

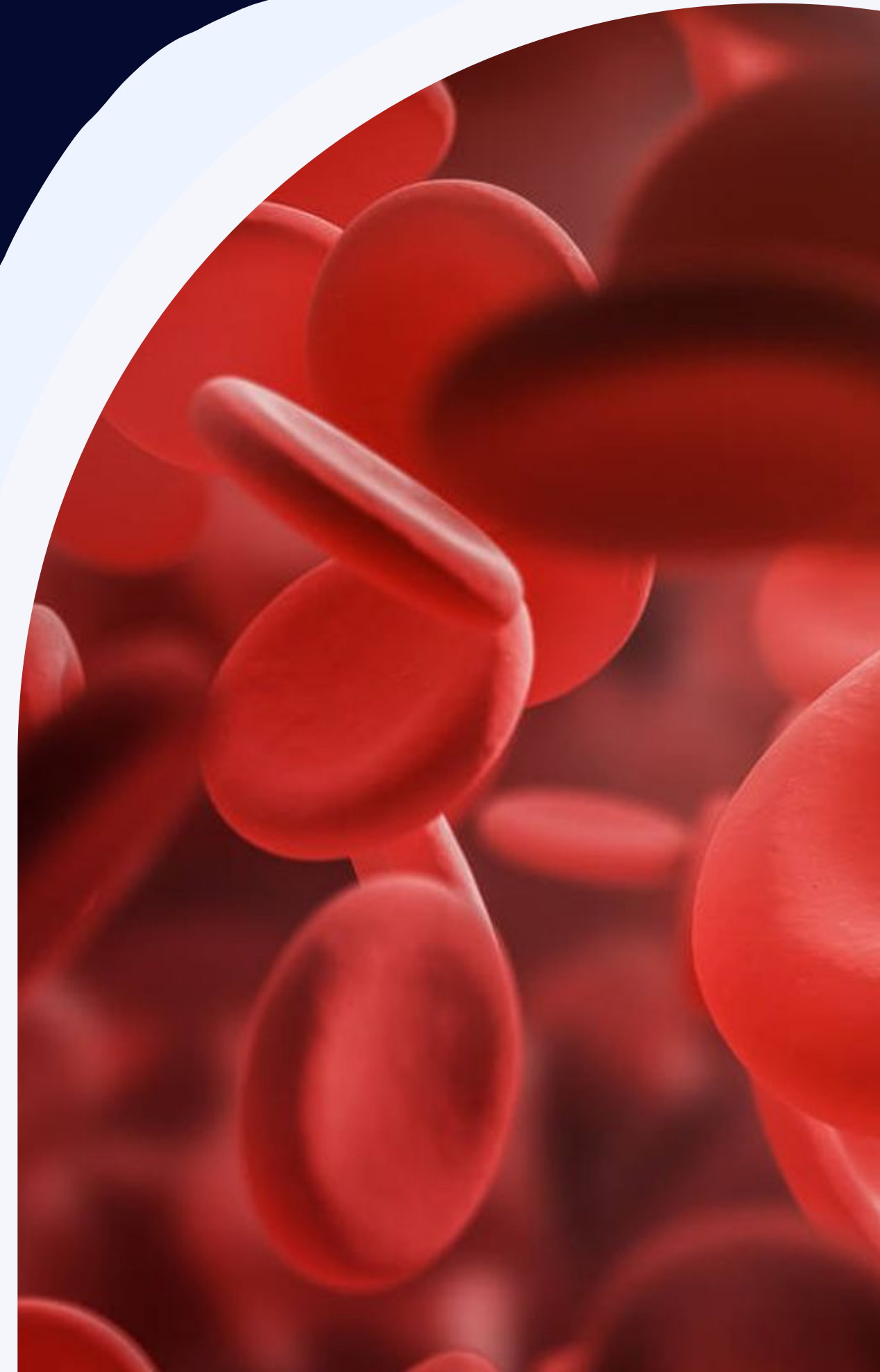
PREDICCIÓN DE ANEMIA

Presentado por:

