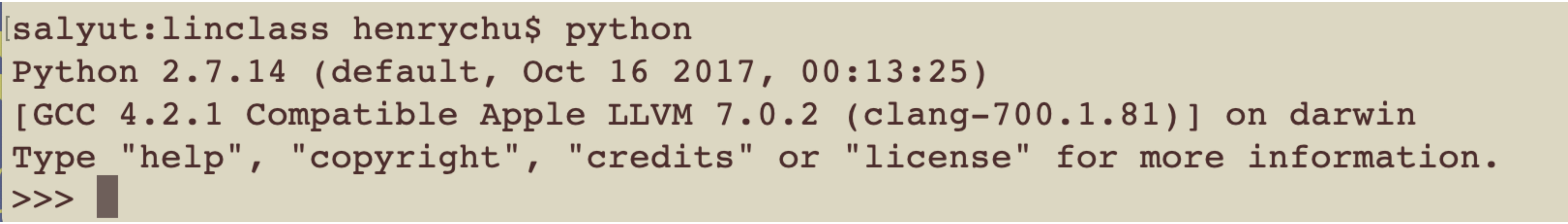
Python

- From terminal, launch Python by

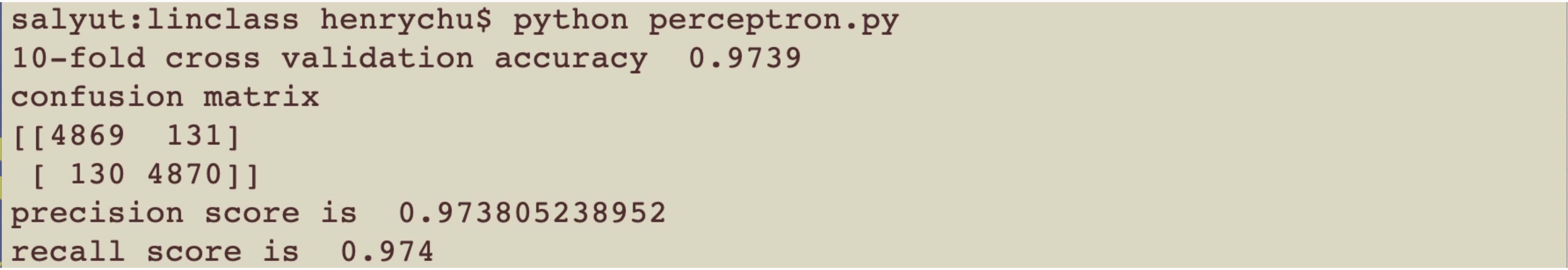


```
salyut:linclass henrychu$ python
Python 2.7.14 (default, Oct 16 2017, 00:13:25)
[GCC 4.2.1 Compatible Apple LLVM 7.0.2 (clang-700.1.81)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

- You can then type in Python calls in an interactive fashion

Python

- Alternatively, you can put your program in a file, then run python



```
salyut:linclass henrychu$ python perceptron.py
10-fold cross validation accuracy  0.9739
confusion matrix
[[4869  131]
 [ 130 4870]]
precision score is  0.973805238952
recall score is  0.974
```

Python print

- Print out variables ‘x’ and ‘y’ by

```
print x, y
```

```
import matplotlib
import matplotlib.pyplot as plt

tscores = cross_val_predict(c_all, xdata, tvec, cv=10,
method="decision_function")

from sklearn.metrics import precision_recall_curve
precisions, recalls, thresholds = precision_recall_curve(tvec,
tscores)

def plot_precision_recall_vs_threshold(precisions, recalls,
thresholds):
  plt.plot(thresholds, precisions[:-1], "b--", label="Precision")
  plt.plot(thresholds, recalls[:-1], "g-", label="Recall")
  plt.xlabel("Threshold")
  plt.legend(loc="center left")
  plt.ylim([0,1])

plot_precision_recall_vs_threshold(precisions, recalls, thresholds)
plt.show()
```

Plotting library

