In [1]:
```python
import numpy as np
import pandas as pd
```

In [2]:
```python
import os
print(os.listdir("C:/Users/ganes/Desktop/6th sem/Nikith"))
```

```
['Bank Personal Loan.xlsx', 'code.py', 'Machine-Learning-ML-model-for-predicting-Mushroom-is-eatable-or-not.zip', 'Machine-Learning-ML-model-to-predict-House-pricing.zip']
```

In [3]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
get_ipython().run_line_magic('matplotlib', 'inline')
```

In [4]:
```python
file= pd.ExcelFile('C:/Users/ganes/Desktop/6th sem/Nikith/Bank Personal Loan.xlsx')
```

In [5]:
```python
description=pd.read_excel(file, 'Description')
df=pd.read_excel(file, 'Data')
```

In [6]:
```python
description.head(10)
```

Out[6]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2
0	NaN	NaN	NaN
1	NaN	NaN	NaN
2	NaN	NaN	NaN
3	NaN	NaN	NaN
4	NaN	Data Description:	NaN
5	NaN	NaN	NaN
6	NaN	ID	Customer ID
7	NaN	Age	Customer's age in completed years
8	NaN	Experience	#years of professional experience
9	NaN	Income	Annual income of the customer ($000)

In [7]:
```python
description.drop('Unnamed: 0',axis=1,inplace=True)
```

In [8]:
```python
description.drop(axis=0,index=[0,1,2,3,4],inplace=True)
```

In [9]:
```python
description.rename(columns={'Unnamed: 1':'Column Name','Unnamed: 2':'Column Description'}, inplace=True)
```

In [10]:
```python
pd.set_option('display.max_colwidth',-1)
print(description)
```

```
           Column Name  \
5   NaN
6   ID
7   Age
8   Experience
9   Income
10  ZIPCode
11  Family
12  CCAvg
13  Education
14  Mortgage
15  Personal Loan
16  Securities Account
17  CD Account
18  Online
19  CreditCard

                                                         Column Description
5   NaN
6   Customer ID
7   Customer's age in completed years
8   #years of professional experience
9   Annual income of the customer ($000)
10  Home Address ZIP code.
11  Family size of the customer
12  Avg. spending on credit cards per month ($000)
13  Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
14  Value of house mortgage if any. ($000)
15  Did this customer accept the personal loan offered in the last campaign?
16  Does the customer have a securities account with the bank?
17  Does the customer have a certificate of deposit (CD) account with the bank?
18  Does the customer use internet banking facilities?
19  Does the customer use a credit card issued by UniversalBank?
```

In [11]:
```python
df.head()
```

Out[11]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditC
0	1	25	1	49	91107	4	1.6	1	0	0	1	0	0	
1	2	45	19	34	90089	3	1.5	1	0	0	1	0	0	
2	3	39	15	11	94720	1	1.0	1	0	0	0	0	0	
3	4	35	9	100	94112	1	2.7	2	0	0	0	0	0	
4	5	35	8	45	91330	4	1.0	2	0	0	0	0	0	

In [12]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 14 columns):
ID                    5000 non-null int64
Age                   5000 non-null int64
Experience            5000 non-null int64
Income                5000 non-null int64
ZIP Code              5000 non-null int64
Family                5000 non-null int64
CCAvg                 5000 non-null float64
Education             5000 non-null int64
Mortgage              5000 non-null int64
Personal Loan         5000 non-null int64
Securities Account    5000 non-null int64
CD Account            5000 non-null int64
Online                5000 non-null int64
CreditCard            5000 non-null int64
dtypes: float64(1), int64(13)
memory usage: 547.0 KB
```

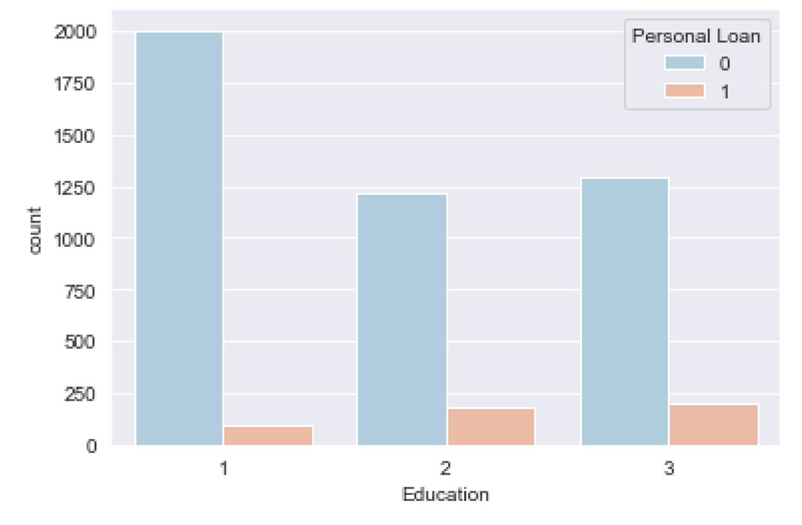
In [13]:
```python
df_loan_accept=df[df['Personal Loan']==1]
```