Sebastián David Mendoza Alvarado- 2191969

Dilan Alessandro Corredor Díaz- 2191976

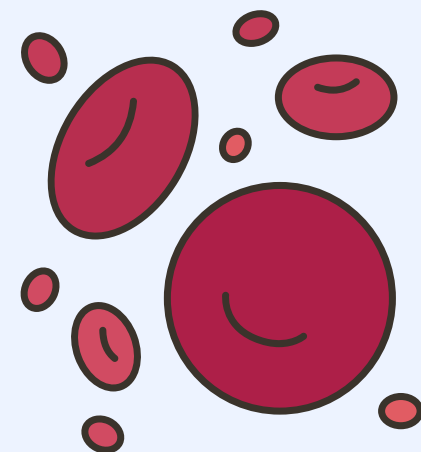
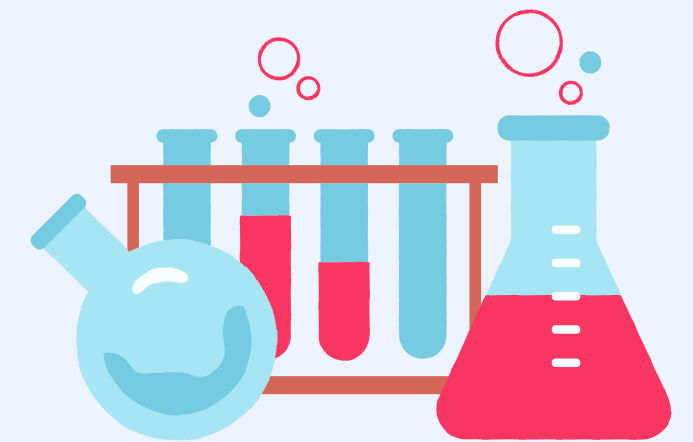
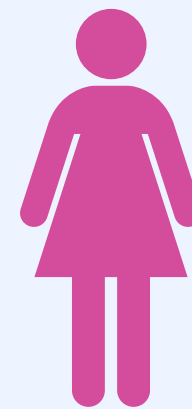
Juan Diego claro guerrero- 2192508



INFORMACIÓN GENERAL

DATOS

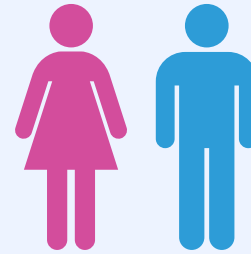
- Género
- Hemoglobina
- MCHC
- MCV
- MCH
- Resultados



Detección correcta de la
enfermedad
Anemia

DATA SET

Género: 0 - masculino, 1 - femenino

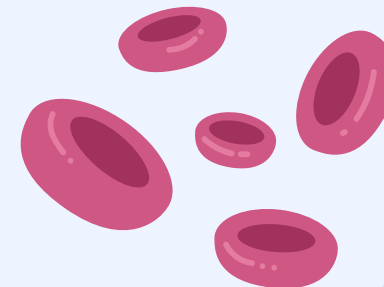


Hemoglobina:

Proteína en los glóbulos rojos que transporta oxígeno a los órganos y tejidos del cuerpo y transporta dióxido de carbono desde los órganos y tejidos de regreso a los pulmones.

MCH

Hemoglobina corpuscular media. Es la cantidad promedio en cada uno de sus glóbulos rojos de una proteína llamada hemoglobina, que transporta oxígeno por todo el cuerpo.



MCHC

Concentración de hemoglobina corpuscular media. Es una medida de la concentración promedio de hemoglobina dentro de un solo glóbulo rojo.

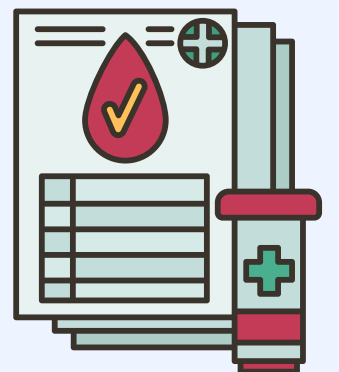
MCV

Volumen corpuscular medio. Un análisis de sangre MCV mide el tamaño promedio de sus glóbulos rojos.



