In [1]:
```
# To support both python 2 and python 3
from __future__ import division, print_function, unicode_literals
```

In [2]:
```
# Common imports
import numpy as np
import os

# to make this notebook's output stable across runs
np.random.seed(456)
```

In [3]:
```
# To plot pretty figures
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.rc('axes', labelsize=14)
mpl.rc('xtick', labelsize=12)
mpl.rc('ytick', labelsize=12)
```

In [4]:
```
# Where to save the figures
PROJECT_ROOT_DIR = "."
CHAPTER_ID = "classification"

def save_fig(fig_id, tight_layout=True):
    path = os.path.join(PROJECT_ROOT_DIR, "images", CHAPTER_ID, fig_id + ".png")
    print("Saving figure", fig_id)
    if tight_layout:
        plt.tight_layout()
    plt.savefig(path, format='png', dpi=300)
```

In [5]:
```
def sort_by_target(mnist):
    reorder_train = np.array(sorted([(target, i) for i, target in enumerate(mnist.target[:60000])]))[:, 1]
    reorder_test = np.array(sorted([(target, i) for i, target in enumerate(mnist.target[60000:])]))[:, 1]
    mnist.data[:60000] = mnist.data[reorder_train]
    mnist.target[:60000] = mnist.target[reorder_train]
    mnist.data[60000:] = mnist.data[reorder_test + 60000]
    mnist.target[60000:] = mnist.target[reorder_test + 60000]
```

In [6]:
```
try:
    from sklearn.datasets import fetch_openml
    mnist = fetch_openml('mnist_784', version=1, cache=True)
    mnist.target = mnist.target.astype(np.int8)             # fetch_openml() returns targets as strings
    sort_by_target(mnist)                                   # fetch_openml() returns an unsorted dataset
except ImportError:
    from sklearn.datasets import fetch_mldata
    mnist = fetch_mldata('MNIST original')
mnist["data"], mnist["target"]
```

Out[6]:
```
(array([[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]]),
 array([0, 0, 0, ..., 9, 9, 9], dtype=int8))
```

In [7]:
```
mnist.data.shape
```

Out[7]:
```
(70000, 784)
```

In [8]:
```
X, y = mnist["data"], mnist["target"]
X.shape
```

Out[8]:
```
(70000, 784)
```

In [9]:
```
y.shape
```

Out[9]:
```
(70000,)
```

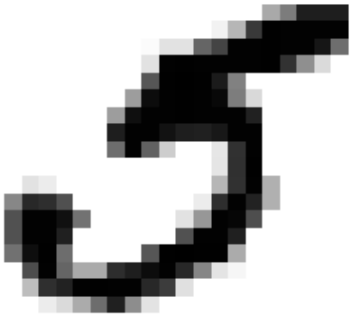
In [10]:

```
some_digit = X[36000]
some_digit_image = some_digit.reshape(28, 28)
plt.imshow(some_digit_image, cmap = mpl.cm.binary,
           interpolation="nearest")
plt.axis("off")

save_fig("some_digit_plot")
plt.show()
```

```
Saving figure some_digit_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-10-134187b971c7> in <module>
      5 plt.axis("off")
      6
----> 7 save_fig("some_digit_plot")
      8 plt.show()

<ipython-input-4-1f22c92c65f0> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: '.\\images\\classification\\some_digit_plot.png'
```



In [11]:
```
def plot_digit(data):
    image = data.reshape(28, 28)
    plt.imshow(image, cmap = mpl.cm.binary,
               interpolation="nearest")
    plt.axis("off")
```

In [12]:
```
# EXTRA
def plot_digits(instances, images_per_row=10, **options):
    size = 28
    images_per_row = min(len(instances), images_per_row)
    images = [instance.reshape(size,size) for instance in instances]
    n_rows = (len(instances) - 1) // images_per_row + 1
    row_images = []
    n_empty = n_rows * images_per_row - len(instances)
    images.append(np.zeros((size, size * n_empty)))
    for row in range(n_rows):
        rimages = images[row * images_per_row : (row + 1) * images_per_row]
        row_images.append(np.concatenate(rimages, axis=1))
    image = np.concatenate(row_images, axis=0)
    plt.imshow(image, cmap = mpl.cm.binary, **options)
    plt.axis("off")
```

In [13]:

```
plt.figure(figsize=(9,9))
example_images = np.r_[X[:12000:600], X[13000:30600:600], X[30600:60000:590]]
plot_digits(example_images, images_per_row=10)
save_fig("more_digits_plot")
plt.show()
```

```
Saving figure more_digits_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-13-93a5c63d36b3> in <module>
      2 example_images = np.r_[X[:12000:600], X[13000:30600:600], X[30600:60000:590]]
      3 plot_digits(example_images, images_per_row=10)
----> 4 save_fig("more_digits_plot")
      5 plt.show()

<ipython-input-4-1f22c92c65f0> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: '.\\images\\classification\\more_digits_plot.png'
```

In [14]:
```
X_train, X_test, y_train, y_test = X[:60000], X[60000:], y[:60000], y[60000:]
```

In [15]:
```
import numpy as np

shuffle_index = np.random.permutation(60000)
X_train, y_train = X_train[shuffle_index], y_train[shuffle_index]
```

In [16]:
```
y_train_5 = (y_train == 5)
y_test_5 = (y_test == 5)
```

In [17]:
```
from sklearn.linear_model import SGDClassifier

sgd_clf = SGDClassifier(max_iter=5, tol=-np.infty, random_state=42)
sgd_clf.fit(X_train, y_train_5)
```

Out[17]:
```
SGDClassifier(alpha=0.0001, average=False, class_weight=None,
              early_stopping=False, epsilon=0.1, eta0=0.0, fit_intercept=True,
              l1_ratio=0.15, learning_rate='optimal', loss='hinge', max_iter=5,
              n_iter_no_change=5, n_jobs=None, penalty='l2', power_t=0.5,
              random_state=42, shuffle=True, tol=-inf, validation_fraction=0.1,
              verbose=0, warm_start=False)
```

In [18]:
```
sgd_clf.predict([some_digit])
```

Out[18]:
```
array([ True])
```

In [19]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(sgd_clf, X_train, y_train_5, cv=3, scoring="accuracy")
```

Out[19]:
```
array([0.96665, 0.95625, 0.95365])
```

In [20]:
```
from sklearn.model_selection import StratifiedKFold
from sklearn.base import clone

skfolds = StratifiedKFold(n_splits=3, random_state=456)

for train_index, test_index in skfolds.split(X_train, y_train_5):
    clone_clf = clone(sgd_clf)
    X_train_folds = X_train[train_index]
    y_train_folds = (y_train_5[train_index])
    X_test_fold = X_train[test_index]
    y_test_fold = (y_train_5[test_index])

    clone_clf.fit(X_train_folds, y_train_folds)
    y_pred = clone_clf.predict(X_test_fold)
    n_correct = sum(y_pred == y_test_fold)
    print(n_correct / len(y_pred))
```

```
0.96665
0.95625
0.95365
```

In [21]:
```
from sklearn.base import BaseEstimator
class Never5Classifier(BaseEstimator):
    def fit(self, X, y=None):
        pass
    def predict(self, X):
        return np.zeros((len(X), 1), dtype=bool)
```

In [22]:
```
never_5_clf = Never5Classifier()
cross_val_score(never_5_clf, X_train, y_train_5, cv=3, scoring="accuracy")
```

Out[22]:
```
array([0.90795, 0.9124 , 0.9086 ])
```

In [23]:
```
from sklearn.model_selection import cross_val_predict

y_train_pred = cross_val_predict(sgd_clf, X_train, y_train_5, cv=3)
```

In [24]:
```
from sklearn.metrics import confusion_matrix

confusion_matrix(y_train_5, y_train_pred)
```

Out[24]:
```
array([[53757,   822],
       [ 1647,  3774]], dtype=int64)
```

In [25]:
```
y_train_perfect_predictions = y_train_5
```

In [26]:
```
confusion_matrix(y_train_5, y_train_perfect_predictions)
```

Out[26]:
```
array([[54579,     0],
       [    0,  5421]], dtype=int64)
```

In [27]:
```
from sklearn.metrics import precision_score, recall_score

precision_score(y_train_5, y_train_pred)
```

Out[27]:
```
0.8211488250652742
```

In [28]:
```
recall_score(y_train_5, y_train_pred)
```

Out[28]:
```
0.6961815163254013
```

In [29]:
```
from sklearn.metrics import f1_score
f1_score(y_train_5, y_train_pred)
```

Out[29]:
```
0.7535190176699611
```

In [30]:
```
y_scores = sgd_clf.decision_function([some_digit])
y_scores
```

Out[30]:
```
array([223648.91999263])
```

In [31]:
```
threshold = 0
y_some_digit_pred = (y_scores > threshold)
```

In [32]:
```
y_some_digit_pred
```

Out[32]:
```
array([ True])
```

In [33]:
```
threshold = 200000
y_some_digit_pred = (y_scores > threshold)
y_some_digit_pred
```

Out[33]:
```
array([ True])
```

In [34]:
```
y_scores = cross_val_predict(sgd_clf, X_train, y_train_5, cv=3,
                             method="decision_function")
```

In [35]:
```
y_scores.shape
```

Out[35]:
```
(60000,)
```

In [36]:
```
# hack to work around issue #9589 in Scikit-Learn 0.19.0
if y_scores.ndim == 2:
    y_scores = y_scores[:, 1]
```

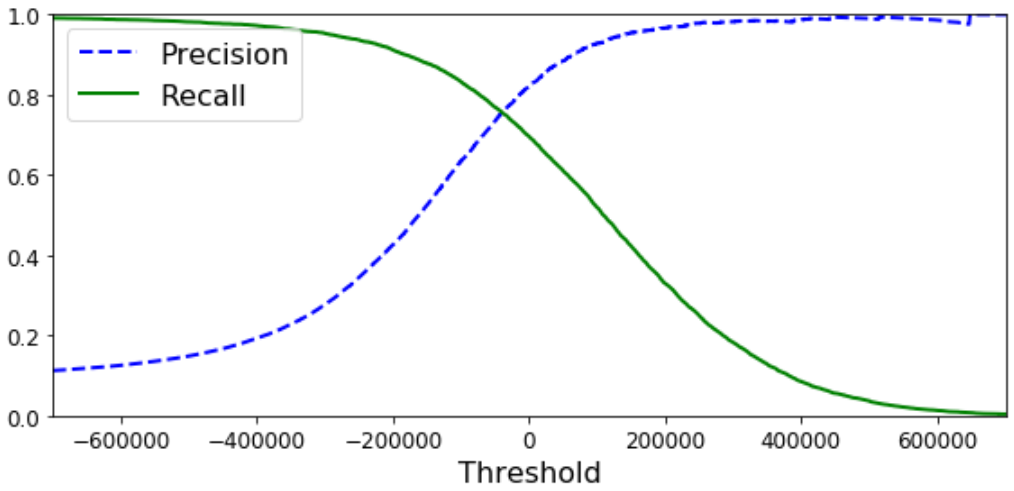
In [37]:
```
from sklearn.metrics import precision_recall_curve

