Practical 9

In [1]:

```
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt # plotting
import numpy as np # linear algebra
import os # accessing directory structure
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

In [3]:

```
df = pd.read_csv('Churn_Modelling.csv')
```

In [4]:

```
# Distribution graphs (histogram/bar graph) of column data
def plotPerColumnDistribution(df, nGraphShown, nGraphPerRow):
    nunique = df.nunique()
    df = df[[col for col in df if nunique[col] > 1 and nunique[col] < 50]] # For displaying purposes, pick columns that h
    nRow, nCol = df.shape
    columnNames = list(df)
    nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
    plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80, facecolor = 'w', edgecolor = 'k')
    for i in range(min(nCol, nGraphShown)):
        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
        columnDf = df.iloc[:, i]
        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
            valueCounts = columnDf.value_counts()
            valueCounts.plot.bar()
        else:
            columnDf.hist()
        plt.ylabel('counts')
        plt.xticks(rotation = 90)
        plt.title(f'{columnNames[i]} (column {i})')
    plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
    plt.show()
```

In [5]:

```
# Correlation matrix
def plotCorrelationMatrix(df, graphWidth):
    filename = df.dataframeName
    df = df.dropna('columns') # drop columns with NaN
    df = df[[col for col in df if df[col].nunique() > 1]] # keep columns where there are more than 1 unique values
    if df.shape[1] < 2:
        print(f'No correlation plots shown: The number of non-NaN or constant columns ({df.shape[1]}) is less than 2')
        return
    corr = df.corr()
    plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80, facecolor='w', edgecolor='k')
    corrMat = plt.matshow(corr, fignum = 1)
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
    plt.yticks(range(len(corr.columns)), corr.columns)
    plt.gca().xaxis.tick_bottom()
    plt.colorbar(corrMat)
    plt.title(f'Correlation Matrix for {filename}', fontsize=15)
    plt.show()
```

In [6]:

```
# Scatter and density plots
def plotScatterMatrix(df, plotSize, textSize):
    df = df.select_dtypes(include =[np.number]) # keep only numerical columns
    # Remove rows and columns that would lead to df being singular
    df = df.dropna('columns')
    df = df[[col for col in df if df[col].nunique() > 1]] # keep columns where there are more than 1 unique values
    columnNames = list(df)
    if len(columnNames) > 10: # reduce the number of columns for matrix inversion of kernel density plots
        columnNames = columnNames[:10]
    df = df[columnNames]
    ax = pd.plotting.scatter_matrix(df, alpha=0.75, figsize=[plotSize, plotSize], diagonal='kde')
    corrs = df.corr().values
    for i, j in zip(*plt.np.triu_indices_from(ax, k = 1)):
        ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords='axes fraction', ha='center', va='cente
    plt.suptitle('Scatter and Density Plot')
    plt.show()
```

In [8]:

```
nRowsRead = 1000 # specify 'None' if want to read whole file
# Churn_Modelling.csv has 10001 rows in reality, but we are only loading/previewing the first 1000 rows
df1 = pd.read_csv('Churn_Modelling.csv', delimiter=',', nrows = nRowsRead)
df1.dataframeName = 'Churn_Modelling.csv'
nRow, nCol = df1.shape
print(f'There are {nRow} rows and {nCol} columns')
```

```
There are 1000 rows and 14 columns
```

In [9]:

```
df1.head(5)
```

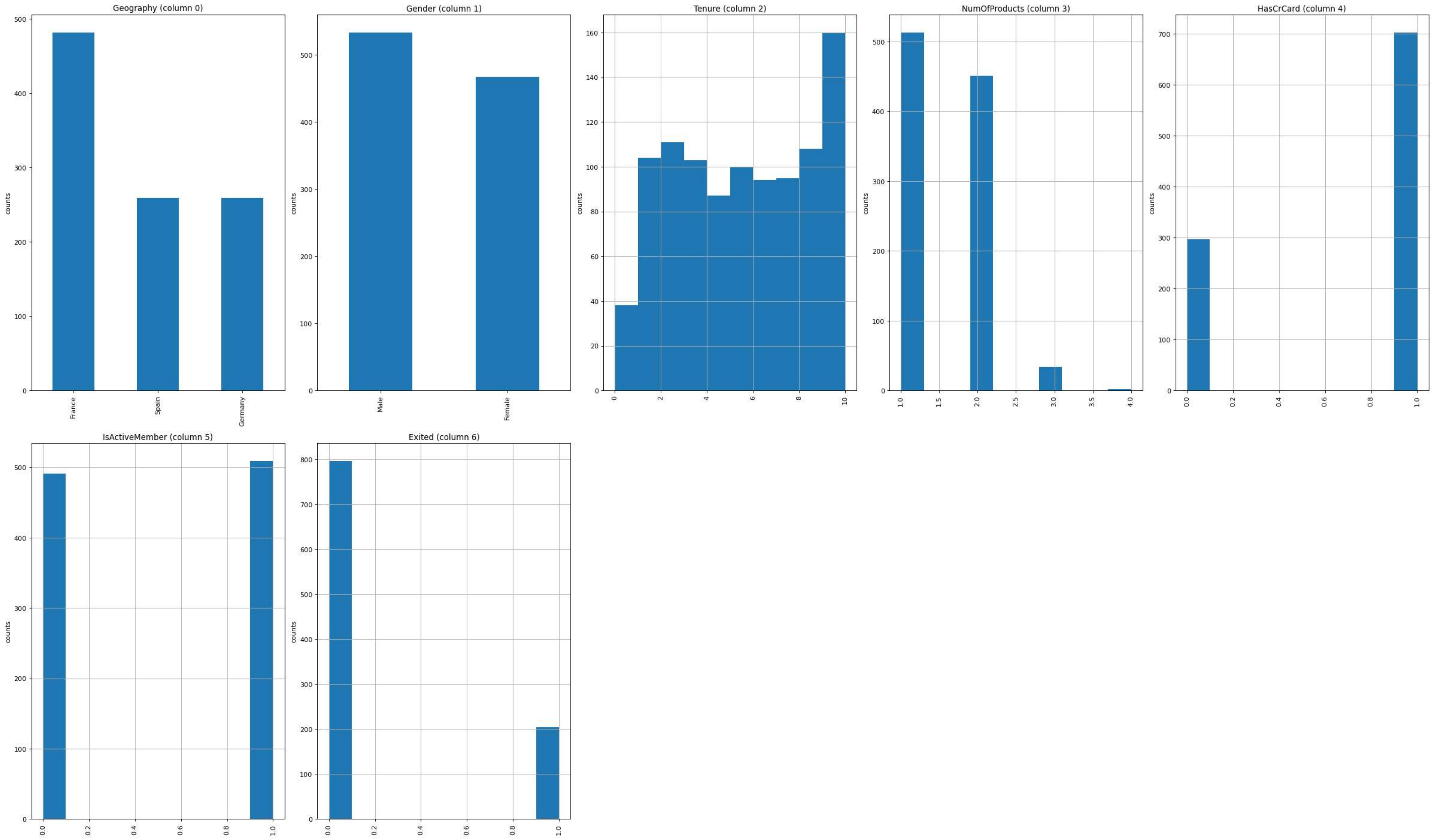
Out[9]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1

In [10]:

```
plotPerColumnDistribution(df1, 10, 5)
```

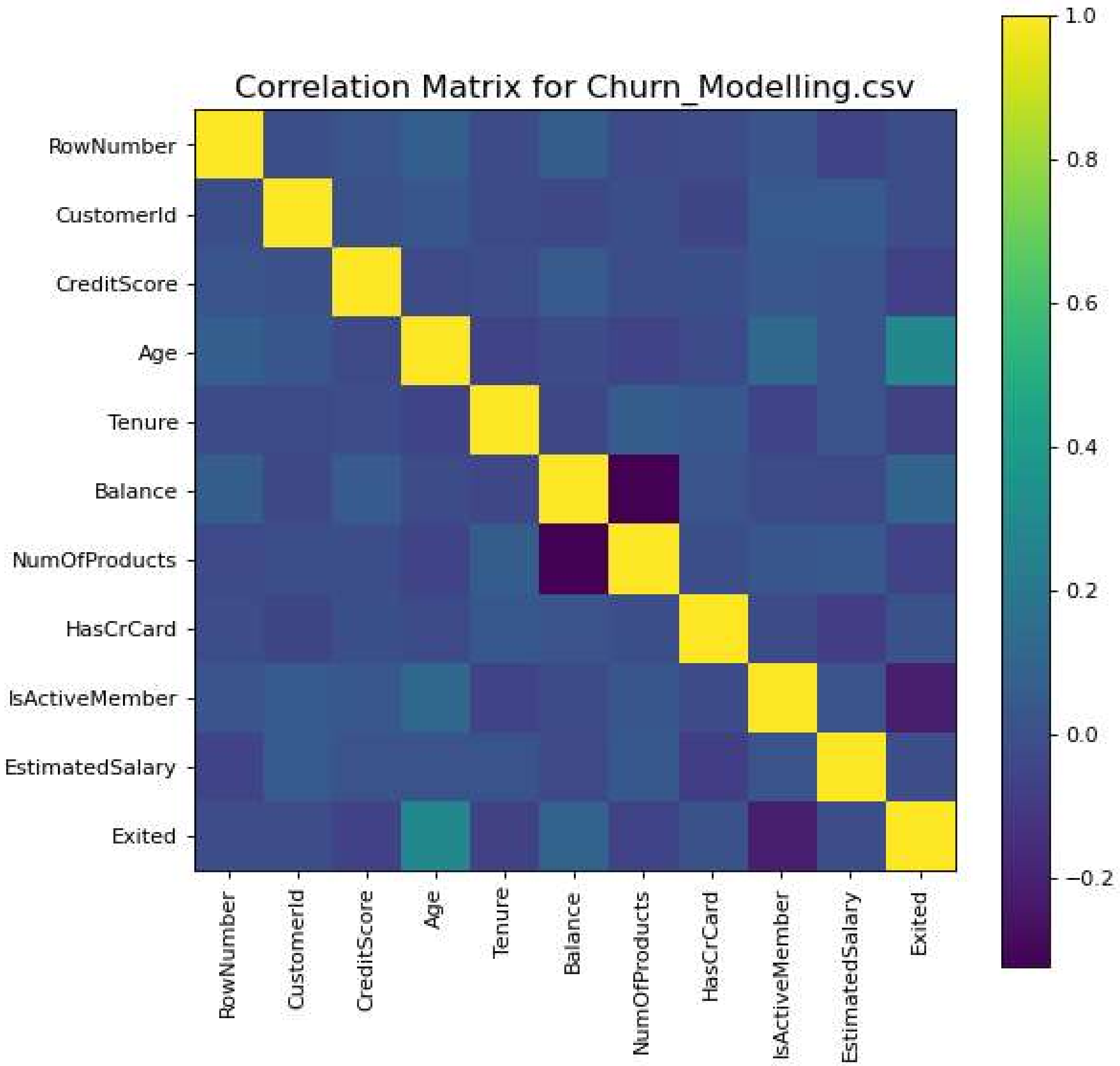
```
C:\Users\omkar\AppData\Local\Temp/ipykernel_14672/964395601.py:10: MatplotlibDeprecationWarning: Passing non-integers as
three-element position specification is deprecated since 3.3 and will be removed two minor releases later.
  plt.subplot(nGraphRow, nGraphPerRow, i + 1)
C:\Users\omkar\AppData\Local\Temp/ipykernel_14672/964395601.py:10: MatplotlibDeprecationWarning: Passing non-integers as
three-element position specification is deprecated since 3.3 and will be removed two minor releases later.
  plt.subplot(nGraphRow, nGraphPerRow, i + 1)
C:\Users\omkar\AppData\Local\Temp/ipykernel_14672/964395601.py:10: MatplotlibDeprecationWarning: Passing non-integers as
three-element position specification is deprecated since 3.3 and will be removed two minor releases later.
  plt.subplot(nGraphRow, nGraphPerRow, i + 1)
```



In [11]:

```
plotCorrelationMatrix(df1, 8)
```

```
C:\Users\omkar\AppData\Local\Temp/ipykernel_14672/3510424060.py:4: FutureWarning: In a future version of pandas all argum
ents of DataFrame.dropna will be keyword-only
  df = df.dropna('columns') # drop columns with NaN
```

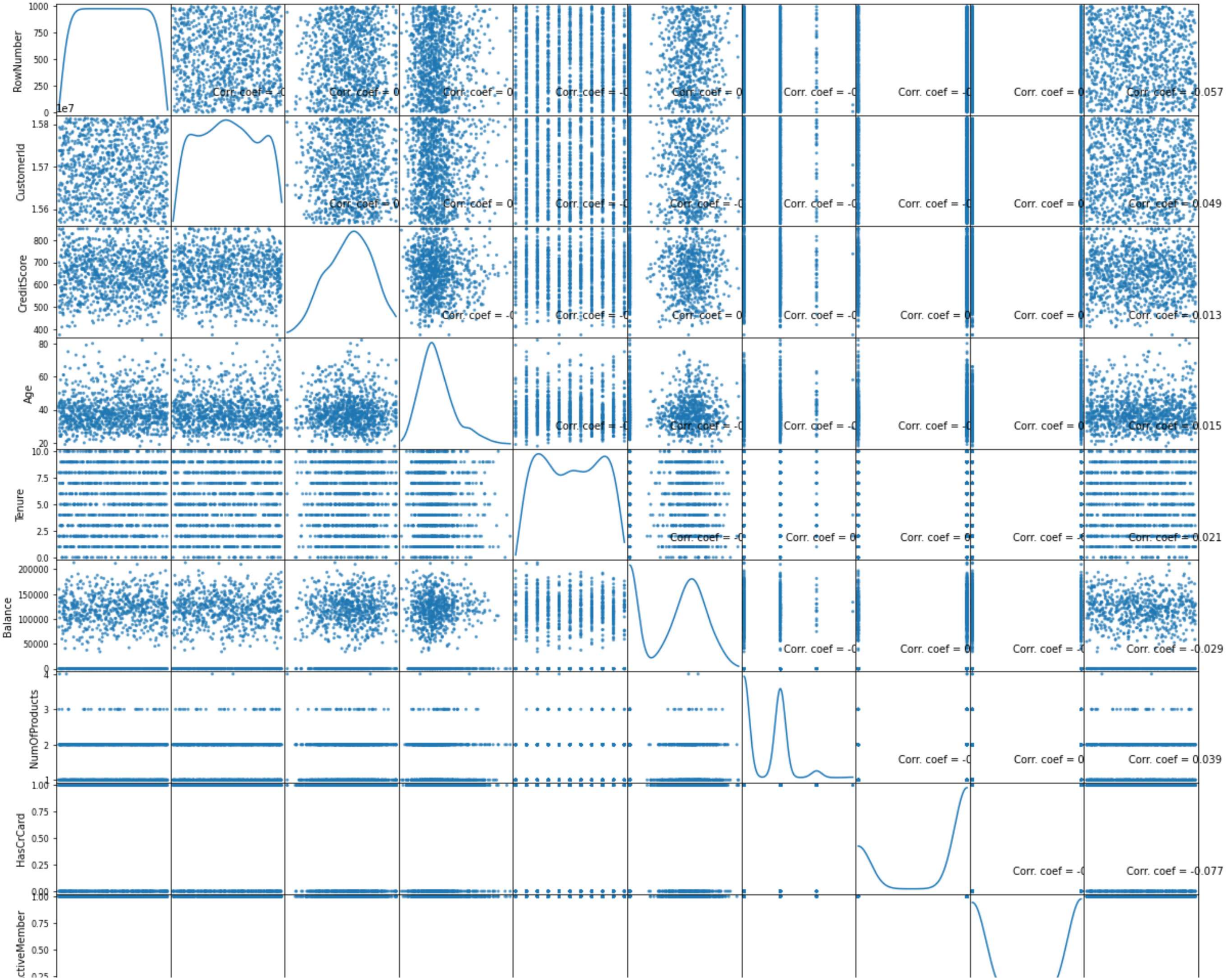


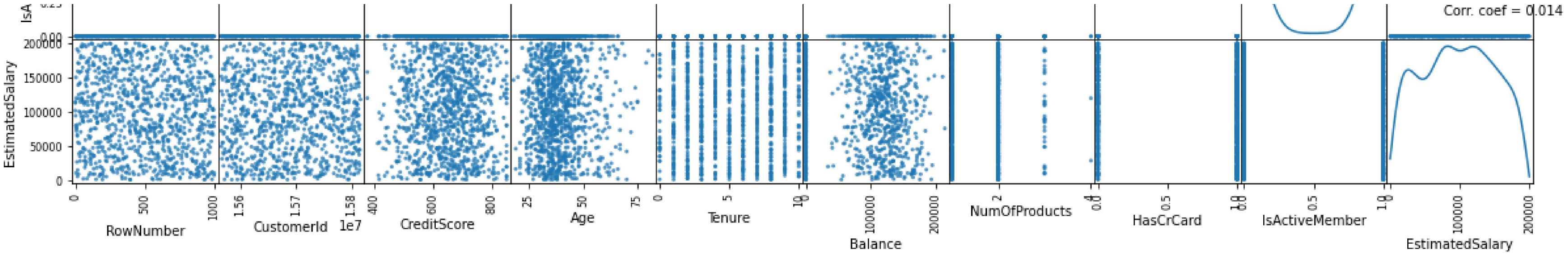
In [12]:

```
plotScatterMatrix(df1, 20, 10)
```

```
C:\Users\omkar\AppData\Local\Temp/ipykernel_14672/102845399.py:5: FutureWarning: In a future version of pandas all argume
nts of DataFrame.dropna will be keyword-only
  df = df.dropna('columns')
```

Scatter and Density Plot


RowNumber
1000
750
500
250
0
Corr. coef = -0
Corr. coef = 0
Corr. coef = 0
Corr. coef = -0
Corr. coef = 0
Corr. coef = -0
Corr. coef = -0
Corr. coef = 0
Corr. coef = -0.057
1e7
CustomerId
1.58
1.57
1.56
Corr. coef = 0
Corr. coef = 0
Corr. coef = -0
Corr. coef = -0
Corr. coef = -0
Corr. coef = -0
Corr. coef = 0
Corr. coef = 0.049
CreditScore
800
700
600
500
400
Corr. coef = -0
Corr. coef = -0
Corr. coef = 0
Corr. coef = -0
Corr. coef = 0
Corr. coef = 0
Corr. coef = 0.013
Age
80
60
40
20
Corr. coef = -0
Corr. coef = -0
Corr. coef = -0
Corr. coef = -0
Corr. coef = 0
Corr. coef = 0.015
Tenure
10.0
7.5
5.0
2.5
0.0
Corr. coef = -0
Corr. coef = 0
Corr. coef = 0
Corr. coef = -
Corr. coef = 0.021
Balance
200000
150000
100000
50000
0
Corr. coef = -0
Corr. coef = 0
Corr. coef = -
Corr. coef = -0.029
NumOfProducts
4
3
2
1
Corr. coef = -0
Corr. coef = 0
Corr. coef = 0.039
HasCrCard
1.00
0.75
0.50
0.25
0.00
Corr. coef = -
Corr. coef = -0.077
ctiveMember
1.00
0.75
0.50
0.25


IsA
0.25
0.00
Corr. coef = 0.014
EstimatedSalary
200000
150000
100000
50000
0
0
500
1000
RowNumber
1.56
1.57
1.58
CustomerId
1e7
400
600
800
CreditScore
25
50
75
Age
0
5
10
Tenure
0
100000
200000
Balance
2
4
NumOfProducts
0.0
0.5
1.0
HasCrCard
0.0
0.5
1.0
IsActiveMember
0
100000
200000
EstimatedSalary

In [ ]: