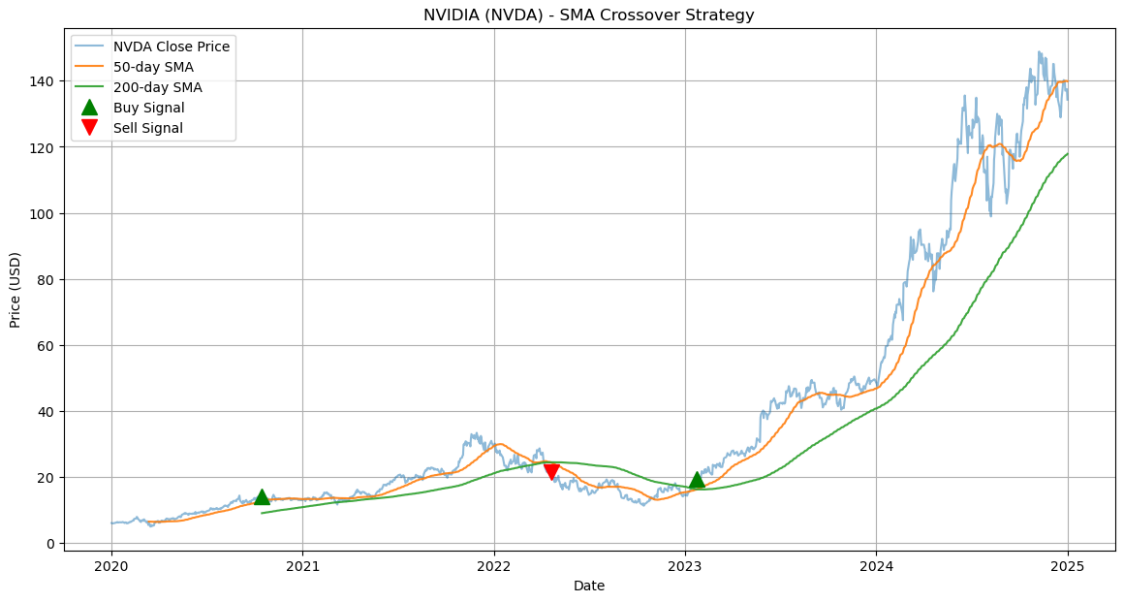
TradersSMANVIDIA

April 29, 2025

[12]:
```
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt
import numpy as np

# Download data
ticker = "NVDA"
data = yf.download(ticker, start="2020-01-01", end="2025-01-01")
data = data[['Close']]

# Calculate SMAs
data['SMA_50'] = data['Close'].rolling(window=50).mean()
data['SMA_200'] = data['Close'].rolling(window=200).mean()

# Generate signals
data['Signal'] = np.where(data['SMA_50'] > data['SMA_200'], 1, 0)
data['Position'] = data['Signal'].diff()

# Plotting
plt.figure(figsize=(14,7))
plt.plot(data['Close'], label='NVDA Close Price', alpha=0.5)
plt.plot(data['SMA_50'], label='50-day SMA', alpha=0.9)
plt.plot(data['SMA_200'], label='200-day SMA', alpha=0.9)

# Buy signals
plt.plot(data[data['Position'] == 1].index,
         data['Close'][data['Position'] == 1],
         '^', markersize=12, color='g', label='Buy Signal')

# Sell signals
plt.plot(data[data['Position'] == -1].index,
         data['Close'][data['Position'] == -1],
         'v', markersize=12, color='r', label='Sell Signal')

plt.title('NVIDIA (NVDA) - SMA Crossover Strategy')
plt.xlabel('Date')
plt.ylabel('Price (USD)')
```

```
plt.legend()
plt.grid()
plt.show()
```

```
[**********************100%**********************]  1 of 1 completed
```



1 Assistant

The error occurs because you're trying to use positional slicing with `.loc`, which is not allowed. The `.loc` indexer is label-based, so it expects labels (like dates in your DataFrame), not integer positions.

The problematic line is:

[5]:
```
data.loc[50:, 'Signal'] = np.where(data['SMA_50'][50:] > data['SMA_200'][50:],␣
 ↪1, 0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[5], line 1
----> 1 data.loc[50:, 'Signal'] = np.where(data['SMA_50'][50:] >␣
 ↪data['SMA_200'][50:], 1, 0)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:845, in _LocationIndexer.__setitem__(self, key, value
    843 else:
    844     key = com.apply_if_callable(key, self.obj)
```

```
--> 845 indexer = self._get_setitem_indexer(key)
    846 self._has_valid_setitem_indexer(key)
    848 iloc = self if self.name == "iloc" else self.obj.iloc

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:710, in _LocationIndexer._get_setitem_indexer(self,␣
 ↪key)
    707 if isinstance(key, tuple):
    708     with suppress(IndexingError):
    709         # suppress "Too many indexers"
--> 710         return self._convert_tuple(key)
    712 if isinstance(key, range):
    713     # GH#45479 test_loc_setitem_range_key
    714     key = list(key)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:927, in _LocationIndexer._convert_tuple(self, key)
    923 @final
    924 def _convert_tuple(self, key: tuple) -> tuple:
    925     # Note: we assume _tupleize_axis_indexer has been called, if␣
 ↪necessary.
    926     self._validate_key_length(key)
--> 927     keyidx = [self._convert_to_indexer(k, axis=i) for i, k in␣
 ↪enumerate(key)]
    928     return tuple(keyidx)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:927, in <listcomp>(.0)
    923 @final
    924 def _convert_tuple(self, key: tuple) -> tuple:
    925     # Note: we assume _tupleize_axis_indexer has been called, if␣
 ↪necessary.
    926     self._validate_key_length(key)
--> 927     keyidx = [self._convert_to_indexer(k, axis=i) for i, k in␣
 ↪enumerate(key)]
    928     return tuple(keyidx)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:1382, in _LocIndexer._convert_to_indexer(self, key,␣
 ↪axis)
   1379 labels = self.obj._get_axis(axis)
   1381 if isinstance(key, slice):
-> 1382     return labels._convert_slice_indexer(key, kind="loc")
   1384 if (
   1385     isinstance(key, tuple)
   1386     and not isinstance(labels, MultiIndex)
   1387     and self.ndim < 2
   1388     and len(key) > 1
```

```
   1389 ):
   1390     raise IndexingError("Too many indexers")

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexes/base.py:4126, in Index._convert_slice_indexer(self, key,␣
 ↪kind)
   4123 elif is_positional:
   4124     if kind == "loc":
   4125         # GH#16121, GH#24612, GH#31810
-> 4126         raise TypeError(
   4127             "Slicing a positional slice with .loc is not allowed, "
   4128             "Use .loc with labels or .iloc with positions instead.",
   4129         )
   4130     indexer = key
   4131 else:

TypeError: Slicing a positional slice with .loc is not allowed, Use .loc with␣
 ↪labels or .iloc with positions instead.
```

Would you like me to provide the corrected code?

2 Assistant

The error occurs because you're trying to use positional slicing with `.loc`, which is not allowed. The `.loc` indexer is label-based, so it expects labels (like dates in your DataFrame), not integer positions.

The problematic line is:

[6]:
```
data.loc[50:, 'Signal'] = np.where(data['SMA_50'][50:] > data['SMA_200'][50:],␣
 ↪1, 0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[6], line 1
----> 1 data.loc[50:, 'Signal'] = np.where(data['SMA_50'][50:] >␣
 ↪data['SMA_200'][50:], 1, 0)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:845, in _LocationIndexer.__setitem__(self, key, value
    843 else:
    844     key = com.apply_if_callable(key, self.obj)
--> 845 indexer = self._get_setitem_indexer(key)
    846 self._has_valid_setitem_indexer(key)
    848 iloc = self if self.name == "iloc" else self.obj.iloc
```

```
File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:710, in _LocationIndexer._get_setitem_indexer(self,␣
 ↪key)
    707 if isinstance(key, tuple):
    708     with suppress(IndexingError):
    709         # suppress "Too many indexers"
--> 710         return self._convert_tuple(key)
    712 if isinstance(key, range):
    713     # GH#45479 test_loc_setitem_range_key
    714     key = list(key)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:927, in _LocationIndexer._convert_tuple(self, key)
    923 @final
    924 def _convert_tuple(self, key: tuple) -> tuple:
    925     # Note: we assume _tupleize_axis_indexer has been called, if␣
 ↪necessary.
    926     self._validate_key_length(key)
--> 927     keyidx = [self._convert_to_indexer(k, axis=i) for i, k in␣
 ↪enumerate(key)]
    928     return tuple(keyidx)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:927, in <listcomp>(.0)
    923 @final
    924 def _convert_tuple(self, key: tuple) -> tuple:
    925     # Note: we assume _tupleize_axis_indexer has been called, if␣
 ↪necessary.
    926     self._validate_key_length(key)
--> 927     keyidx = [self._convert_to_indexer(k, axis=i) for i, k in␣
 ↪enumerate(key)]
    928     return tuple(keyidx)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:1382, in _LocIndexer._convert_to_indexer(self, key,␣
 ↪axis)
   1379 labels = self.obj._get_axis(axis)
   1381 if isinstance(key, slice):
-> 1382     return labels._convert_slice_indexer(key, kind="loc")
   1384 if (
   1385     isinstance(key, tuple)
   1386     and not isinstance(labels, MultiIndex)
   1387     and self.ndim < 2
   1388     and len(key) > 1
   1389 ):
   1390     raise IndexingError("Too many indexers")
```

```
File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexes/base.py:4126, in Index._convert_slice_indexer(self, key,␣
 ↪kind)
   4123 elif is_positional:
   4124     if kind == "loc":
   4125         # GH#16121, GH#24612, GH#31810
-> 4126         raise TypeError(
   4127             "Slicing a positional slice with .loc is not allowed, "
   4128             "Use .loc with labels or .iloc with positions instead.",
   4129         )
   4130     indexer = key
   4131 else:

TypeError: Slicing a positional slice with .loc is not allowed, Use .loc with␣
 ↪labels or .iloc with positions instead.
```

Would you like me to provide the corrected code?

3 Assistant

The error occurs because you're trying to use positional slicing with `.loc`, which is not allowed. The `.loc` indexer is label-based, so it expects labels (like dates in your DataFrame), not integer positions.

The problematic line is:

[7]:
```
data.loc[50:, 'Signal'] = np.where(data['SMA_50'][50:] > data['SMA_200'][50:],␣
 ↪1, 0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[7], line 1
----> 1 data.loc[50:, 'Signal'] = np.where(data['SMA_50'][50:] >␣
 ↪data['SMA_200'][50:], 1, 0)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:845, in _LocationIndexer.__setitem__(self, key, value
    843 else:
    844     key = com.apply_if_callable(key, self.obj)
--> 845 indexer = self._get_setitem_indexer(key)
    846 self._has_valid_setitem_indexer(key)
    848 iloc = self if self.name == "iloc" else self.obj.iloc

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:710, in _LocationIndexer._get_setitem_indexer(self,␣
 ↪key)
    707 if isinstance(key, tuple):
    708     with suppress(IndexingError):
```

```
    709         # suppress "Too many indexers"
--> 710         return self._convert_tuple(key)
    712 if isinstance(key, range):
    713     # GH#45479 test_loc_setitem_range_key
    714     key = list(key)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:927, in _LocationIndexer._convert_tuple(self, key)
    923 @final
    924 def _convert_tuple(self, key: tuple) -> tuple:
    925     # Note: we assume _tupleize_axis_indexer has been called, if␣
 ↪necessary.
    926     self._validate_key_length(key)
--> 927     keyidx = [self._convert_to_indexer(k, axis=i) for i, k in␣
 ↪enumerate(key)]
    928     return tuple(keyidx)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:927, in <listcomp>(.0)
    923 @final
    924 def _convert_tuple(self, key: tuple) -> tuple:
    925     # Note: we assume _tupleize_axis_indexer has been called, if␣
 ↪necessary.
    926     self._validate_key_length(key)
--> 927     keyidx = [self._convert_to_indexer(k, axis=i) for i, k in␣
 ↪enumerate(key)]
    928     return tuple(keyidx)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:1382, in _LocIndexer._convert_to_indexer(self, key,␣
 ↪axis)
   1379 labels = self.obj._get_axis(axis)
   1381 if isinstance(key, slice):
-> 1382     return labels._convert_slice_indexer(key, kind="loc")
   1384 if (
   1385     isinstance(key, tuple)
   1386     and not isinstance(labels, MultiIndex)
   1387     and self.ndim < 2
   1388     and len(key) > 1
   1389 ):
   1390     raise IndexingError("Too many indexers")

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexes/base.py:4126, in Index._convert_slice_indexer(self, key,␣
 ↪kind)
   4123 elif is_positional:
   4124     if kind == "loc":
   4125         # GH#16121, GH#24612, GH#31810
```

```
-> 4126         raise TypeError(
   4127             "Slicing a positional slice with .loc is not allowed, "
   4128             "Use .loc with labels or .iloc with positions instead.",
   4129         )
   4130     indexer = key
   4131 else:

TypeError: Slicing a positional slice with .loc is not allowed, Use .loc with␣
 ↪labels or .iloc with positions instead.
```

Would you like me to provide the corrected code?

4 Assistant

The error occurs because you're trying to use positional slicing with `.loc`, which is not allowed. The `.loc` indexer is label-based, so it expects labels (like dates in your DataFrame), not integer positions.

The problematic line is:

[8]:
```
data.loc[50:, 'Signal'] = np.where(data['SMA_50'][50:] > data['SMA_200'][50:],␣
 ↪1, 0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[8], line 1
----> 1 data.loc[50:, 'Signal'] = np.where(data['SMA_50'][50:] >␣
 ↪data['SMA_200'][50:], 1, 0)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:845, in _LocationIndexer.__setitem__(self, key, value
    843 else:
    844     key = com.apply_if_callable(key, self.obj)
--> 845 indexer = self._get_setitem_indexer(key)
    846 self._has_valid_setitem_indexer(key)
    848 iloc = self if self.name == "iloc" else self.obj.iloc

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:710, in _LocationIndexer._get_setitem_indexer(self,␣
 ↪key)
    707 if isinstance(key, tuple):
    708     with suppress(IndexingError):
    709         # suppress "Too many indexers"
--> 710         return self._convert_tuple(key)
    712 if isinstance(key, range):
    713     # GH#45479 test_loc_setitem_range_key
    714     key = list(key)
```

```
File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:927, in _LocationIndexer._convert_tuple(self, key)
    923 @final
    924 def _convert_tuple(self, key: tuple) -> tuple:
    925     # Note: we assume _tupleize_axis_indexer has been called, if␣
 ↪necessary.
    926     self._validate_key_length(key)
--> 927     keyidx = [self._convert_to_indexer(k, axis=i) for i, k in␣
 ↪enumerate(key)]
    928     return tuple(keyidx)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:927, in <listcomp>(.0)
    923 @final
    924 def _convert_tuple(self, key: tuple) -> tuple:
    925     # Note: we assume _tupleize_axis_indexer has been called, if␣
 ↪necessary.
    926     self._validate_key_length(key)
--> 927     keyidx = [self._convert_to_indexer(k, axis=i) for i, k in␣
 ↪enumerate(key)]
    928     return tuple(keyidx)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexing.py:1382, in _LocIndexer._convert_to_indexer(self, key,␣
 ↪axis)
   1379 labels = self.obj._get_axis(axis)
   1381 if isinstance(key, slice):
-> 1382     return labels._convert_slice_indexer(key, kind="loc")
   1384 if (
   1385     isinstance(key, tuple)
   1386     and not isinstance(labels, MultiIndex)
   1387     and self.ndim < 2
   1388     and len(key) > 1
   1389 ):
   1390     raise IndexingError("Too many indexers")

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
 ↪pandas/core/indexes/base.py:4126, in Index._convert_slice_indexer(self, key,␣
 ↪kind)
   4123 elif is_positional:
   4124     if kind == "loc":
   4125         # GH#16121, GH#24612, GH#31810
-> 4126         raise TypeError(
   4127             "Slicing a positional slice with .loc is not allowed, "
   4128             "Use .loc with labels or .iloc with positions instead.",
   4129         )
   4130     indexer = key
   4131 else:
```

```
TypeError: Slicing a positional slice with .loc is not allowed, Use .loc with␣
 ↪labels or .iloc with positions instead.
```

Would you like me to provide the corrected code?