In [14]:
```python
sns.set_style('darkgrid')
g=sns.FacetGrid(df,row='Education',col='Family',hue='Personal Loan',palette='Set2')
g=g.map(plt.hist, 'CCAvg', alpha=0.5)
plt.legend(bbox_to_anchor=(1.7,3))
```

Out[14]: `<matplotlib.legend.Legend at 0x202f64e1b08>`

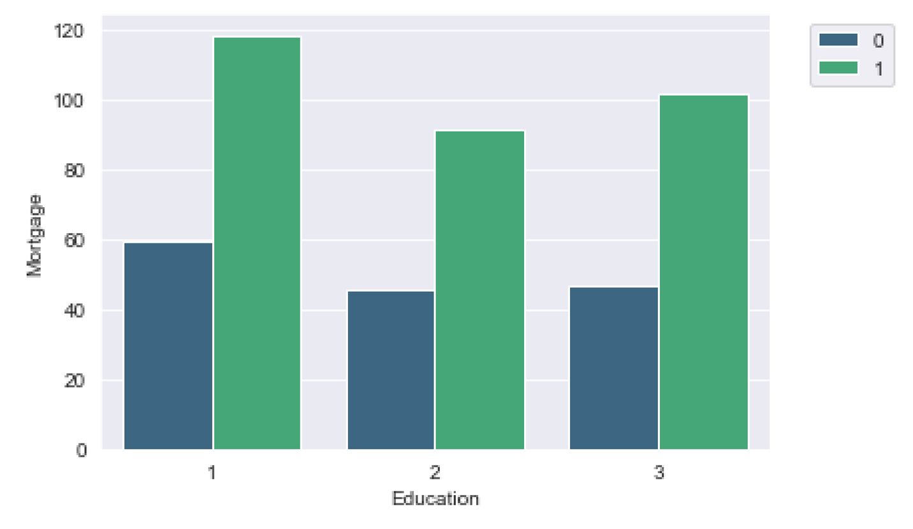
In [15]:
```python
sns.countplot(data=df,x='Education',hue='Personal Loan',palette='RdBu_r')
```

Out[15]: `<matplotlib.axes._subplots.AxesSubplot at 0x202f9548508>`



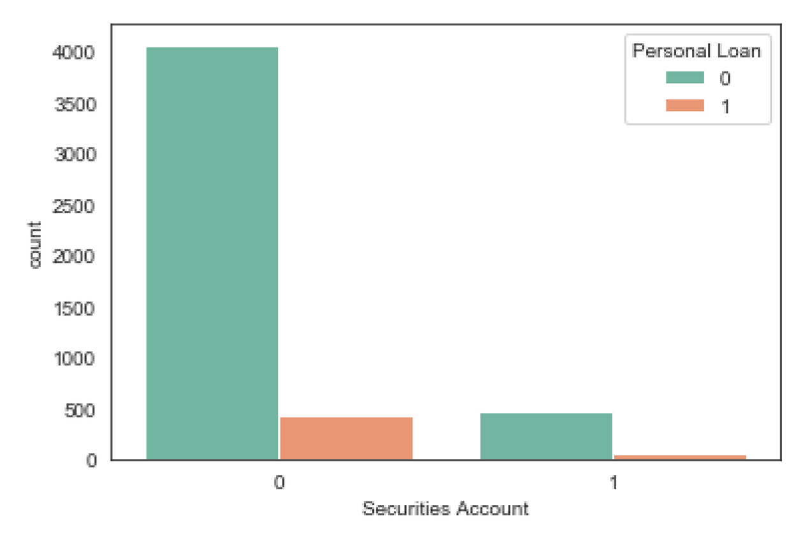
In [16]:
```python
sns.barplot('Education','Mortgage',hue='Personal Loan',data=df,palette='viridis',ci=None)
plt.legend(bbox_to_anchor=(1.2,1))
```