```
import matplotlib
import matplotlib.pyplot as plt

tscores = cross_val_predict(c_all, xdata, tvec, cv=10,
method="decision_function")

from sklearn.metrics import precision_recall_curve
precisions, recalls, thresholds = precision_recall_curve(tvec,
tscores)

def plot_precision_recall_vs_threshold(precisions, recalls,
thresholds):
  plt.plot(thresholds, precisions[:-1], "b--", label="Precision")
  plt.plot(thresholds, recalls[:-1], "g-", label="Recall")
  plt.xlabel("Threshold")
  plt.legend(loc="center left")
  plt.ylim([0,1])

plot_precision_recall_vs_threshold(precisions, recalls, thresholds)
plt.show()
```

```
import matplotlib
import matplotlib.pyplot as plt

tscores = cross_val_predict(c_all, xdata, tvec, cv=10,
method="decision_function")

from sklearn.metrics import precision_recall_curve
precisions, recalls, thresholds = precision_recall_curve(tvec,
tscores)

def plot_precision_recall_vs_threshold(precisions, recalls,
thresholds):
  plt.plot(thresholds, precisions[:-1], "b--", label="Precision")
  plt.plot(thresholds, recalls[:-1], "g-", label="Recall")
  plt.xlabel("Threshold")
  plt.legend(loc="center left")
  plt.ylim([0,1])

plot_precision_recall_vs_threshold(precisions, recalls, thresholds)
plt.show()
```

Use “decision_function” to calculate the (inner product) scores before thresholding to make a decision

```
import matplotlib
import matplotlib.pyplot as plt

tscores = cross_val_predict(c_all, xdata, tvec, cv=10,
method="decision_function")

from sklearn.metrics import precision_recall_curve
precisions, recalls, thresholds = precision_recall_curve(tvec,
tscores)

def plot_precision_recall_vs_threshold(precisions, recalls,
thresholds):
  plt.plot(thresholds, precisions[:-1], "b--", label="Precision")
  plt.plot(thresholds, recalls[:-1], "g-", label="Recall")
  plt.xlabel("Threshold")
  plt.legend(loc="center left")
  plt.ylim([0,1])

plot_precision_recall_vs_threshold(precisions, recalls, thresholds)
plt.show()
```

Library function that calculates the precision and recall values at corresponding thresholds

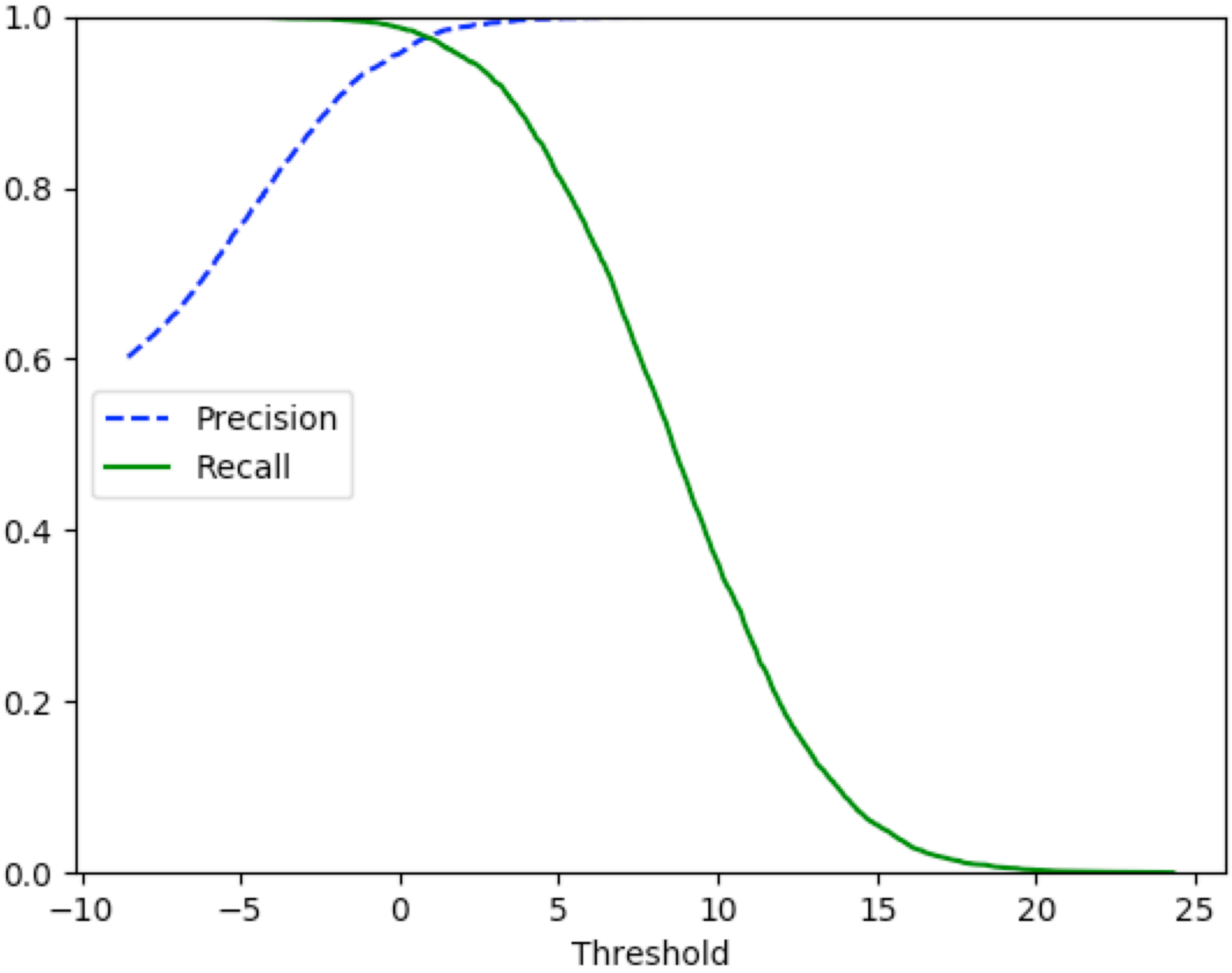
```
import matplotlib
import matplotlib.pyplot as plt

tscores = cross_val_predict(c_all, xdata, tvec, cv=10,
method="decision_function")

from sklearn.metrics import precision_recall_curve
precisions, recalls, thresholds = precision_recall_curve(tvec,
tscores)

def plot_precision_recall_vs_threshold(precisions, recalls,
thresholds):
  plt.plot(thresholds, precisions[:-1], "b--", label="Precision")
  plt.plot(thresholds, recalls[:-1], "g-", label="Recall")
  plt.xlabel("Threshold")
  plt.legend(loc="center left")
  plt.ylim([0,1])

plot_precision_recall_vs_threshold(precisions, recalls, thresholds)
plt.show()
```

Our function that plots a blue dashed (“b--”) line for precision; a green solid (“g-”) line for recall.

```
import matplotlib
import matplotlib.pyplot as plt

tscores = cross_val_predict(c_all, xdata, tvec, cv=10,
method="decision_function")

from sklearn.metrics import precision_recall_curve
precisions, recalls, thresholds = precision_recall_curve(tvec,
tscores)

def plot_precision_recall_vs_threshold(precisions, recalls,
thresholds):
  plt.plot(thresholds, precisions[:-1], "b--", label="Precision")
  plt.plot(thresholds, recalls[:-1], "g-", label="Recall")
  plt.xlabel("Threshold")
  plt.legend(loc="center left")
  plt.ylim([0,1])

plot_precision_recall_vs_threshold(precisions, recalls, thresholds)
plt.show()
```

Call the function to plot.

Show the plot.


1.0
0.8
0.6
0.4
0.2
0.0
−10
−5
0
5
10
15
20
25
Threshold
Precision
Recall