratch.utils import calculate_entropy, accuracy_score, calculate_variance
     7
     8▾ class DecisionNode():
     9      """Class that represents a decision node or leaf in the decision tree
    10▾     Parameters:
    11      -----------
    12      feature_i: int
    13          Feature index which we want to use as the threshold measure.
    14      threshold: float
    15          The value that we will compare feature values at feature_i against to
    16          determine the prediction.
    17      value: float
    18          The class prediction if classification tree, or float value if regression tree.
    19      true_branch: DecisionNode
    20          Next decision node for samples where features value met the threshold.
    21      false_branch: DecisionNode
    22          Next decision node for samples where features value did not meet the threshold.
    23      """
    24      def __init__(self, feature_i=None, threshold=None,
    25▾                  value=None, true_branch=None, false_branch=None):
    26          self.feature_i = feature_i            # Index for the feature that is tested
    27          self.threshold = threshold           # Threshold value for feature
    28          self.value = value                   # Value if the node is a leaf in the tree
    29          self.true_branch = true_branch       # 'Left' subtree
    30          self.false_branch = false_branch     # 'Right' subtree
    31
    32
```

Failed (exit code: 1) (3408 ms)

```
main.py:3: error: Cannot find implementation or library stub for module named "numpy"  [import]
main.py:5: error: Cannot find implementation or library stub for module named "mlfromscratch.utils"  [import]
main.py:5: note: See https://mypy.readthedocs.io/en/stable/running_mypy.html#missing-imports
Found 2 errors in 1 file (checked 1 source file)
```

Missing Library - False negative
When I run this in the installed library compiler, I won't get any missing library errors.

[2]

```python
 1  import progressbar
 2  from mpl_toolkits.mplot3d import Axes3D
 3  import matplotlib.pyplot as plt
 4  import matplotlib.cm as cmx
 5  import matplotlib.colors as colors
 6  import numpy as np
 7
 8  from mlfromscratch.utils.data_operation import calculate_covariance_matrix
 9  from mlfromscratch.utils.data_operation import calculate_correlation_matrix
10  from mlfromscratch.utils.data_manipulation import standardize
11
12  bar_widgets = [
13      'Training: ', progressbar.Percentage(), ' ', progressbar.Bar(marker="-", left="[", right="]"),
14      ' ', progressbar.ETA()
15  ]
16
17  class Plot()
18      def __init__(self):
19          self.cmap = plt.get_cmap('viridis')
20
21      def _transform(self, X, dim):
22          covariance = calculate_covariance_matrix(X)
23          eigenvalues, eigenvectors = np.linalg.eig(covariance)
24          # Sort eigenvalues and eigenvector by largest eigenvalues
25          idx = eigenvalues.argsort()[::-1]
26          eigenvalues = eigenvalues[idx][:dim]
27          eigenvectors = np.atleast_1d(eigenvectors[:, idx])[:, :dim]
28          # Project the data onto principal components
29          X_transformed = X.dot(eigenvectors)
30
31          return X_transformed
32
```

Failed (exit code: 2) (1103 ms)

```
main.py:17: error: expected ':'  [syntax]
Found 1 error in 1 file (errors prevented further checking)
```
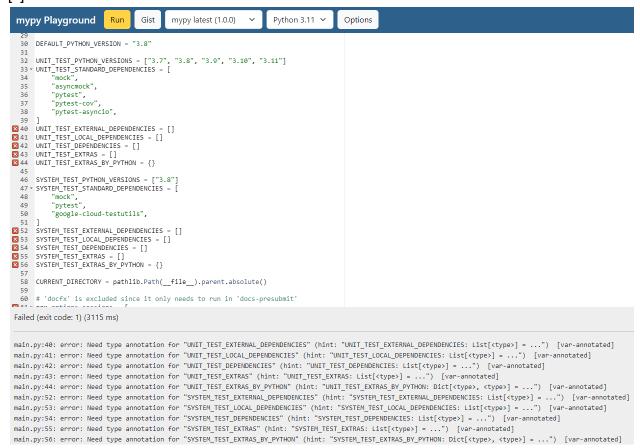
Syntax error - True positive
Expected ':' after the name of the class.

[3]

Run   Gist   mypy latest (1.0.0) ∨   Python 3.11 ∨   Options

```python
1
2  def linear_kernel(**kwargs):
3      def f(x1, x2):
4          return np.inner(x1, x2)
5      return f
6
7
8  def polynomial_kernel(power, coef, **kwargs):
9      def f(x1, x2):
10         return (np.inner(x1, x2) + coef)**power
11     return f
12
13
14 def rbf_kernel(gamma, **kwargs):
15     def f(x1, x2):
16         distance = np.linalg.norm(x1 - x2) ** 2
17         return np.exp(-gamma * distance)
18     return f
19
20 polynomial_kernel(1)
```

Failed (exit code: 1) (3146 ms)

```
main.py:20: error: Missing positional argument "coef" in call to "polynomial_kernel"  [call-arg]
Found 1 error in 1 file (checked 1 source file)
```

Call-arg error - True positive
Missing 1 argument to pass named coef.

[4]

```python
29
30    DEFAULT_PYTHON_VERSION = "3.8"
31
32    UNIT_TEST_PYTHON_VERSIONS = ["3.7", "3.8", "3.9", "3.10", "3.11"]
33    UNIT_TEST_STANDARD_DEPENDENCIES = [
34        "mock",
35        "asyncmock",
36        "pytest",
37        "pytest-cov",
38        "pytest-asyncio",
39    ]
40    UNIT_TEST_EXTERNAL_DEPENDENCIES = []
41    UNIT_TEST_LOCAL_DEPENDENCIES = []
42    UNIT_TEST_DEPENDENCIES = []
43    UNIT_TEST_EXTRAS = []
44    UNIT_TEST_EXTRAS_BY_PYTHON = {}
45
46    SYSTEM_TEST_PYTHON_VERSIONS = ["3.8"]
47    SYSTEM_TEST_STANDARD_DEPENDENCIES = [
48        "mock",
49        "pytest",
50        "google-cloud-testutils",
51    ]
52    SYSTEM_TEST_EXTERNAL_DEPENDENCIES = []
53    SYSTEM_TEST_LOCAL_DEPENDENCIES = []
54    SYSTEM_TEST_DEPENDENCIES = []
55    SYSTEM_TEST_EXTRAS = []
56    SYSTEM_TEST_EXTRAS_BY_PYTHON = {}
57
58    CURRENT_DIRECTORY = pathlib.Path(__file__).parent.absolute()
59
60    # 'docfx' is excluded since it only needs to run in 'docs-presubmit'
61    # nox options sessions = [
```

Failed (exit code: 1) (3115 ms)

```
main.py:40: error: Need type annotation for "UNIT_TEST_EXTERNAL_DEPENDENCIES" (hint: "UNIT_TEST_EXTERNAL_DEPENDENCIES: List[<type>] = ...")  [var-annotated]
main.py:41: error: Need type annotation for "UNIT_TEST_LOCAL_DEPENDENCIES" (hint: "UNIT_TEST_LOCAL_DEPENDENCIES: List[<type>] = ...")  [var-annotated]
main.py:42: error: Need type annotation for "UNIT_TEST_DEPENDENCIES" (hint: "UNIT_TEST_DEPENDENCIES: List[<type>] = ...")  [var-annotated]
main.py:43: error: Need type annotation for "UNIT_TEST_EXTRAS" (hint: "UNIT_TEST_EXTRAS: List[<type>] = ...")  [var-annotated]
main.py:44: error: Need type annotation for "UNIT_TEST_EXTRAS_BY_PYTHON" (hint: "UNIT_TEST_EXTRAS_BY_PYTHON: Dict[<type>, <type>] = ...")  [var-annotated]
main.py:52: error: Need type annotation for "SYSTEM_TEST_EXTERNAL_DEPENDENCIES" (hint: "SYSTEM_TEST_EXTERNAL_DEPENDENCIES: List[<type>] = ...")  [var-annotated]
main.py:53: error: Need type annotation for "SYSTEM_TEST_LOCAL_DEPENDENCIES" (hint: "SYSTEM_TEST_LOCAL_DEPENDENCIES: List[<type>] = ...")  [var-annotated]
main.py:54: error: Need type annotation for "SYSTEM_TEST_DEPENDENCIES" (hint: "SYSTEM_TEST_DEPENDENCIES: List[<type>] = ...")  [var-annotated]
main.py:55: error: Need type annotation for "SYSTEM_TEST_EXTRAS" (hint: "SYSTEM_TEST_EXTRAS: List[<type>] = ...")  [var-annotated]
main.py:56: error: Need type annotation for "SYSTEM_TEST_EXTRAS_BY_PYTHON" (hint: "SYSTEM_TEST_EXTRAS_BY_PYTHON: Dict[<type>, <type>] = ...")  [var-annotated]
```

Var-annotated error - True positive
Need type annotation for variable name.

[5]

```
11  # distributed under the License is distributed on an "AS IS" BASIS,
12  # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13  # See the License for the specific language governing permissions and
14  # limitations under the License.
15  #
16  import io
17  import os
18
19  import setuptools  # type: ignore
20
21  package_root = os.path.abspath(os.path.dirname(__file__))
22
23  name = "google-cloud-error-reporting"
24
25
26  description = "Google Cloud Error Reporting API client library"
27  |
28  with open(
29      os.path.join(package_root, "google/cloud/error_reporting/gapic_version.py")
30 ▾) as fp:
31      exec(fp.read(), version)
32  version = version["__version__"]
33
34 ▾if version[0] == "0":
35      release_status = "Development Status :: 4 - Beta"
36 ▾else:
37      release_status = "Development Status :: 5 - Production/Stable"
38
39 ▾dependencies = [
40      "google-cloud-logging>=1.14.0, <4.0.0dev",
41      "google-api-core[grpc] >= 1.34.0, <3.0.0dev,!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,!=2.10.*",
42      "proto-plus >= 1.22.0, <2.0.0dev",
43      "proto-plus >= 1.22.2, <2.0.0dev; python version>='3.11'",
```

Failed (exit code: 1) (3074 ms)

```
main.py:31: error: Cannot resolve name "version" (possible cyclic definition)  [misc]
main.py:31: error: Name "version" is used before definition  [used-before-def]
main.py:32: error: Cannot resolve name "version" (possible cyclic definition)  [misc]
Found 3 errors in 1 file (checked 1 source file)
```

[6]

```
mypy Playground    Run    Gist    mypy latest (1.0.0)  ∨    Python 3.11 ∨    Options

60        templated_files,
61 ▾      excludes=[
62            ".coveragerc",
63            ".github/release-please.yml",
64            ".github/auto-label.yaml",
65            "docs/index.rst",
66            "testing/constraints-3.7.txt",
67        ],
68  )
69
❌70  python.py_samples(skip_readmes)
71
72  # run format session for all directories which have a noxfile
73 ▾ for noxfile in Path(".").glob("**/noxfile.py"):
74        s.shell.run(["nox", "-s", "blacken"], cwd=noxfile.parent, hide_output=False)
75
76  # ----------------------------------------------------------------------
77  # Modify test configs
78  # ----------------------------------------------------------------------
79
80  # add shared environment variables to test configs
81  tracked_subdirs = ["continuous", "presubmit", "release", "samples", "docs"]
82 ▾ for subdir in tracked_subdirs:
83 ▾    for path, subdirs, files in os.walk(f".kokoro/{subdir}"):
84 ▾        for name in files:
85 ▾            if name == "common.cfg":
86                    file_path = os.path.join(path, name)
87                    s.move(
88                        ".kokoro/common_env_vars.cfg",
89                        file_path,
90                        merge=lambda src, dst, _,: f"{dst}\n{src}",
91                    )
```

Failed (exit code: 1) (3156 ms)

```
main.py:20: error: Cannot find implementation or library stub for module named "synthtool"  [import]
main.py:21: error: Cannot find implementation or library stub for module named "synthtool.gcp"  [import]
main.py:21: note: See https://mypy.readthedocs.io/en/stable/running_mypy.html#missing-imports
main.py:22: error: Cannot find implementation or library stub for module named "synthtool.languages"  [import]
main.py:70: error: Name "skip_readmes" is not defined  [name-defined]
Found 4 errors in 1 file (checked 1 source file)
```

Name-defined error - True positive
Name is not defined, but used, maybe a spelling mistake.