Out[16]: `<matplotlib.legend.Legend at 0x202f9bc1ac8>`



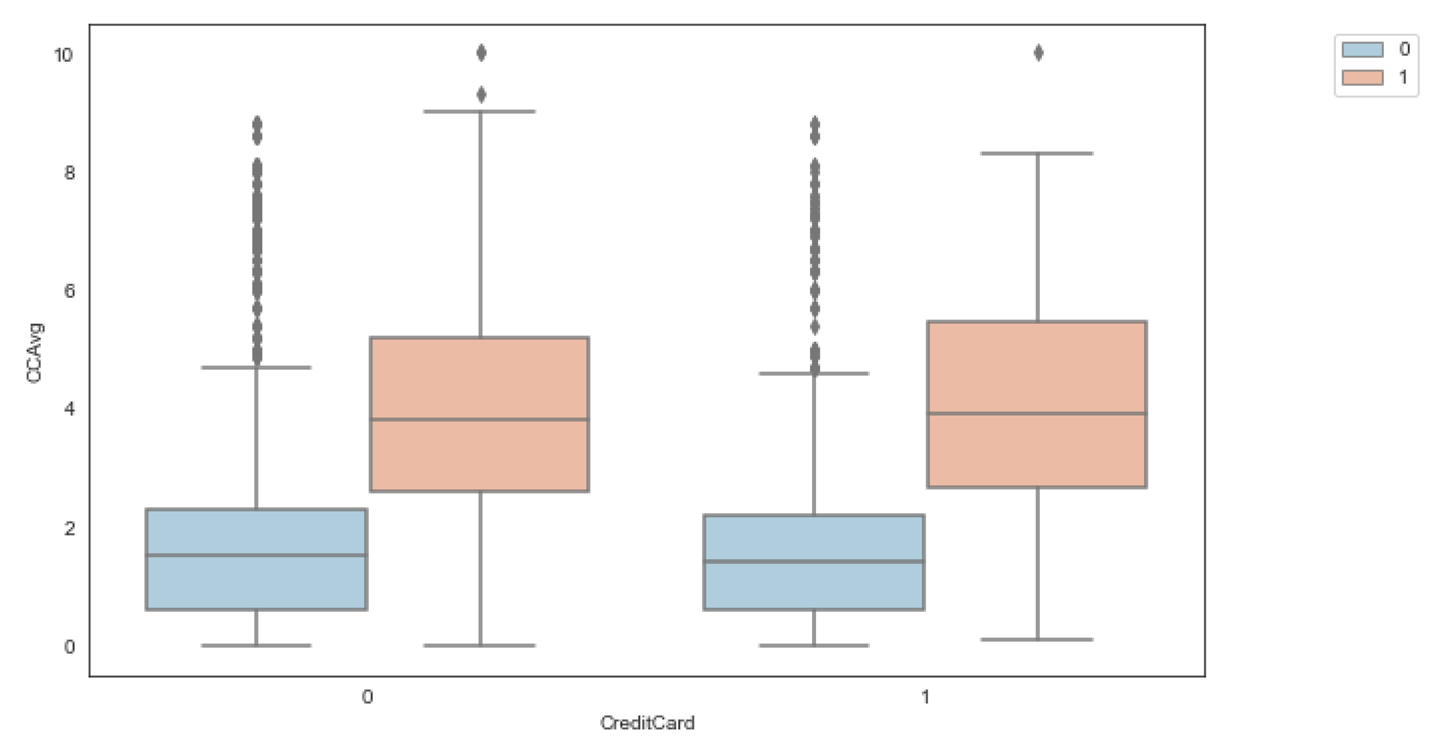
In [17]:
```python
sns.set_style('white')
sns.countplot(data=df,x='Securities Account',hue='Personal Loan',palette='Set2')
```

Out[17]: `<matplotlib.axes._subplots.AxesSubplot at 0x202f9c339c8>`



In [18]:
```python
plt.figure(figsize=(10,6))
sns.boxplot('CreditCard','CCAvg',hue='Personal Loan',data=df,palette='RdBu_r')
plt.legend(bbox_to_anchor=(1.2,1))
```

Out[18]: `<matplotlib.legend.Legend at 0x202f9c941c8>`



In [19]:
```python
df.columns
```

Out[19]:
```
Index(['ID', 'Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Personal Loan', 'Securities Account',
       'CD Account', 'Online', 'CreditCard'],
      dtype='object')
```

In [20]:
```python
X=pd.DataFrame(columns=['Age','Experience','Income','Family','CCAvg','Education','Mortgage',
'Securities Account','CD Account','CreditCard','Online'],data=df)
```

In [21]:
```python
y=df['Personal Loan']
```

In [22]:
```python
from sklearn.model_selection import train_test_split
```

In [23]:
```python
X_train, X_test, y_train, y_test= train_test_split(X,y)
```

In [24]:
```python
from sklearn.tree import DecisionTreeClassifier
```

In [25]:
```python
dtree= DecisionTreeClassifier(max_leaf_nodes=3)
```

In [26]:
```python
dtree.fit(X_train,y_train)
```

Out[26]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=3,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

In [27]:
```python
predictions= dtree.predict(X_test)
```

In [28]:
```python
from sklearn.metrics import classification_report, confusion_matrix
```

In [29]:
```python
print(classification_report(y_test,predictions))
```

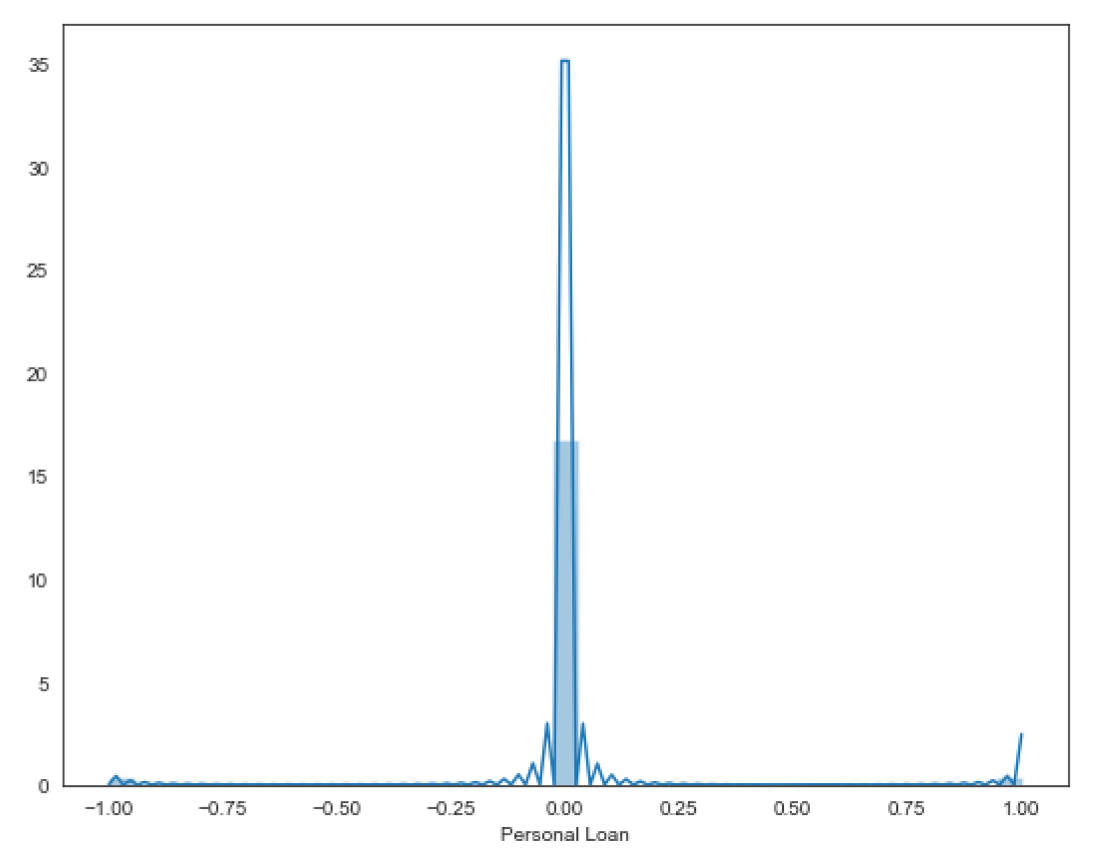
```
              precision    recall  f1-score   support

           0       0.97      0.98      0.98      1136
           1       0.79      0.75      0.77       114

    accuracy                           0.96      1250
   macro avg       0.88      0.86      0.87      1250
weighted avg       0.96      0.96      0.96      1250
```

In [30]:
```python
print(confusion_matrix(y_test,predictions))
```

```
[[1113   23]
 [  29   85]]
```

In [31]:
```python
plt.figure(figsize=(9,7))
sns.distplot(y_test-predictions)
```

Out[31]: `<matplotlib.axes._subplots.AxesSubplot at 0x202fa8baa48>`



In [ ]: