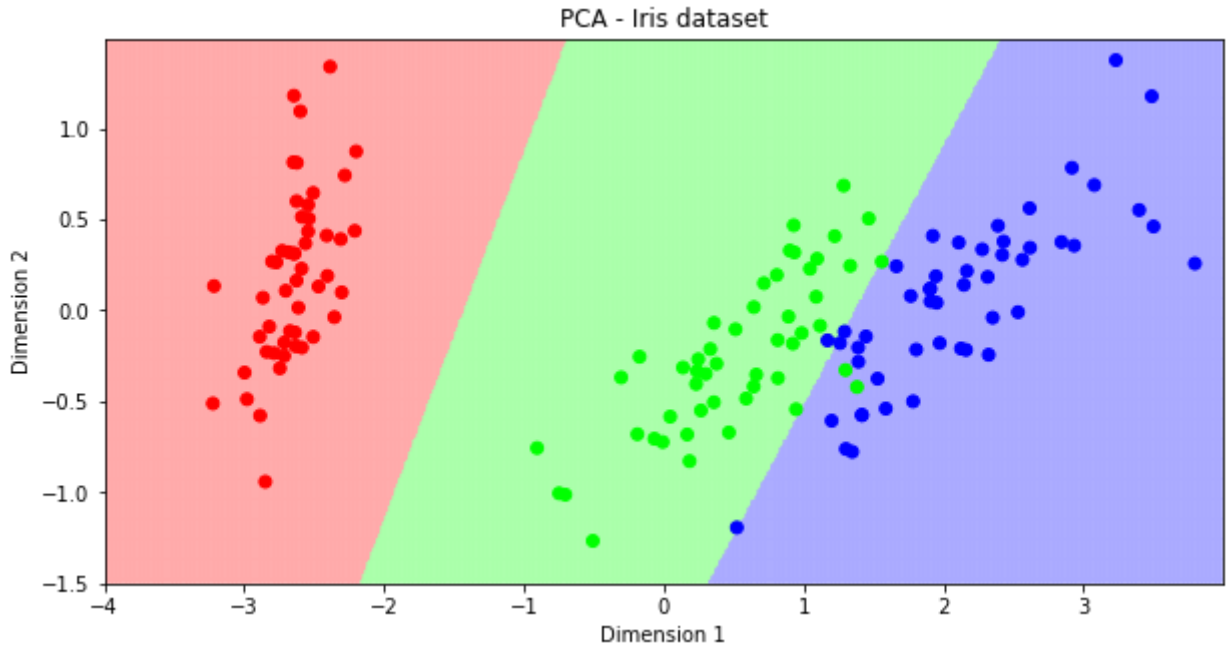
Dataset Scaling

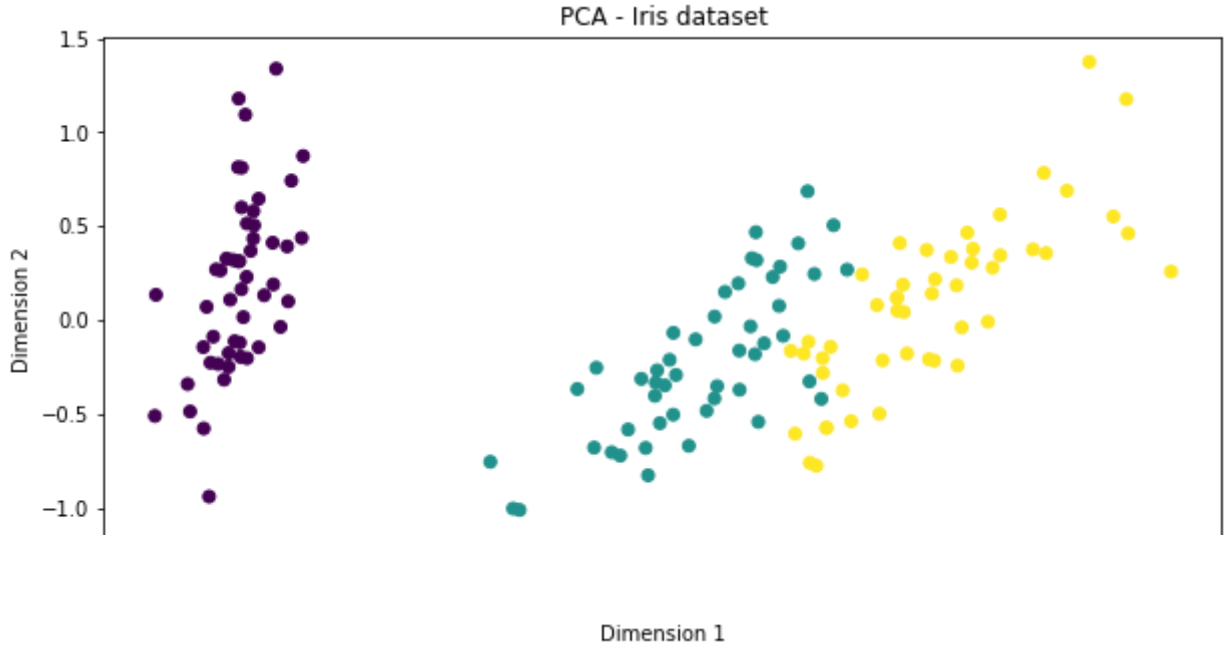
```
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
from sklearn.svm import SVC
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
import numpy as np
h = .01
x_min, x_max = -4,4
y_min, y_max = -1.5,1.5
# loading dataset
data = load_iris()
X, y = data.data, data.target
# selecting first 2 components of PCA
X_pca = PCA().fit_transform(X)
X_selected = X_pca[:,:2]
# training classifier and evaluating on the whole plane
clf = SVC(kernel='linear')
clf.fit(X_selected,y)
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                     np.arange(y_min, y_max, h))
Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
# Plotting
cmap_light = ListedColormap(['#FFAAAA', '#AAFFAA', '#AAAAFF'])
cmap_bold = ListedColormap(['#FF0000', '#00FF00', '#0000FF'])
plt.figure(figsize=(10,5))
plt.pcolormesh(xx, yy, Z, alpha=.6,cmap=cmap_light)
plt.title('PCA - Iris dataset')
plt.xlabel('Dimension 1')
plt.ylabel('Dimension 2')
plt.scatter(X_pca[:,0],X_pca[:,1],c=data.target,cmap=cmap_bold)
plt.show()
```



Feature Selection

```
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
data = load_iris()
X, y = data.data, data.target
plt.figure(figsize=(10,5))
X_pca = PCA().fit_transform(X)
plt.title('PCA - Iris dataset')
plt.xlabel('Dimension 1')
plt.ylabel('Dimension 2')
plt.scatter(X_pca[:,0],X_pca[:,1],c=data.target)
```

```
<matplotlib.collections.PathCollection at 0x7f42f7963e10>
```



-

Handling Missing Values

```
import pandas as pd
import numpy as np
nfl_data = pd.read_csv("/content/drive/MyDrive/DATA/NFL Play by Play 2009-2016 (v3).csv")
np.random.seed(0)
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2882: DtypeWarning: Columns (25,51) have mixed types.Sp
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
nfl_data.head()
```

	Date	GameID	Drive	qtr	down	time	TimeUnder	TimeSecs	PlayTimeDiff	SideofField	...	yacEPA	Home_WP_pre	Away_
0	2009-09-10	2009091000	1	1	NaN	15:00	15	3600.0	0.0	TEN	...	NaN	0.485675	0.5
1	2009-09-10	2009091000	1	1	1.0	14:53	15	3593.0	7.0	PIT	...	1.146076	0.546433	0.4
2	2009-09-10	2009091000	1	1	2.0	14:16	15	3556.0	37.0	PIT	...	NaN	0.551088	0.4
3	2009-09-10	2009091000	1	1	3.0	13:35	14	3515.0	41.0	PIT	...	-5.031425	0.510793	0.4
4	2009-09-10	2009091000	1	1	4.0	13:27	14	3507.0	8.0	PIT	...	NaN	0.461217	0.5

5 rows × 102 columns

```
missing_values_count = nfl_data.isnull().sum()
missing_values_count[0:10]
```

```
Date               0
GameID             0
Drive              0
qtr                0
down           54218
time             188
TimeUnder          0
TimeSecs         188
PlayTimeDiff     374
SideofField      450
dtype: int64
```

```
total_cells = np.product(nfl_data.shape)
total_missing = missing_values_count.sum()
```

```
percent_missing = (total_missing/total_cells) * 100
print(percent_missing)
```

```
24.85847694188906
```

```
missing_values_count[0:10]
```

```
Date               0
GameID             0
Drive              0
qtr                0
down           54218
time             188
TimeUnder          0
TimeSecs         188
PlayTimeDiff     374
SideofField      450
dtype: int64
```

```
nfl_data.dropna()
```

Date	GameID	Drive	qtr	down	time	TimeUnder	TimeSecs	PlayTimeDiff	SideofField	...	yacEPA	Home_WP_pre

0 rows × 102 columns

```
columns_with_na_dropped = nfl_data.dropna(axis=1)
columns_with_na_dropped.head()
```

	Date	GameID	Drive	qtr	TimeUnder	ydstogo	ydsnet	PlayAttempted	Yards.Gained	sp	...	Timeout_Indica
0	2009-09-10	2009091000	1	1	15	0	0	1	39	0	...	
1	2009-09-10	2009091000	1	1	15	10	5	1	5	0	...	
2	2009-09-10	2009091000	1	1	15	5	2	1	-3	0	...	
3	2009-09-10	2009091000	1	1	14	8	2	1	0	0	...	
4	2009-09-10	2009091000	1	1	14	8	2	1	0	0	...	

```
print("Columns in original dataset: %d \n" % nfl_data.shape[1])
print("Columns with na's dropped: %d" % columns_with_na_dropped.shape[1])
```

```
Columns in original dataset: 102

Columns with na's dropped: 41
```

```
#Finding Missing Values Automatically
subset_nfl_data = nfl_data.loc[:, 'EPA':'Season'].head()
subset_nfl_data
```

	EPA	airEPA	yacEPA	Home_WP_pre	Away_WP_pre	Home_WP_post	Away_WP_post	Win_Prob	WPA	air
0	2.014474	NaN	NaN	0.485675	0.514325	0.546433	0.453567	0.485675	0.060758	N
1	0.077907	-1.068169	1.146076	0.546433	0.453567	0.551088	0.448912	0.546433	0.004655	-0.032
2	-1.402760	NaN	NaN	0.551088	0.448912	0.510793	0.489207	0.551088	-0.040295	N
3	-1.712583	3.318841	-5.031425	0.510793	0.489207	0.461217	0.538783	0.510793	-0.049576	0.106
4	2.097796	NaN	NaN	0.461217	0.538783	0.558929	0.441071	0.461217	0.097712	N

```
subset_nfl_data.fillna(0)
```

	EPA	airEPA	yacEPA	Home_WP_pre	Away_WP_pre	Home_WP_post	Away_WP_post	Win_Prob	WPA	air
0	2.014474	0.000000	0.000000	0.485675	0.514325	0.546433	0.453567	0.485675	0.060758	0.000
1	0.077907	-1.068169	1.146076	0.546433	0.453567	0.551088	0.448912	0.546433	0.004655	-0.032
2	-1.402760	0.000000	0.000000	0.551088	0.448912	0.510793	0.489207	0.551088	-0.040295	0.000
3	-1.712583	3.318841	-5.031425	0.510793	0.489207	0.461217	0.538783	0.510793	-0.049576	0.106
4	2.097796	0.000000	0.000000	0.461217	0.538783	0.558929	0.441071	0.461217	0.097712	0.000

```
subset_nfl_data.fillna(method='bfill', axis=0).fillna(0)
```

	EPA	airEPA	yacEPA	Home_WP_pre	Away_WP_pre	Home_WP_post	Away_WP_post	Win_Prob	WPA	air
0	2.014474	-1.068169	1.146076	0.485675	0.514325	0.546433	0.453567	0.485675	0.060758	-0.032
1	0.077907	-1.068169	1.146076	0.546433	0.453567	0.551088	0.448912	0.546433	0.004655	-0.032
2	-1.402760	3.318841	-5.031425	0.551088	0.448912	0.510793	0.489207	0.551088	-0.040295	0.106
3	-1.712583	3.318841	-5.031425	0.510793	0.489207	0.461217	0.538783	0.510793	-0.049576	0.106
4	2.097796	0.000000	0.000000	0.461217	0.538783	0.558929	0.441071	0.461217	0.097712	0.000