precisions, recalls, thresholds = precision_recall_curve(y_train_5, y_scores)
```

In [38]:
```
def plot_precision_recall_vs_threshold(precisions, recalls, thresholds):
    plt.plot(thresholds, precisions[:-1], "b--", label="Precision", linewidth=2)
    plt.plot(thresholds, recalls[:-1], "g-", label="Recall", linewidth=2)
    plt.xlabel("Threshold", fontsize=16)
    plt.legend(loc="upper left", fontsize=16)
    plt.ylim([0, 1])

plt.figure(figsize=(8, 4))
plot_precision_recall_vs_threshold(precisions, recalls, thresholds)
plt.xlim([-700000, 700000])
save_fig("precision_recall_vs_threshold_plot")
plt.show()
```

```
Saving figure precision_recall_vs_threshold_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-38-70e383195bc9> in <module>
      9 plot_precision_recall_vs_threshold(precisions, recalls, thresholds)
     10 plt.xlim([-700000, 700000])
---> 11 save_fig("precision_recall_vs_threshold_plot")
     12 plt.show()

<ipython-input-4-1f22c92c65f0> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: '.\\images\\classification\\precision_recall_vs_threshold_plot.png'
```



In [ ]:
```
(y_train_pred == (y_scores > 0)).all()
```

In [39]:
```
y_train_pred_90 = (y_scores > 70000)
precision_score(y_train_5, y_train_pred_90)
```

Out[39]:
```
0.9055893426006372
```

In [40]:
```
recall_score(y_train_5, y_train_pred_90)
```

Out[40]:
```
0.5768308430178933
```

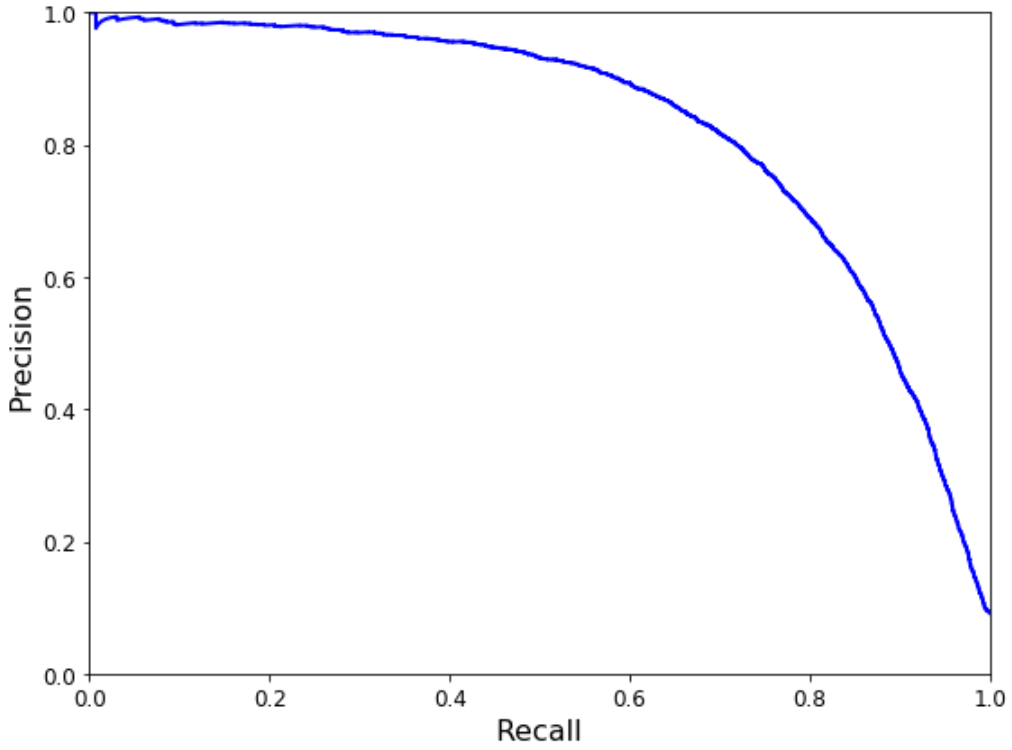
In [41]:

```
def plot_precision_vs_recall(precisions, recalls):
    plt.plot(recalls, precisions, "b-", linewidth=2)
    plt.xlabel("Recall", fontsize=16)
    plt.ylabel("Precision", fontsize=16)
    plt.axis([0, 1, 0, 1])

plt.figure(figsize=(8, 6))
plot_precision_vs_recall(precisions, recalls)
save_fig("precision_vs_recall_plot")
plt.show()
```

```
Saving figure precision_vs_recall_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-41-1d7d9eea628c> in <module>
      7 plt.figure(figsize=(8, 6))
      8 plot_precision_vs_recall(precisions, recalls)
----> 9 save_fig("precision_vs_recall_plot")
     10 plt.show()

<ipython-input-4-1f22c92c65f0> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: '.\\images\\classification\\precision_vs_recall_plot.png'
```



In [42]:
```
from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_train_5, y_scores)
```

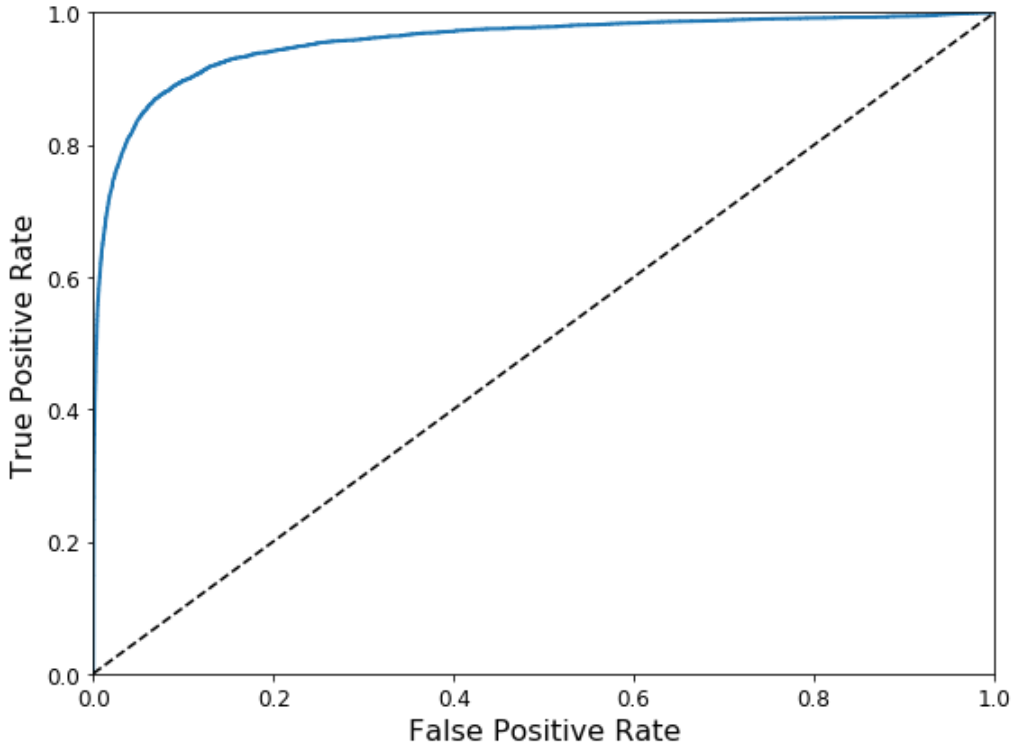
In [43]:

```
def plot_roc_curve(fpr, tpr, label=None):
    plt.plot(fpr, tpr, linewidth=2, label=label)
    plt.plot([0, 1], [0, 1], 'k--')
    plt.axis([0, 1, 0, 1])
    plt.xlabel('False Positive Rate', fontsize=16)
    plt.ylabel('True Positive Rate', fontsize=16)

plt.figure(figsize=(8, 6))
plot_roc_curve(fpr, tpr)
save_fig("roc_curve_plot")
plt.show()
```

Saving figure roc_curve_plot

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-43-21c44ac20efa> in <module>
      8 plt.figure(figsize=(8, 6))
      9 plot_roc_curve(fpr, tpr)
---> 10 save_fig("roc_curve_plot")
     11 plt.show()

<ipython-input-4-1f22c92c65f0> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: '.\\images\\classification\\roc_curve_plot.png'
```



In [44]:
```
from sklearn.metrics import roc_auc_score

roc_auc_score(y_train_5, y_scores)
```

Out[44]:
```
0.9561162168363057
```

In [45]:
```
from sklearn.ensemble import RandomForestClassifier
forest_clf = RandomForestClassifier(n_estimators=10, random_state=42)
y_probas_forest = cross_val_predict(forest_clf, X_train, y_train_5, cv=3,
                                    method="predict_proba")
```

In [46]:
```
y_scores_forest = y_probas_forest[:, 1] # score = proba of positive class
fpr_forest, tpr_forest, thresholds_forest = roc_curve(y_train_5,y_scores_forest)
```

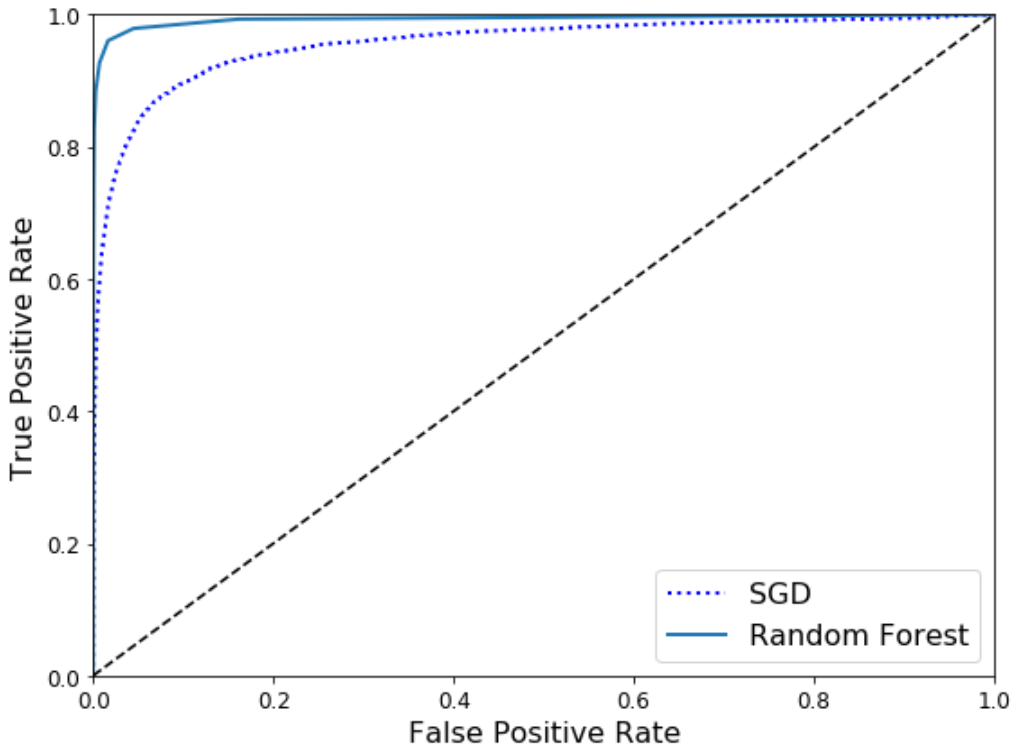
In [47]:

```
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, "b:", linewidth=2, label="SGD")
plot_roc_curve(fpr_forest, tpr_forest, "Random Forest")
plt.legend(loc="lower right", fontsize=16)
save_fig("roc_curve_comparison_plot")
plt.show()
```

```
Saving figure roc_curve_comparison_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-47-9d16666261fb> in <module>
      3 plot_roc_curve(fpr_forest, tpr_forest, "Random Forest")
      4 plt.legend(loc="lower right", fontsize=16)
----> 5 save_fig("roc_curve_comparison_plot")
      6 plt.show()

<ipython-input-4-1f22c92c65f0> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: '.\\images\\classification\\roc_curve_comparison_plot.png'
```



In [49]:
```
roc_auc_score(y_train_5, y_scores_forest)
```

Out[49]: 0.9931100348444042

In [50]:
```
y_train_pred_forest = cross_val_predict(forest_clf, X_train, y_train_5, cv=3)
precision_score(y_train_5, y_train_pred_forest)
```

Out[50]: 0.9855008787346221

In [51]:
```
recall_score(y_train_5, y_train_pred_forest)
```

Out[51]: 0.8275225973067699

In [52]:
```
sgd_clf.fit(X_train, y_train)
sgd_clf.predict([some_digit])
```

Out[52]:
```
array([5], dtype=int8)
```

In [53]:
```
some_digit_scores = sgd_clf.decision_function([some_digit])
some_digit_scores
```

Out[53]:
```
array([[-219389.21161635, -451506.81100161, -350564.46651566,
        -157194.19442459, -441237.58191287,   25546.83990702,
        -694051.98006284, -307887.46429025, -624080.017969  ,
        -758376.71717823]])
```

In [54]:
```
np.argmax(some_digit_scores)
```

Out[54]:
```
5
```

In [55]:
```
sgd_clf.classes_
```

Out[55]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], dtype=int8)
```

In [56]:
```
sgd_clf.classes_[5]
```

Out[56]:
```
5
```

In [57]:
```
from sklearn.multiclass import OneVsOneClassifier
ovo_clf = OneVsOneClassifier(SGDClassifier(max_iter=5, tol=-np.infty, random_state=42))
ovo_clf.fit(X_train, y_train)
ovo_clf.predict([some_digit])
```

Out[57]:
```
array([5], dtype=int8)
```

In [58]:
```
len(ovo_clf.estimators_)
```

Out[58]:
```
45
```

In [59]:
```
forest_clf.fit(X_train, y_train)
forest_clf.predict([some_digit])
```

Out[59]:
```
array([5], dtype=int8)
```

In [60]:
```
forest_clf.predict_proba([some_digit])
```

Out[60]:
```
array([[0. , 0. , 0. , 0.1, 0. , 0.9, 0. , 0. , 0. , 0. ]])
```

In [61]:
```
cross_val_score(sgd_clf, X_train, y_train, cv=3, scoring="accuracy")
```

Out[61]:
```
array([0.88372326, 0.87654383, 0.8586788 ])
```

In [62]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train.astype(np.float64))
cross_val_score(sgd_clf, X_train_scaled, y_train, cv=3, scoring="accuracy")
```

Out[62]:
```
array([0.90881824, 0.91324566, 0.90958644])
```

In [63]:
```
y_train_pred = cross_val_predict(sgd_clf, X_train_scaled, y_train, cv=3)
conf_mx = confusion_matrix(y_train, y_train_pred)
conf_mx
```

Out[63]:
```
array([[5735,    2,   20,   11,   10,   47,   48,    9,   37,    4],
       [   2, 6474,   44,   28,    6,   41,    6,   11,  116,   14],
       [  52,   39, 5338,   97,   86,   24,   90,   58,  157,   17],
       [  45,   42,  139, 5343,    4,  219,   35,   59,  150,   95],
       [  19,   27,   34,    6, 5376,    7,   48,   29,   85,  211],
       [  66,   40,   35,  187,   76, 4605,  113,   28,  179,   92],
       [  32,   23,   44,    2,   38,   84, 5641,    5,   49,    0],
       [  23,   21,   69,   29,   52,   11,    7, 5820,   18,  215],
       [  48,  156,   79,  153,   18,  154,   54,   23, 5035,  131],
       [  46,   25,   29,   87,  159,   31,    2,  218,   86, 5266]],
      dtype=int64)
```

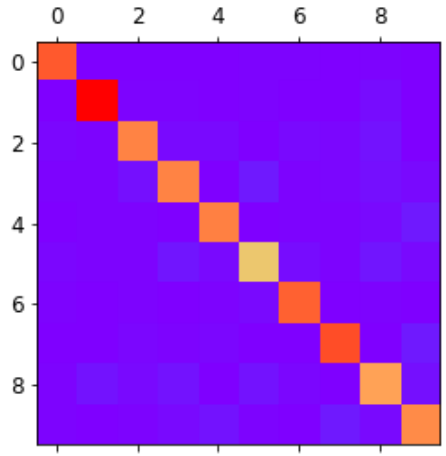
In [64]:
```
def plot_confusion_matrix(matrix):
    """If you prefer color and a colorbar"""
    fig = plt.figure(figsize=(8,8))
    ax = fig.add_subplot(111)
    cax = ax.matshow(matrix)
    fig.colorbar(cax)
```

In [65]:
```
plt.matshow(conf_mx, cmap=plt.cm.rainbow)
save_fig("confusion_matrix_plot", tight_layout=False)
plt.show()
```

```
Saving figure confusion_matrix_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-65-6970fa4afed8> in <module>
      1 plt.matshow(conf_mx, cmap=plt.cm.rainbow)
----> 2 save_fig("confusion_matrix_plot", tight_layout=False)
      3 plt.show()

<ipython-input-4-1f22c92c65f0> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
```

```
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: '.\\images\\classification\\confusion_matrix_plot.png'
```



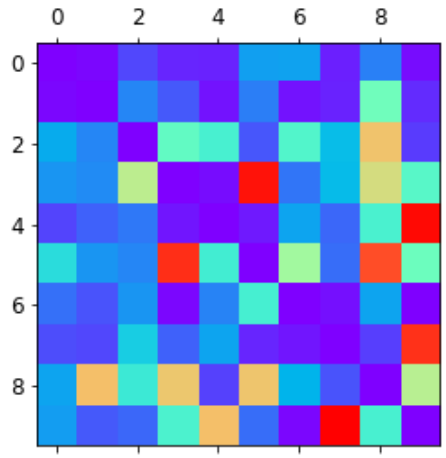
In [66]:
```
row_sums = conf_mx.sum(axis=1, keepdims=True)
norm_conf_mx = conf_mx / row_sums
```

In [67]:
```
np.fill_diagonal(norm_conf_mx, 0)
plt.matshow(norm_conf_mx, cmap=plt.cm.rainbow)
save_fig("confusion_matrix_errors_plot", tight_layout=False)
plt.show()
```

```
Saving figure confusion_matrix_errors_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-67-e53666b6173c> in <module>
      1 np.fill_diagonal(norm_conf_mx, 0)
      2 plt.matshow(norm_conf_mx, cmap=plt.cm.rainbow)
----> 3 save_fig("confusion_matrix_errors_plot", tight_layout=False)
      4 plt.show()

<ipython-input-4-1f22c92c65f0> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: '.\\images\\classification\\confusion_matrix_errors_plot.png'
```



In [68]:
```
cl_a, cl_b = 3, 5
X_aa = X_train[(y_train == cl_a) & (y_train_pred == cl_a)]
X_ab = X_train[(y_train == cl_a) & (y_train_pred == cl_b)]
X_ba = X_train[(y_train == cl_b) & (y_train_pred == cl_a)]
X_bb = X_train[(y_train == cl_b) & (y_train_pred == cl_b)]

plt.figure(figsize=(8,8))
plt.subplot(221); plot_digits(X_aa[:25], images_per_row=5)
plt.subplot(222); plot_digits(X_ab[:25], images_per_row=5)
plt.subplot(223); plot_digits(X_ba[:25], images_per_row=5)
plt.subplot(224); plot_digits(X_bb[:25], images_per_row=5)
save_fig("error_analysis_digits_plot")
plt.show()
```

```
Saving figure error_analysis_digits_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-68-c5c8191724d5> in <module>
     10 plt.subplot(223); plot_digits(X_ba[:25], images_per_row=5)
     11 plt.subplot(224); plot_digits(X_bb[:25], images_per_row=5)
---> 12 save_fig("error_analysis_digits_plot")
     13 plt.show()

<ipython-input-4-1f22c92c65f0> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: '.\\images\\classification\\error_analysis_digits_plot.png'
```

In [69]:
```
from sklearn.neighbors import KNeighborsClassifier

y_train_large = (y_train >= 7)
y_train_odd = (y_train % 2 == 1)
y_multilabel = np.c_[y_train_large, y_train_odd]

knn_clf = KNeighborsClassifier()
knn_clf.fit(X_train, y_multilabel)
```

Out[69]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

In [70]:
```
knn_clf.predict([some_digit])
```

Out[70]:
```
array([[False,  True]])
```

In [71]:
```
y_train_knn_pred = cross_val_predict(knn_clf, X_train, y_multilabel, cv=3, n_jobs=-1)
f1_score(y_multilabel, y_train_knn_pred, average="macro")
```

Out[71]:
```
0.9774437142668024
```

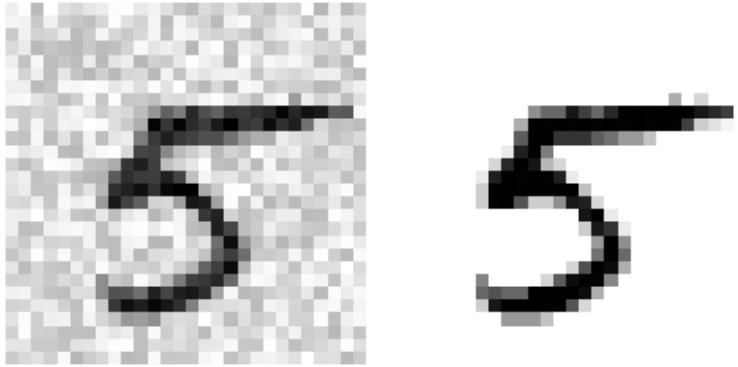
In [73]:
```
noise = np.random.randint(0, 100, (len(X_train), 784))
X_train_mod = X_train + noise
noise = np.random.randint(0, 100, (len(X_test), 784))
X_test_mod = X_test + noise
y_train_mod = X_train
y_test_mod = X_test
```

In [74]:
```
some_index = 5500
plt.subplot(121); plot_digit(X_test_mod[some_index])
plt.subplot(122); plot_digit(y_test_mod[some_index])
save_fig("noisy_digit_example_plot")
plt.show()
```

```
Saving figure noisy_digit_example_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-74-595149420a32> in <module>
      2 plt.subplot(121); plot_digit(X_test_mod[some_index])
      3 plt.subplot(122); plot_digit(y_test_mod[some_index])
----> 4 save_fig("noisy_digit_example_plot")
      5 plt.show()

<ipython-input-4-1f22c92c65f0> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: '.\\images\\classification\\noisy_digit_example_plot.png'
```

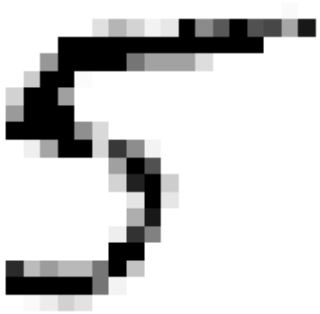


In [75]:
```
knn_clf.fit(X_train_mod, y_train_mod)
clean_digit = knn_clf.predict([X_test_mod[some_index]])
plot_digit(clean_digit)
save_fig("cleaned_digit_example_plot")
```

```
Saving figure cleaned_digit_example_plot
```

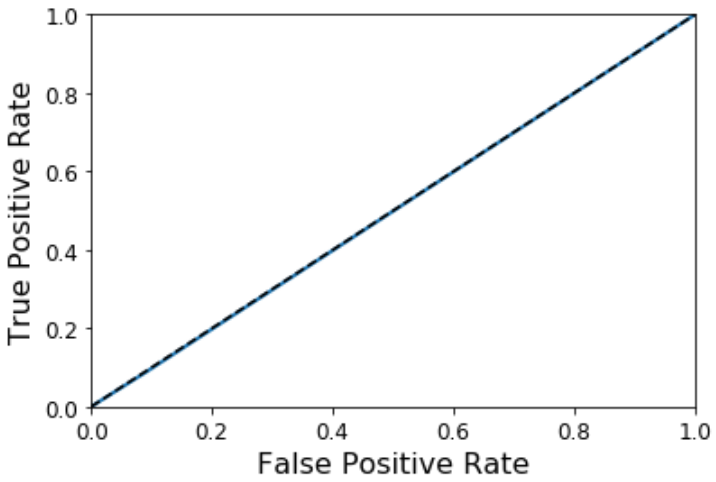
```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-75-08e0d544bdd4> in <module>
      2 clean_digit = knn_clf.predict([X_test_mod[some_index]])
      3 plot_digit(clean_digit)
----> 4 save_fig("cleaned_digit_example_plot")

<ipython-input-4-1f22c92c65f0> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
```

```
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: '.\\images\\classification\\cleaned_digit_example_plot.png'
```



In [76]:
```
from sklearn.dummy import DummyClassifier
dmy_clf = DummyClassifier()
y_probas_dmy = cross_val_predict(dmy_clf, X_train, y_train_5, cv=3, method="predict_proba")
y_scores_dmy = y_probas_dmy[:, 1]
```

In [77]:
```
fprr, tprr, thresholdsr = roc_curve(y_train_5, y_scores_dmy)
plot_roc_curve(fprr, tprr)
```



In [78]:
```
from sklearn.neighbors import KNeighborsClassifier
knn_clf = KNeighborsClassifier(n_jobs=-1, weights='distance', n_neighbors=4)
knn_clf.fit(X_train, y_train)
```

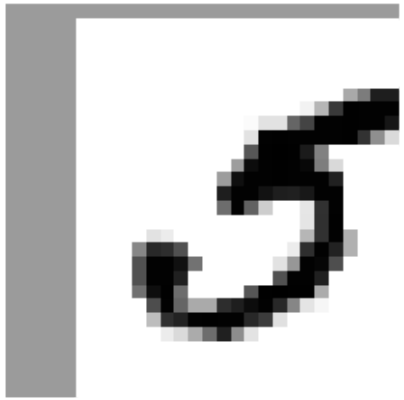
Out[78]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=-1, n_neighbors=4, p=2,
                     weights='distance')
```

In [79]:
```
y_knn_pred = knn_clf.predict(X_test)
```

In [80]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_knn_pred)
```

Out[80]:
```
0.9714
```

In [81]:
```
from scipy.ndimage.interpolation import shift
def shift_digit(digit_array, dx, dy, new=0):
    return shift(digit_array.reshape(28, 28), [dy, dx], cval=new).reshape(784)

plot_digit(shift_digit(some_digit, 5, 1, new=100))
```



In [82]:
```
X_train_expanded = [X_train]
y_train_expanded = [y_train]
for dx, dy in ((1, 0), (-1, 0), (0, 1), (0, -1)):
    shifted_images = np.apply_along_axis(shift_digit, axis=1, arr=X_train, dx=dx, dy=dy)
    X_train_expanded.append(shifted_images)
    y_train_expanded.append(y_train)

X_train_expanded = np.concatenate(X_train_expanded)
y_train_expanded = np.concatenate(y_train_expanded)
X_train_expanded.shape, y_train_expanded.shape
```

Out[82]:
```
((300000, 784), (300000,))
```

In [83]:
```
knn_clf.fit(X_train_expanded, y_train_expanded)
```

Out[83]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=-1, n_neighbors=4, p=2,
                     weights='distance')
```

In [84]:
```
y_knn_expanded_pred = knn_clf.predict(X_test)
```

In [85]:
```
accuracy_score(y_test, y_knn_expanded_pred)
```

Out[85]:
```
0.9763
```

In [86]:
```
ambiguous_digit = X_test[2589]
knn_clf.predict_proba([ambiguous_digit])
```

Out[86]:
```
array([[0.       , 0.       , 0.5053645, 0.       , 0.       , 0.       ,
        0.       , 0.4946355, 0.       , 0.       ]])
```

In [87]:
```
plot_digit(ambiguous_digit)
```