Caso de Estudio: BANCO UNO

Universidad Castro Carazo

Programa Técnico de Análisis de Datos

Curso: Introducción a Machine Learning

Estudiante: Wendy Venegas Quiros

Objetivo: Diseñar o implementar una solucion mediante el cual se facilite la toma de decisiones respecto al proceso de otorgamiento de creditos. El sentido es lograr determinar si un credito debe ser otorgado o no, donde se utilice una herramienta informatica como apoyo para la creacion de modelos predictivos.

Definiendo el Proceso para el diseño del algoritmo de Machine Learning

Conociendo los datos y su dominio.

Información del conjunto de datos:

Esta investigación se centró en el caso de los pagos por defecto de los clientes de Banco Uno y compara la precisión predictiva de la probabilidad de incumplimiento entre seis métodos de minería de datos.

De la perspectiva de la gestión de riesgos, el resultado de la precisión predictiva de la estimación La probabilidad de incumplimiento será más valiosa que el resultado binario de la clasificación de Clientes creíbles o no creíbles.

Debido a que se desconoce la probabilidad real de incumplimiento, este estudio presentó el novedoso método de clasificación de suavizado para estimar la probabilidad real de defecto. Con la probabilidad real de incumplimiento como variable de respuesta (Y), y la probabilidad predictiva de incumplimiento como la variable independiente (X), la simple lineal El resultado de la regresión (Y = A + BX) muestra que el modelo de pronóstico producido por la red neuronal tiene el coeficiente de determinación más alto; su intersección de regresión (A) es cercano a cero y coeficiente de regresión (B) a uno. Por lo tanto, entre los seis datos de minería técnicas, la red neuronal artificial es la única que puede estimar con precisión el valor real probabilidad de incumplimiento.

Información de los atributos:

NOTA: La siguiente es información actualizada del autor de la fuente.

Esta investigación empleó una variable binaria, pago predeterminado (Sí = 1, No = 0), como variable de respuesta. Este estudio revisó la literatura y utilizó las siguientes 23 variables como

variables explicativas:

X1: Monto del crédito otorgado (dólar NT): incluye tanto al consumidor individual crédito y su crédito familiar (complementario).

X2: Género (1 = masculino; 2 = femenino).

X3: Educación (1 = posgrado; 2 = universidad; 3 = bachillerato; 0, 4, 5, 6 = otros).

X4: Estado civil (1 = casado; 2 = soltero; 3 = divorciado; 0 = otros).

X5: Edad (año).

X6 - X11: Historial de pagos pasados. Realizamos un seguimiento de los últimos registros de pagos mensuales (desde Abril a septiembre de 2005) de la siguiente manera: X6 = el estado de reembolso en septiembre de 2005; X7 = el estado de reembolso en agosto de 2005; . . .; X11 = el estado de reembolso en abril de 2005.

La escala de medición para el estado de reembolso es:

2: Sin consumo; -1: pagado en su totalidad; 0: El uso de crédito renovable; 1 = retraso en el pago por un mes; 2 = retraso en el pago de dos meses; . . .; 8 = retraso en el pago de ocho meses; 9 = retraso en el pago de nueve meses o más.

X12-X17: Monto del extracto de la factura (dólar NT). X12 = monto del extracto de la factura en Septiembre de 2005; X13 = monto del estado de cuenta en agosto de 2005; . . .; X17 = cantidad de estado de cuenta en abril de 2005.

X18-X23: Monto del pago anterior (dólar NT). X18 = monto pagado en septiembre, 2005; X19 = monto pagado en agosto de 2005; . . .; X23 = monto pagado en abril de 2005.

Y: comportamiento del cliente; Y = 0 entonces no predeterminado, Y = 1 luego predeterminado "

Cargando y Examinando los datos.

In [1]:
```
from sqlalchemy import create_engine
import pymysql
import pandas as pd
```

In [2]:
```
connection = pymysql.connect(host='data-analytics-2018.cbrosir2cswx.us-east-1.r
user='deepAnalytics',
password='Sqltask1234!',
database='Credit',
charset='utf8mb4',
cursorclass=pymysql.cursors.DictCursor)
```

In [3]:
```
df = pd.read_sql('SELECT * FROM credit', con=connection)
```

```
C:\Users\bryan\anaconda3\lib\site-packages\pandas\io\sql.py:761: UserWarning:
pandas only support SQLAlchemy connectable(engine/connection) ordatabase stri
ng URI or sqlite3 DBAPI2 connectionother DBAPI2 objects are not tested, pleas
e consider using SQLAlchemy
  warnings.warn(
```

In [4]:
```
df. to_csv ( 'BancoUno.csv' , header = False , index = False )
```

In [5]:
```
df.head()
```

Out[5]:

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	...
0	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	...
1	20000	female	university	1	24	2	2	-1	-1	-2	...
2	120000	female	university	2	26	-1	2	0	0	0	...
3	90000	female	university	2	34	0	0	0	0	0	...
4	50000	female	university	1	37	0	0	0	0	0	...

5 rows × 24 columns

Cambiando el nombre al DataFrame

In [7]:
```
credit=df
```

In [8]:
```
credit.head()
```

Out[8]:

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	...
0	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	...
1	20000	female	university	1	24	2	2	-1	-1	-2	...
2	120000	female	university	2	26	-1	2	0	0	0	...
3	90000	female	university	2	34	0	0	0	0	0	...
4	50000	female	university	1	37	0	0	0	0	0	...

5 rows × 24 columns

Aplicando Pandas Profiling

conda install -c conda-forge pandas-profiling

In [10]:
```
import pandas_profiling
```

pandas_profiling.ProfileReport(credit)

#Se coloca en markdown para no hacer muy pesado el codigo porque mas adelante se genera otro reporte

Limpieza de Datos, trabajo con valores nulos y faltantes, reducción de datos (eliminación de observaciones repetidas) y discretización de la información (creación de bins para reducir la data)

In [11]:
```
credit.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3670 entries, 0 to 3669
Data columns (total 24 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   X1      3670 non-null   object
 1   X2      3670 non-null   object
 2   X3      3670 non-null   object
 3   X4      3670 non-null   object
 4   X5      3670 non-null   object
 5   X6      3670 non-null   object
 6   X7      3670 non-null   object
 7   X8      3670 non-null   object
 8   X9      3670 non-null   object
 9   X10     3670 non-null   object
 10  X11     3670 non-null   object
 11  X12     3670 non-null   object
 12  X13     3670 non-null   object
 13  X14     3670 non-null   object
 14  X15     3670 non-null   object
 15  X16     3670 non-null   object
 16  X17     3670 non-null   object
 17  X18     3670 non-null   object
 18  X19     3670 non-null   object
 19  X20     3670 non-null   object
 20  X21     3670 non-null   object
 21  X22     3670 non-null   object
 22  X23     3670 non-null   object
 23  Y       3670 non-null   object
dtypes: object(24)
memory usage: 688.2+ KB
```

In [12]:
```
credit.columns
```

Out[12]:
```
Index(['X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X7', 'X8', 'X9', 'X10', 'X11',
       'X12', 'X13', 'X14', 'X15', 'X16', 'X17', 'X18', 'X19', 'X20', 'X21',
       'X22', 'X23', 'Y'],
      dtype='object')
```

In [13]:
```
credit.rename({'X1':'LIMIT_BAL'}, axis=1, inplace=True)
credit.rename({'X2':'SEX'}, axis=1, inplace=True)
credit.rename({'X3':'EDUCATION'}, axis=1, inplace=True)
credit.rename({'X4':'MARRIAGE'}, axis=1, inplace=True)
credit.rename({'X5':'AGE'}, axis=1, inplace=True)
credit.rename({'X6':'PAY_0'}, axis=1, inplace=True)
credit.rename({'X7':'PAY_2'}, axis=1, inplace=True)
credit.rename({'X8':'PAY_3'}, axis=1, inplace=True)
credit.rename({'X9':'PAY_4'}, axis=1, inplace=True)
credit.rename({'X10':'PAY_5'}, axis=1, inplace=True)
credit.rename({'X11':'PAY_6'}, axis=1, inplace=True)
credit.rename({'X12':'BILL_AMT1'}, axis=1, inplace=True)
credit.rename({'X13':'BILL_AMT2'}, axis=1, inplace=True)
credit.rename({'X14':'BILL_AMT3'}, axis=1, inplace=True)
credit.rename({'X15':'BILL_AMT4'}, axis=1, inplace=True)
credit.rename({'X16':'BILL_AMT5'}, axis=1, inplace=True)
credit.rename({'X17':'BILL_AMT6'}, axis=1, inplace=True)
credit.rename({'X18':'PAY_AMT1'}, axis=1, inplace=True)
credit.rename({'X19':'PAY_AMT2'}, axis=1, inplace=True)
credit.rename({'X20':'PAY_AMT3'}, axis=1, inplace=True)
credit.rename({'X21':'PAY_AMT4'}, axis=1, inplace=True)
credit.rename({'X22':'PAY_AMT5'}, axis=1, inplace=True)
credit.rename({'X23':'PAY_AMT6'}, axis=1, inplace=True)
credit.rename({'Y':'default payment next month'}, axis=1, inplace=True)
credit.columns
```

Out[13]:
```
Index(['LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2',
       'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2',
       'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1',
       'PAY_AMT2', 'PAY_AMT3', 'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6',
       'default payment next month'],
      dtype='object')
```

In [14]:
```
credit
```

Out[14]:

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5
0	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5
1	20000	female	university	1	24	2	2	-1	-1	-2
2	120000	female	university	2	26	-1	2	0	0	0
3	90000	female	university	2	34	0	0	0	0	0
4	50000	female	university	1	37	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
3665	220000	female	university	1	32	0	0	0	0	0
3666	70000	female	university	2	34	1	2	2	2	0
3667	120000	male	university	2	37	-1	2	0	0	0
3668	180000	female	university	2	32	0	0	0	0	0
3669	50000	female	high school	1	57	0	0	0	0	0

3670 rows × 24 columns

In [15]:
```
credit.drop_duplicates(inplace=True)
credit
```

Out[15]:

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5
0	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5
1	20000	female	university	1	24	2	2	-1	-1	-2
2	120000	female	university	2	26	-1	2	0	0	0
3	90000	female	university	2	34	0	0	0	0	0
4	50000	female	university	1	37	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
2393	50000	male	high school	1	32	2	3	2	2	2
2394	20000	female	high school	2	49	0	0	2	0	-1
2395	130000	female	university	2	24	1	-2	-1	-1	-1
2396	110000	female	high school	1	27	0	0	0	0	0
2397	200000	male	university	1	29	0	0	0	2	2

2397 rows × 24 columns

In [16]:
```
credit.drop([0],axis=0, inplace=True)
credit
```

Out[16]:

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5
1	20000	female	university	1	24	2	2	-1	-1	-2
2	120000	female	university	2	26	-1	2	0	0	0
3	90000	female	university	2	34	0	0	0	0	0
4	50000	female	university	1	37	0	0	0	0	0
5	50000	male	university	1	57	-1	0	-1	0	0
...	...	...	...	...	...	...	...	...	...	...
2393	50000	male	high school	1	32	2	3	2	2	2
2394	20000	female	high school	2	49	0	0	2	0	-1
2395	130000	female	university	2	24	1	-2	-1	-1	-1
2396	110000	female	high school	1	27	0	0	0	0	0
2397	200000	male	university	1	29	0	0	0	2	2

2396 rows × 24 columns

In [17]:
```
credit.duplicated().sum()
```

Out[17]:
```
0
```

In [18]:
```
credit.shape
```

Out[18]:
```
(2396, 24)
```

In [19]:
```
credit.dtypes
```

Out[19]:
```
LIMIT_BAL                   object
SEX                         object
EDUCATION                   object
MARRIAGE                    object
AGE                         object
PAY_0                       object
PAY_2                       object
PAY_3                       object
PAY_4                       object
PAY_5                       object
PAY_6                       object
BILL_AMT1                   object
BILL_AMT2                   object
BILL_AMT3                   object
BILL_AMT4                   object
BILL_AMT5                   object
BILL_AMT6                   object
PAY_AMT1                    object
PAY_AMT2                    object
PAY_AMT3                    object
PAY_AMT4                    object
PAY_AMT5                    object
PAY_AMT6                    object
default payment next month  object
dtype: object
```

In [20]:
```
credit[['LIMIT_BAL', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4', 'PA
        'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AM
        'PAY_AMT6']]=credit[['LIMIT_BAL','MARRIAGE', 'AGE', 'PAY_0', 'PAY_2',
        'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_A
        'PAY_AMT5', 'PAY_AMT6']].astype("int")
```

In [21]:
```
credit.dtypes
```

Out[21]:
```
LIMIT_BAL                    int32
SEX                         object
EDUCATION                   object
MARRIAGE                     int32
AGE                          int32
PAY_0                        int32
PAY_2                        int32
PAY_3                        int32
PAY_4                        int32
PAY_5                        int32
PAY_6                        int32
BILL_AMT1                    int32
BILL_AMT2                    int32
BILL_AMT3                    int32
BILL_AMT4                    int32
BILL_AMT5                    int32
BILL_AMT6                    int32
PAY_AMT1                     int32
PAY_AMT2                     int32
PAY_AMT3                     int32
PAY_AMT4                     int32
PAY_AMT5                     int32
PAY_AMT6                     int32
default payment next month  object
dtype: object
```

In [22]:
```
pandas_profiling.ProfileReport(credit)
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Out[22]:

Trabajando con datos no numéricos (One-Hot Encoding)

In [23]:
```
credit= pd.get_dummies(credit)
```

In [24]:
```
credit.dtypes
```

Out[24]:
```
LIMIT_BAL                                int32
MARRIAGE                                 int32
AGE                                      int32
PAY_0                                    int32
PAY_2                                    int32
PAY_3                                    int32
PAY_4                                    int32
PAY_5                                    int32
PAY_6                                    int32
BILL_AMT1                                int32
BILL_AMT2                                int32
BILL_AMT3                                int32
BILL_AMT4                                int32
BILL_AMT5                                int32
BILL_AMT6                                int32
PAY_AMT1                                 int32
PAY_AMT2                                 int32
PAY_AMT3                                 int32
PAY_AMT4                                 int32
PAY_AMT5                                 int32
PAY_AMT6                                 int32
SEX_female                               uint8
SEX_male                                 uint8
EDUCATION_graduate school                uint8
EDUCATION_high school                    uint8
EDUCATION_other                          uint8
EDUCATION_university                     uint8
default payment next month_default       uint8
default payment next month_not default   uint8
dtype: object
```

In [25]:
```
credit.head()
```

Out[25]:

	LIMIT_BAL	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	...
1	20000	1	24	2	2	-1	-1	-2	-2	3913	...
2	120000	2	26	-1	2	0	0	0	2	2682	...
3	90000	2	34	0	0	0	0	0	0	29239	...
4	50000	1	37	0	0	0	0	0	0	46990	...
5	50000	1	57	-1	0	-1	0	0	0	8617	...

5 rows × 29 columns

Visualizando los datos

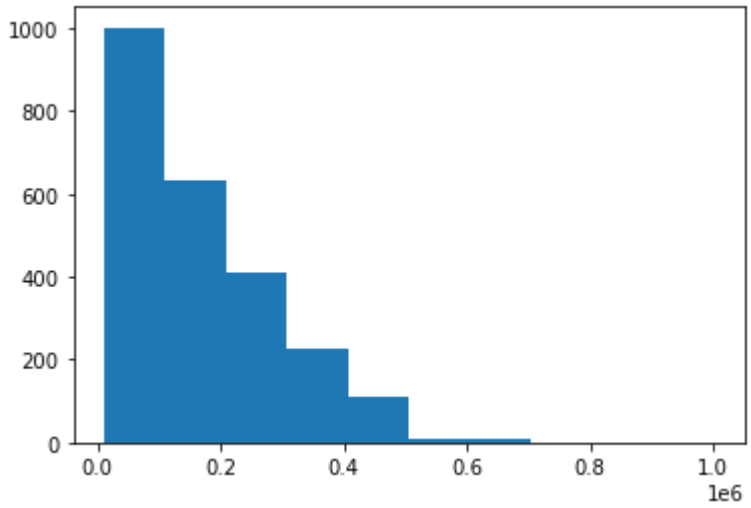
In [28]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [29]:
```
header= credit.dtypes.index
print(header)
```

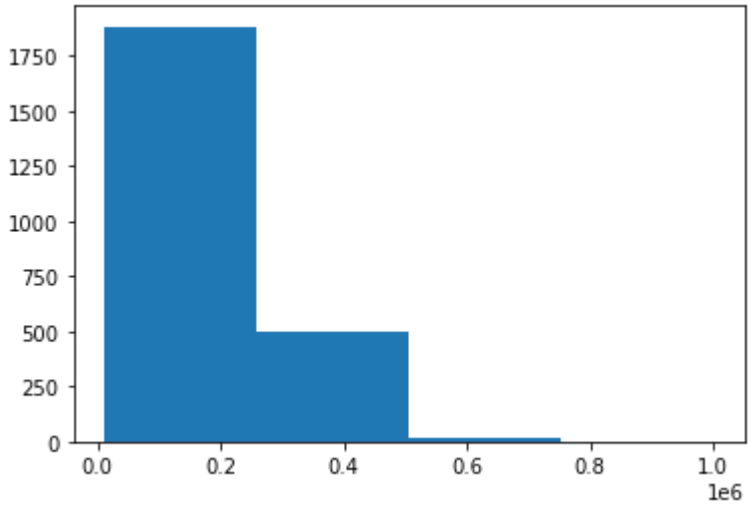
```
Index(['LIMIT_BAL', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4',
       'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4',
       'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3',
       'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6', 'SEX_female', 'SEX_male',
       'EDUCATION_graduate school', 'EDUCATION_high school', 'EDUCATION_othe
r',
       'EDUCATION_university', 'default payment next month_default',
       'default payment next month_not default'],
      dtype='object')
```

- HISTOGRAMAS

In [30]:
```
plt.hist(credit['LIMIT_BAL'])
plt.show()
```

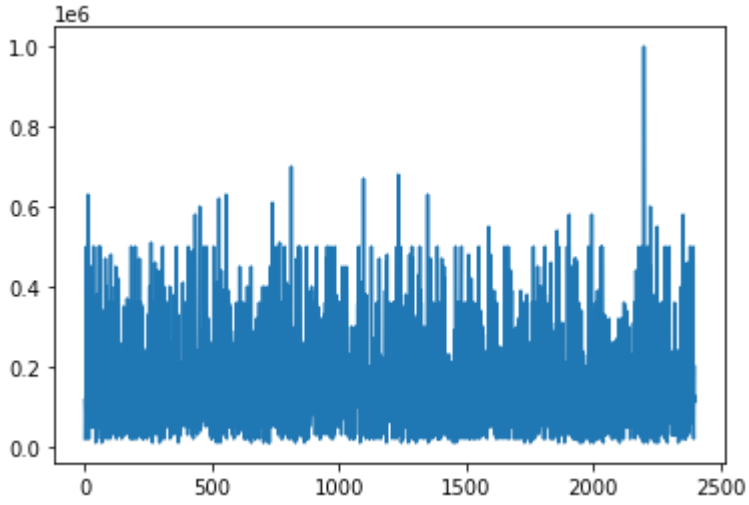


In [31]:
```
plt.hist(credit['LIMIT_BAL'], bins=4)
plt.show()
```



- GRAFICAS DE LINEAS

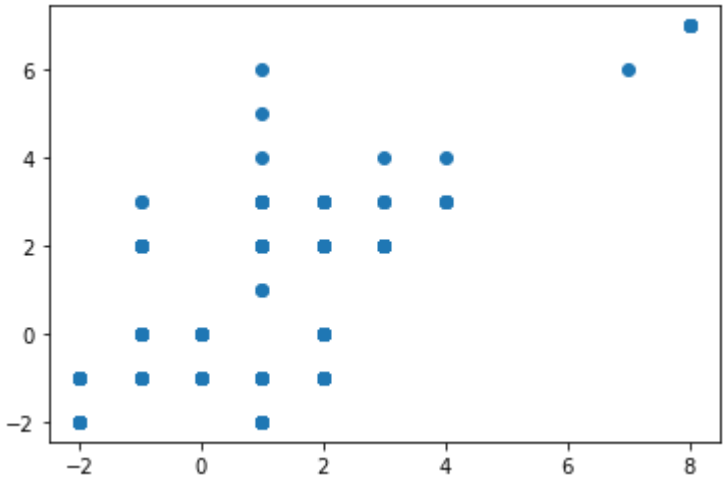
In [32]:
```
plt.plot(credit['LIMIT_BAL'])
plt.show()
```



- GRAFICAS DE DISPERSION

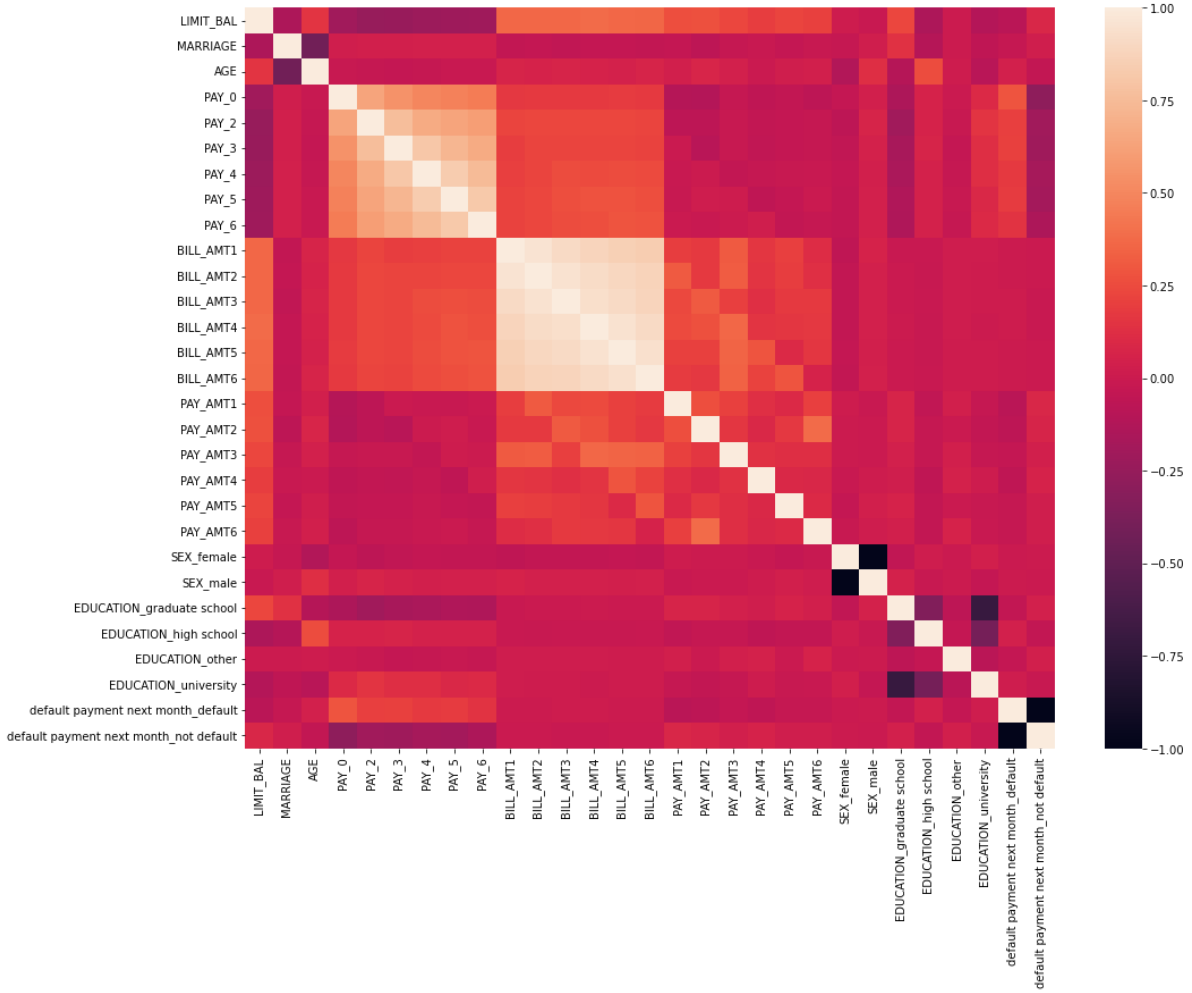
In [33]:
```
x=credit['PAY_0']
y=credit['PAY_2']
```

In [34]:
```
plt.scatter(x,y)
plt.show()
```

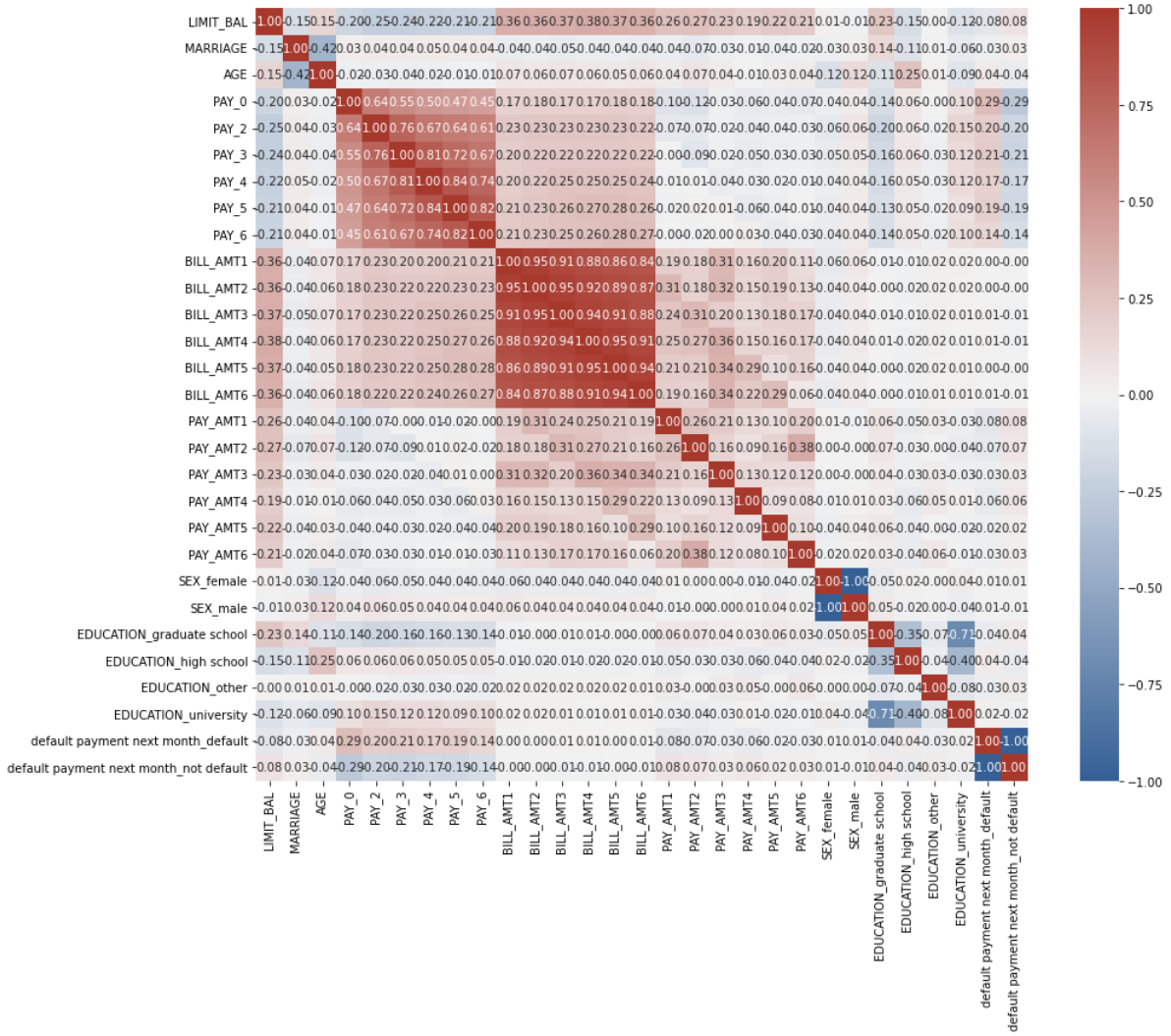


- MATRIZ DE CORRELACION

In [35]:
```
import seaborn as sns

matrix= credit.corr()
plt.figure(figsize=(16,12))
_= sns.heatmap(matrix)
```



In [36]:
```
plt.figure(figsize=(16,12))

cmap = sns.diverging_palette(250, 15, s=75, l=40,
                             n=9, center="light", as_cmap=True)

_ = sns.heatmap(matrix, center=0, annot=True,
                fmt='.2f', square=True, cmap=cmap)
```



- Eliminacion de las correlaciones altas

In [37]:
```
ccredit=credit.copy()
```

In [38]:
```
ccredit.head()
```

Out[38]:

	LIMIT_BAL	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	...
1	20000	1	24	2	2	-1	-1	-2	-2	3913	...
2	120000	2	26	-1	2	0	0	0	2	2682	...
3	90000	2	34	0	0	0	0	0	0	29239	...
4	50000	1	37	0	0	0	0	0	0	46990	...
5	50000	1	57	-1	0	-1	0	0	0	8617	...

5 rows × 29 columns

In [39]:
```
ccredit.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2396 entries, 1 to 2397
Data columns (total 29 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   LIMIT_BAL                               2396 non-null   int32
 1   MARRIAGE                                2396 non-null   int32
 2   AGE                                     2396 non-null   int32
 3   PAY_0                                   2396 non-null   int32
 4   PAY_2                                   2396 non-null   int32
 5   PAY_3                                   2396 non-null   int32
 6   PAY_4                                   2396 non-null   int32
 7   PAY_5                                   2396 non-null   int32
 8   PAY_6                                   2396 non-null   int32
 9   BILL_AMT1                               2396 non-null   int32
 10  BILL_AMT2                               2396 non-null   int32
 11  BILL_AMT3                               2396 non-null   int32
 12  BILL_AMT4                               2396 non-null   int32
 13  BILL_AMT5                               2396 non-null   int32
 14  BILL_AMT6                               2396 non-null   int32
 15  PAY_AMT1                                2396 non-null   int32
 16  PAY_AMT2                                2396 non-null   int32
 17  PAY_AMT3                                2396 non-null   int32
 18  PAY_AMT4                                2396 non-null   int32
 19  PAY_AMT5                                2396 non-null   int32
 20  PAY_AMT6                                2396 non-null   int32
 21  SEX_female                              2396 non-null   uint8
 22  SEX_male                                2396 non-null   uint8
 23  EDUCATION_graduate school               2396 non-null   uint8
 24  EDUCATION_high school                   2396 non-null   uint8
 25  EDUCATION_other                         2396 non-null   uint8
 26  EDUCATION_university                    2396 non-null   uint8
 27  default payment next month_default      2396 non-null   uint8
 28  default payment next month_not default  2396 non-null   uint8
dtypes: int32(21), uint8(8)
memory usage: 298.5 KB
```

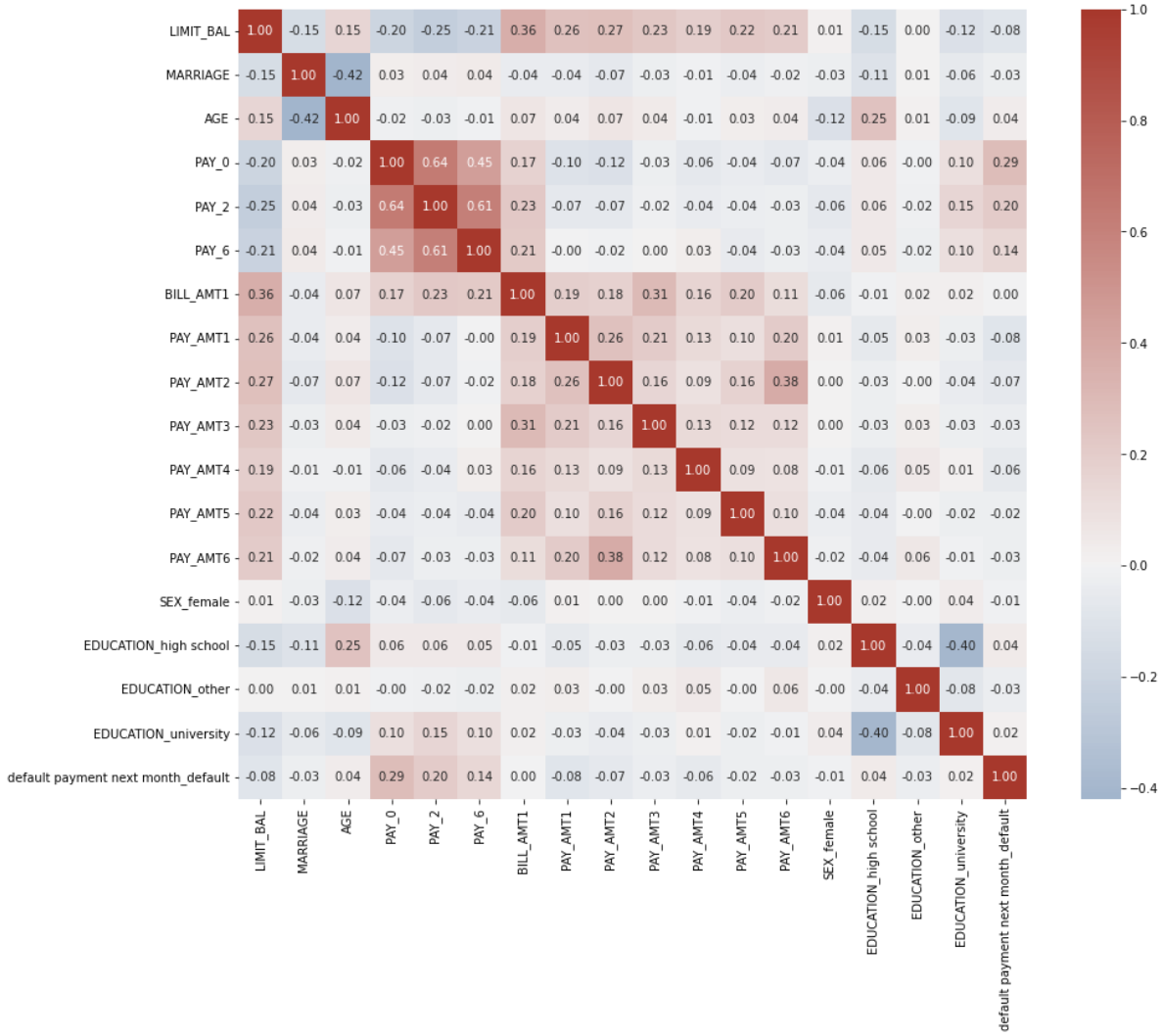
In [40]:
```
ccredit=ccredit.drop(['PAY_3','PAY_4', 'PAY_5', 'BILL_AMT2', 'BILL_AMT3', 'BIL
```

In [41]:
```
ccredit= ccredit.drop(['EDUCATION_graduate school'], axis=1)
```

In [42]:
```
ccredit.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2396 entries, 1 to 2397
Data columns (total 18 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   LIMIT_BAL                             2396 non-null   int32
 1   MARRIAGE                              2396 non-null   int32
 2   AGE                                   2396 non-null   int32
 3   PAY_0                                 2396 non-null   int32
 4   PAY_2                                 2396 non-null   int32
 5   PAY_6                                 2396 non-null   int32
 6   BILL_AMT1                             2396 non-null   int32
 7   PAY_AMT1                              2396 non-null   int32
 8   PAY_AMT2                              2396 non-null   int32
 9   PAY_AMT3                              2396 non-null   int32
 10  PAY_AMT4                              2396 non-null   int32
 11  PAY_AMT5                              2396 non-null   int32
 12  PAY_AMT6                              2396 non-null   int32
 13  SEX_female                            2396 non-null   uint8
 14  EDUCATION_high school                 2396 non-null   uint8
 15  EDUCATION_other                       2396 non-null   uint8
 16  EDUCATION_university                  2396 non-null   uint8
 17  default payment next month_default    2396 non-null   uint8
dtypes: int32(13), uint8(5)
memory usage: 216.6 KB
```

In [43]:
```
matrix= ccredit.corr()
plt.figure(figsize=(16,12))

cmap = sns.diverging_palette(250, 15, s=75, l=40,
                             n=9, center="light", as_cmap=True)

_ = sns.heatmap(matrix, center=0, annot=True,
                fmt='.2f', square=True, cmap=cmap)
```



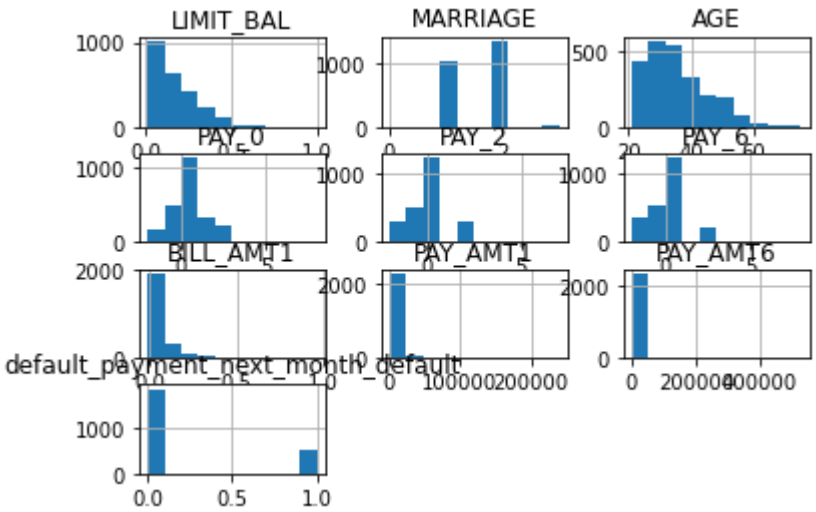
In [44]:
```
ccredit.rename({'default payment next month_default':'default_payment_next_mon
```

In [45]:
```
ccredit.info()
```

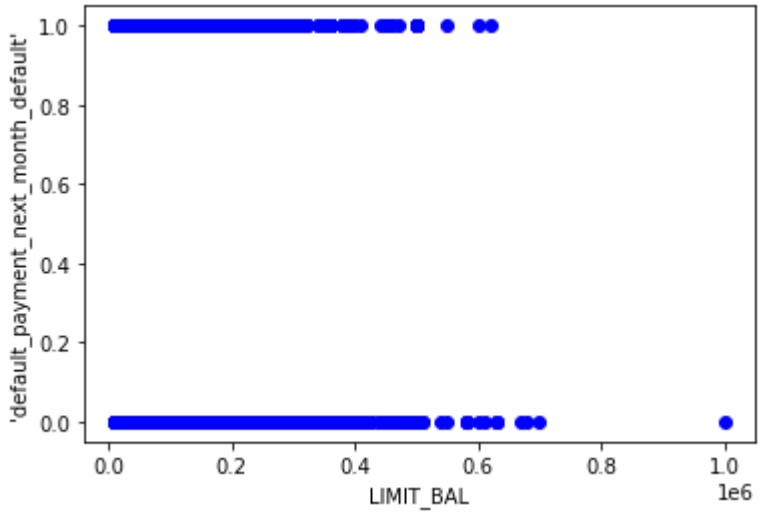
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2396 entries, 1 to 2397
Data columns (total 18 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   LIMIT_BAL                             2396 non-null   int32
 1   MARRIAGE                              2396 non-null   int32
 2   AGE                                   2396 non-null   int32
 3   PAY_0                                 2396 non-null   int32
 4   PAY_2                                 2396 non-null   int32
 5   PAY_6                                 2396 non-null   int32
 6   BILL_AMT1                             2396 non-null   int32
 7   PAY_AMT1                              2396 non-null   int32
 8   PAY_AMT2                              2396 non-null   int32
 9   PAY_AMT3                              2396 non-null   int32
 10  PAY_AMT4                              2396 non-null   int32
 11  PAY_AMT5                              2396 non-null   int32
 12  PAY_AMT6                              2396 non-null   int32
 13  SEX_female                            2396 non-null   uint8
 14  EDUCATION_high school                 2396 non-null   uint8
 15  EDUCATION_other                       2396 non-null   uint8
 16  EDUCATION_university                  2396 non-null   uint8
 17  default_payment_next_month_default    2396 non-null   uint8
dtypes: int32(13), uint8(5)
memory usage: 216.6 KB
```

- Distribucion de los valores de cada variable

In [47]:
```
viz = ccredit[['LIMIT_BAL', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2', 'PAY_6', 'BIL
viz.hist()
plt.show()
```



In [49]:
```
plt.scatter(ccredit.LIMIT_BAL, ccredit.default_payment_next_month_default,  co
plt.xlabel("LIMIT_BAL")
plt.ylabel("'default_payment_next_month_default'")
plt.show()
```



In [50]:
```
#Variables que generan dispersion son: 'LIMIT_BAL', 'AGE', 'BILL_AMT1', 'PAY_A
```

In [51]:
```
#No es viable utilizar la variable "Y" o "default_payment_next_month_default" p
```

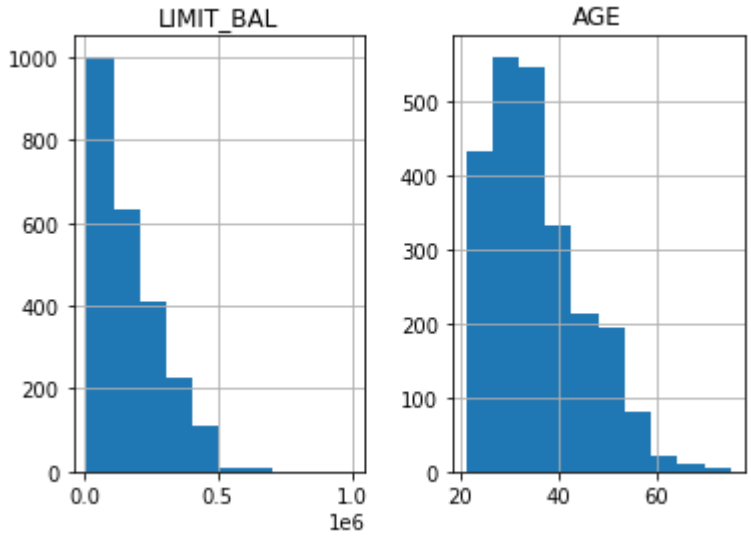
Modelo de regresion lineal simple

In [62]:
```
cdf=ccredit[['LIMIT_BAL', 'AGE']]
cdf.head()
```

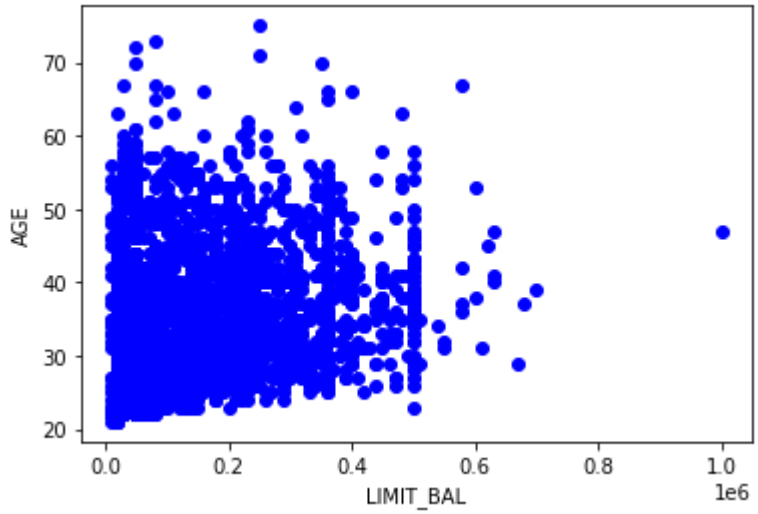
Out[62]:

	LIMIT_BAL	AGE
1	20000	24
2	120000	26
3	90000	34
4	50000	37
5	50000	57

In [56]:
```
viz = cdf[['LIMIT_BAL', 'AGE']]
viz.hist()
plt.show()
```



In [57]:
```
plt.scatter(cdf.LIMIT_BAL, cdf.AGE,  color='blue')
plt.xlabel("LIMIT_BAL")
plt.ylabel("AGE")
plt.show()
```



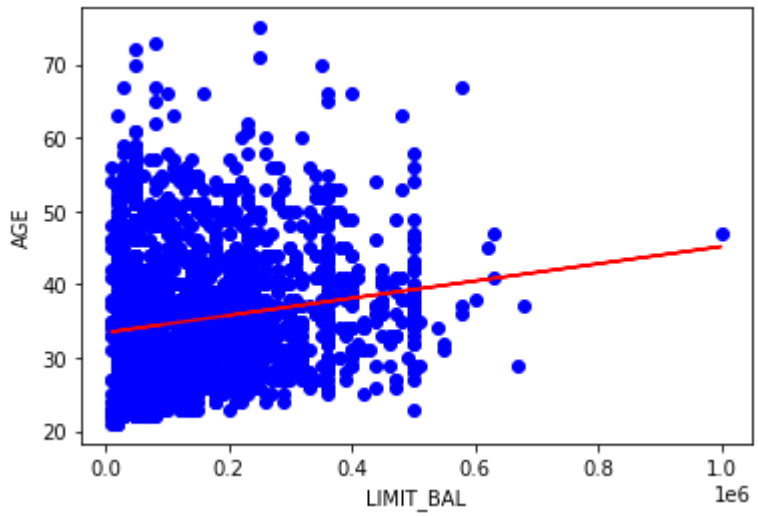
In [63]:
```
import numpy as np
msk = np.random.rand(len(ccredit)) < 0.8 #Esto selecciona aleatoriamente el 80%
train = cdf[msk] #aca se indica que el set de entrenamiento esta conformado por
test = cdf[~msk] #aca se especifica que el set de prueba esta conformado por e
```

In [65]:
```
#Generacion del modelo de regresion lineal
from sklearn import linear_model
regr = linear_model.LinearRegression()
train_x = np.asanyarray(train[['LIMIT_BAL']])
train_y = np.asanyarray(train[['AGE']])
regr.fit (train_x, train_y)

#Obtener los coeficientes de resultados, datos informativos
print ('Coefficients: ', regr.coef_)
print ('Intercept: ', regr.intercept_)
```

```
Coefficients:  [[1.17193084e-05]]
Intercept:  [33.44405189]
```

In [66]:
```
plt.scatter(train.LIMIT_BAL, train.AGE,  color='blue')
plt.plot(train_x, regr.coef_[0][0]*train_x + regr.intercept_[0], '-r')
plt.xlabel('LIMIT_BAL')
plt.ylabel('AGE')
```

Out[66]: Text(0, 0.5, 'AGE')



In [67]:
```
from sklearn.metrics import r2_score

test_x = np.asanyarray(test[['LIMIT_BAL']])
test_y = np.asanyarray(test[['AGE']])
test_y_ = regr.predict(test_x)

print("Mean absolute error: %.2f" % np.mean(np.absolute(test_y_ - test_y)))
print("Residual sum of squares (MSE): %.2f" % np.mean((test_y_ - test_y) ** 2))
print("R2-score: %.2f" % r2_score(test_y , test_y_) )
```

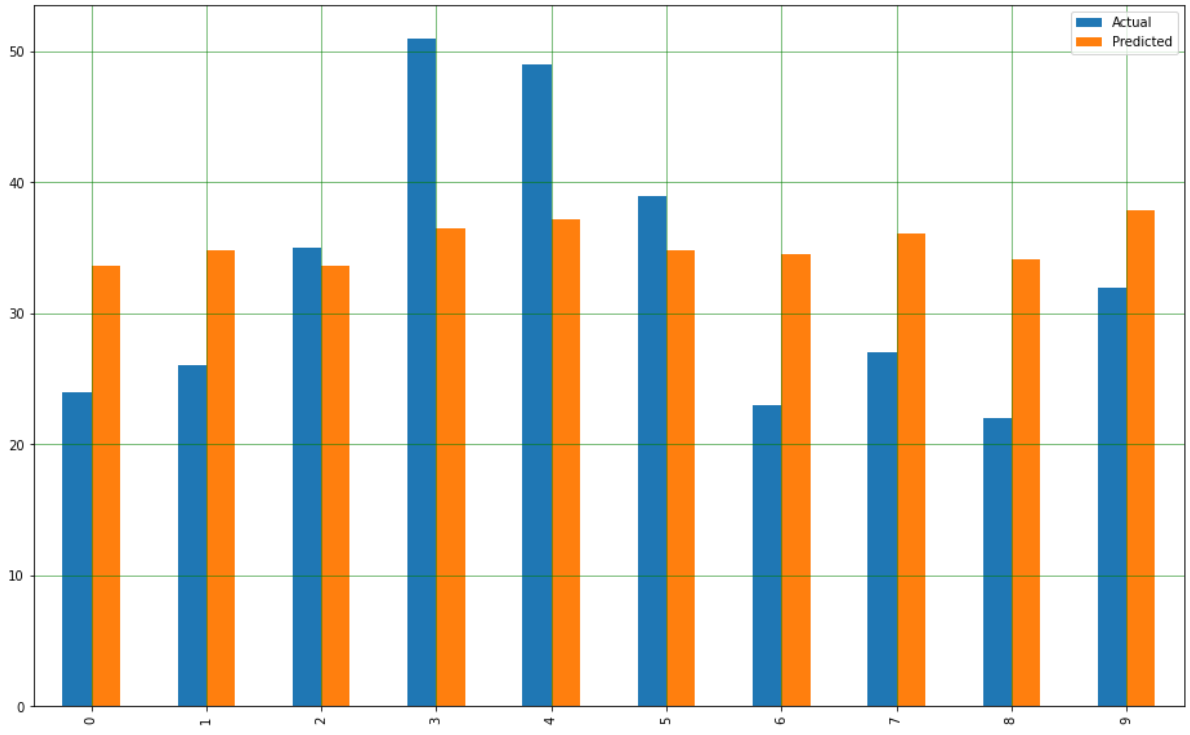
```
Mean absolute error: 7.47
Residual sum of squares (MSE): 81.02
R2-score: 0.02
```

In [68]:
```
df = pd.DataFrame({'Actual': test_y.flatten(), 'Predicted': test_y_.flatten()})
df.head(10)
```

Out[68]:

	Actual	Predicted
0	24	33.678438
1	26	34.850369
2	35	33.678438
3	51	36.491072
4	49	37.194231
5	39	34.850369
6	23	34.498790
7	27	36.139493
8	22	34.147210
9	32	37.897389

In [69]:
```
df1 = df.head(10)
df1.plot(kind='bar',figsize=(16,10))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```



In [ ]:
```

```