Resultados:

0- no anémico, 1-anémico



EXPLORACIÓN DE LOS DATOS

MEDIA

```
data_anemia = pd.read_csv('anemia.csv')
genero = data_anemia.Gender
hemoglobina = data_anemia.Hemoglobin
mch = data_anemia.MCH
mchc = data_anemia.MCHC
mcv = data_anemia.MCV
resultado = data_anemia.Result

print('Genero:' + ' ' + str(np.mean(genero)))
print('hemoglobina:' + ' ' + str(np.mean(hemoglobina)))
print('MCH:' + ' ' + str(np.mean(mch)))
print('MCHC:' + ' ' + str(np.mean(mchc)))
print('MCV:' + ' ' + str(np.mean(mcv)))
print('Resultado:' + ' ' + str(np.mean(resultado)))
```

```
Genero: 0.5207600281491908
hemoglobina: 13.412737508796623
MCH: 22.90562983814215
MCHC: 30.251231527093594
MCV: 85.5237860661506
Resultado: 0.4363124560168895
```

MEDIANA

```
print('Genero:' + ' ' + str(np.median(genero)))
print('hemoglobina:' + ' ' + str(np.median(hemoglobina)))
print('MCH:' + ' ' + str(np.median(mch)))
print('MCHC:' + ' ' + str(np.median(mchc)))
print('MCV:' + ' ' + str(np.median(mcv)))
print('Resultado:' + ' ' + str(np.median(resultado)))
```

```
Genero: 1.0
hemoglobina: 13.2
MCH: 22.7
MCHC: 30.4
MCV: 85.3
Resultado: 0.0
```

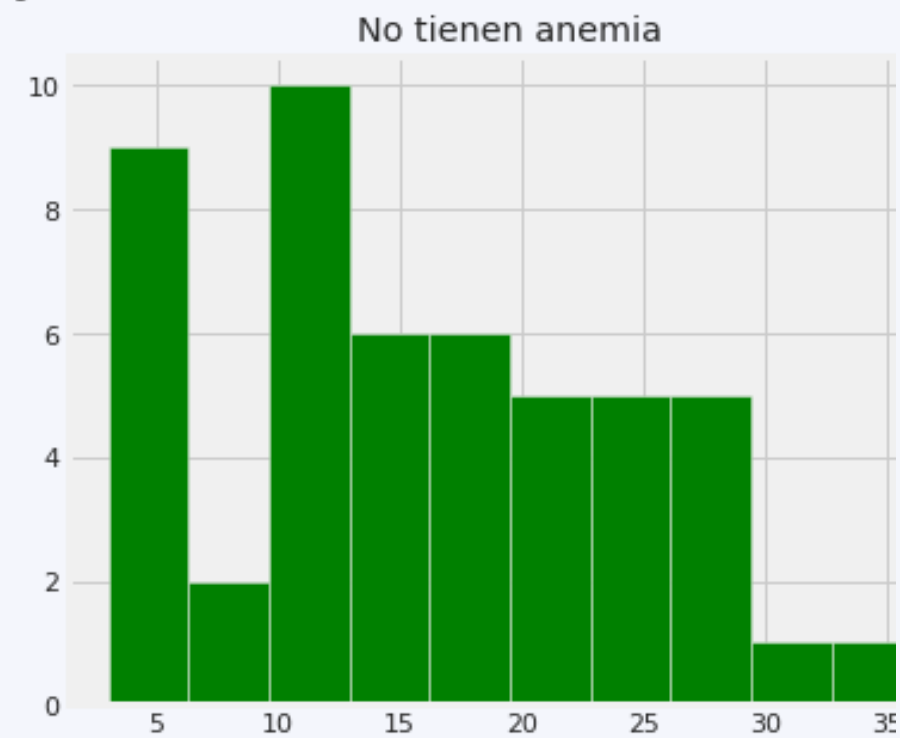
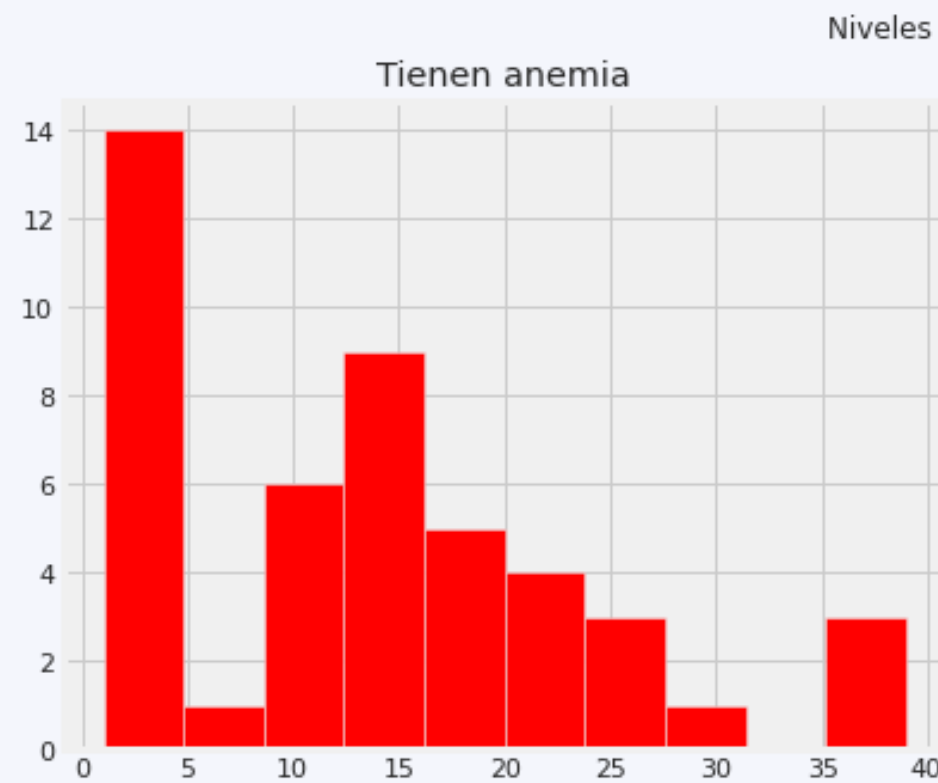
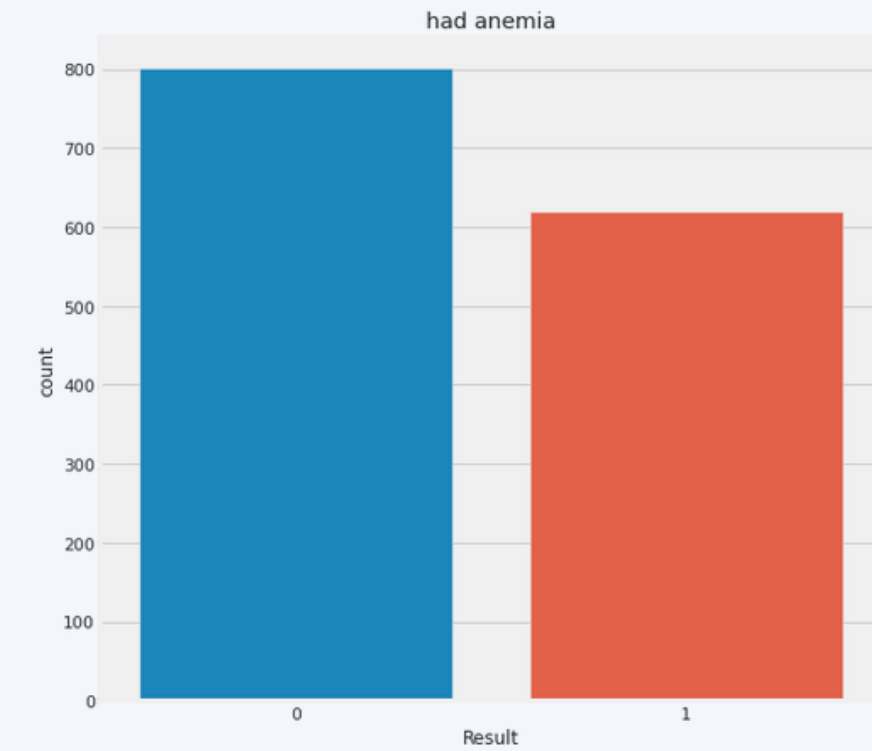
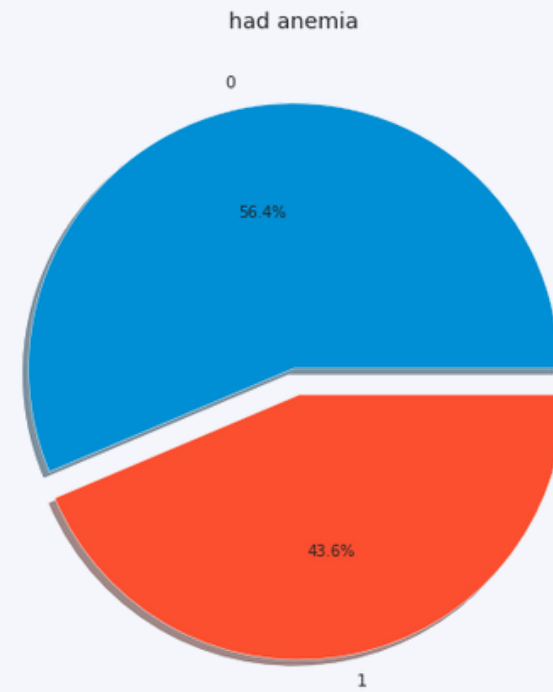
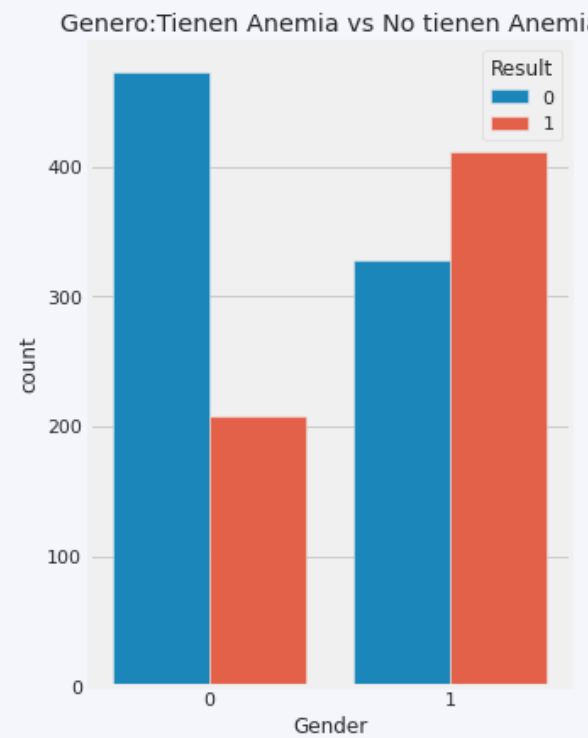
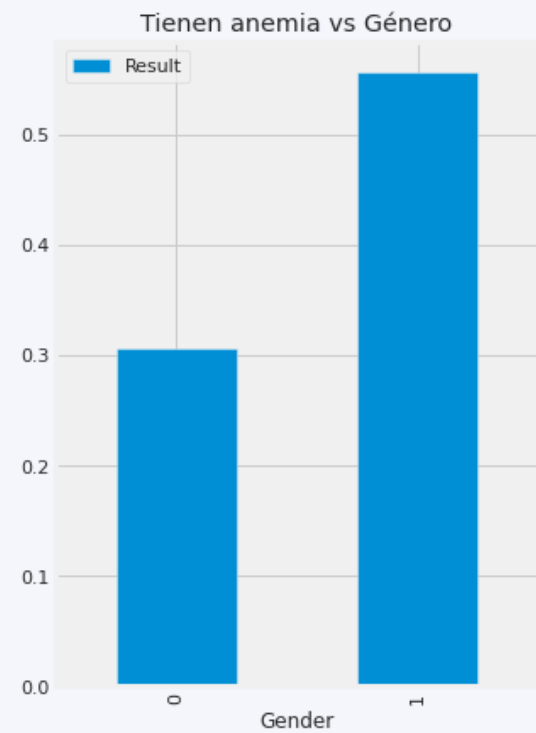
MODA

```
import statistics as stat

print('Genero:' + ' ' + str(stat.mode(genero)))
print('hemoglobina:' + ' ' + str(stat.mode(hemoglobina)))
print('MCH:' + ' ' + str(stat.mode(mch)))
print('MCHC:' + ' ' + str(stat.mode(mchc)))
print('Resultado:' + ' ' + str(stat.mode(resultado)))
```

```
Genero: 1
hemoglobina: 11.5
MCH: 24.3
MCHC: 32.2
Resultado: 0
```

GRÁFICOS MUESTRALES

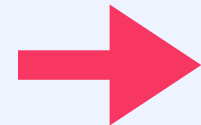


PRUEBA Y ENTRENAMIENTO

```
#@title **code** División de los datos en conjuntos de prueba y entrenamiento
train,test=train_test_split(data_anemia,test_size=0.3,random_state=0,stratify=data_anemia['Result'])
train_X=train[train.columns[:-1]]
train_Y=train[train.columns[-1:]]
test_X=test[test.columns[:-1]]
test_Y=test[test.columns[-1:]]
X=data_anemia[data_anemia.columns[:-1]]
Y=data_anemia['Result']
len(train_X), len(train_Y), len(test_X), len(test_Y)
```

CLASIFICACIÓN DE COMPONENTES TENIENDO EN CUENTA CIERTOS VALORES

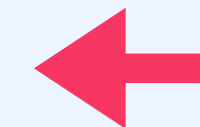
GAUSSIANNB



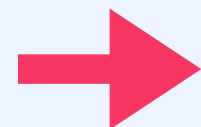
```
model=GaussianNB()  
model.fit(train_X,train_Y)  
prediction6=model.predict(test_X)  
print('La precisión de GaussianNB es:', metrics.accuracy_score(prediction6,test_Y))
```

```
model=DecisionTreeClassifier()  
model.fit(train_X,train_Y)  
prediction4=model.predict(test_X)  
print('La precisión del árbol de decisión es',metrics.accuracy_score(prediction4,test_Y))
```

**DECISION TREE
CLASSIFIER
(APLICACION)**



**RANDOM FOREST
CLASSIFIER (RFC)**



```
model=RandomForestClassifier(n_estimators=100)  
model.fit(train_X,train_Y)  
prediction7=model.predict(test_X)  
print('La precisión de los Random Forests es',metrics.accuracy_score(prediction7,test_Y))
```

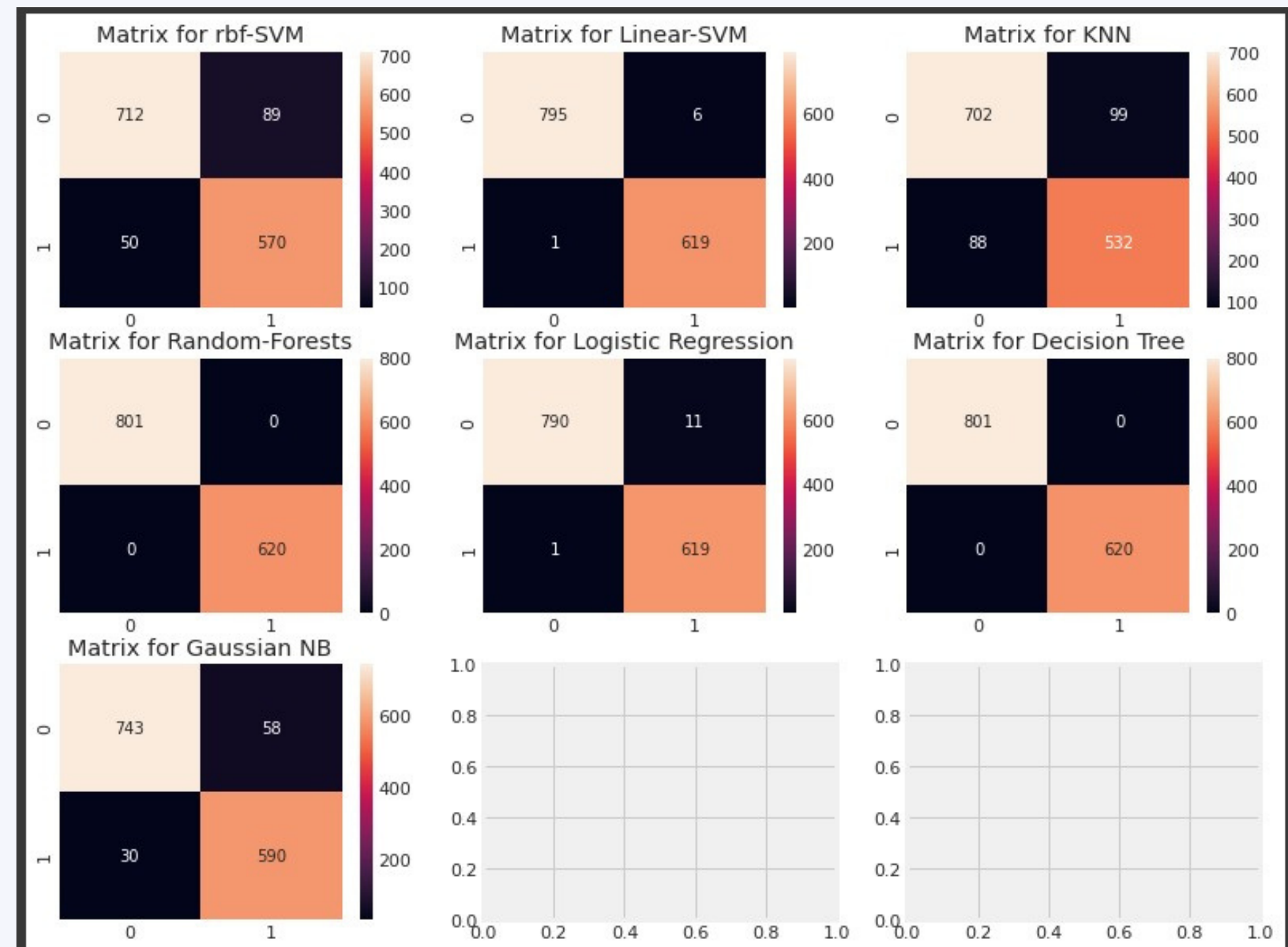
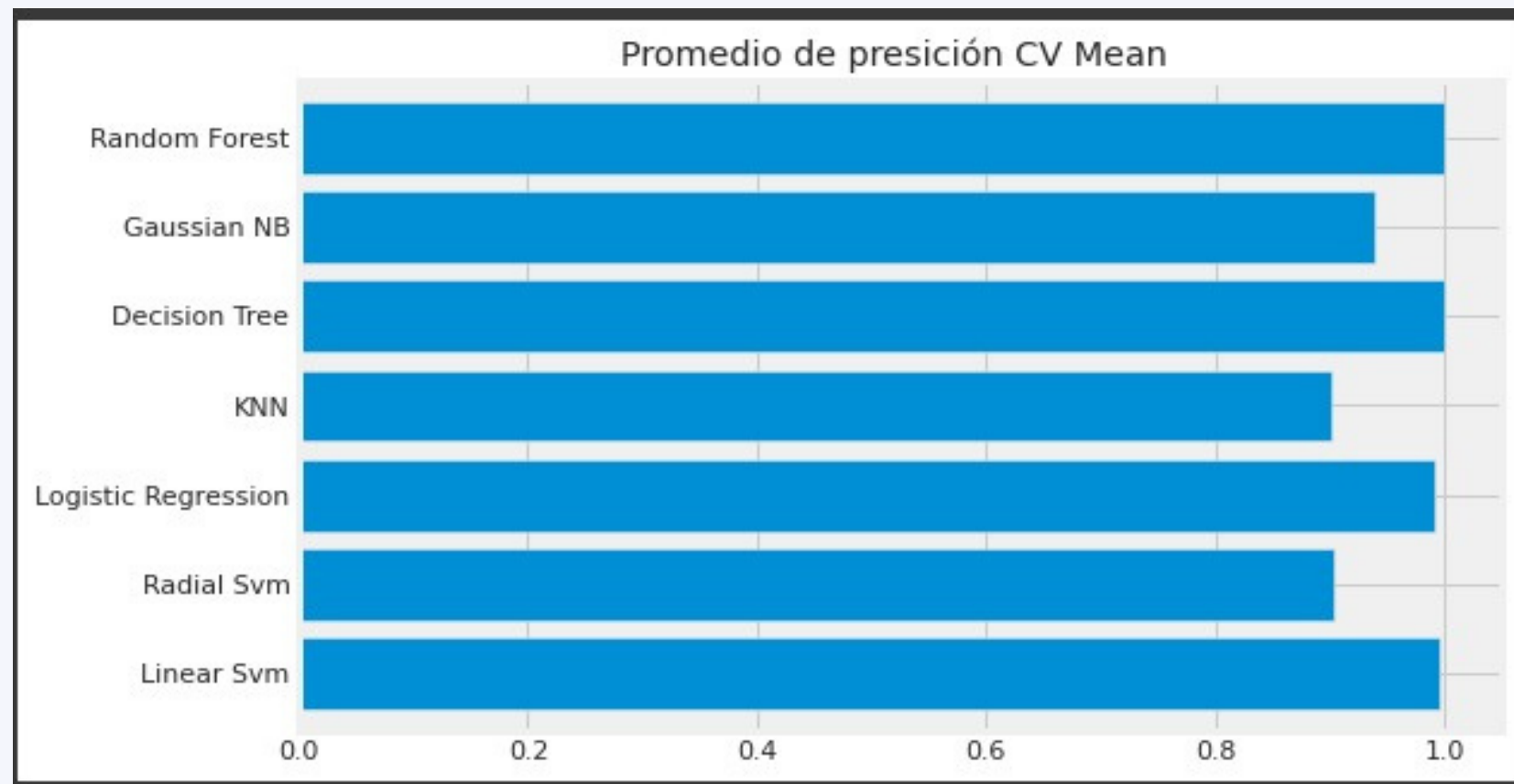
```
model=svm.SVC(kernel='linear',C=0.1,gamma=0.1)  
model.fit(train_X,train_Y)  
prediction2=model.predict(test_X)  
print('La precisión para SVM lineal es',metrics.accuracy_score(prediction2,test_Y))
```

**SUPPORT VECTOR
MACHINE
(APLICACION)**



VALIDACIÓN Y COMPARACIÓN DE LOS METÓDOS DE CLASIFICACIÓN

| | CV Mean | Std |
|---------------------|----------|----------|
| Linear Svm | 0.994366 | 0.008213 |
| Radial Svm | 0.902191 | 0.019214 |
| Logistic Regression | 0.991554 | 0.007586 |
| KNN | 0.900079 | 0.028450 |
| Decision Tree | 1.000000 | 0.000000 |
| Gaussian NB | 0.938782 | 0.023560 |
| Random Forest | 1.000000 | 0.000000 |



GRACIAS