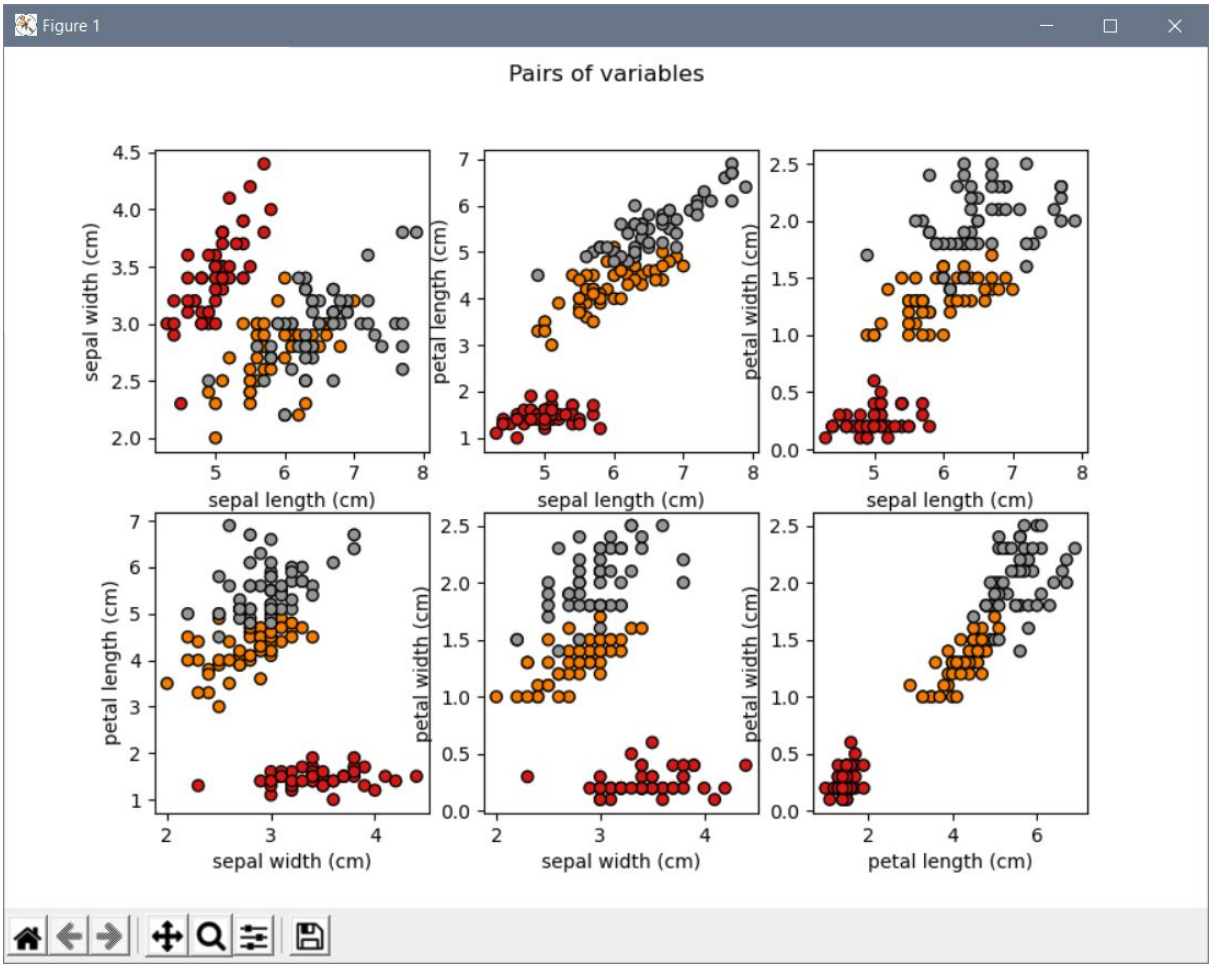
Sergio Omar Alvarado Pérez
Guillermo Herrera Acosta
Juan Antonio Melendres Villa
Ricardo Palma Mendoza

Problemas de clasificación

Problema 1

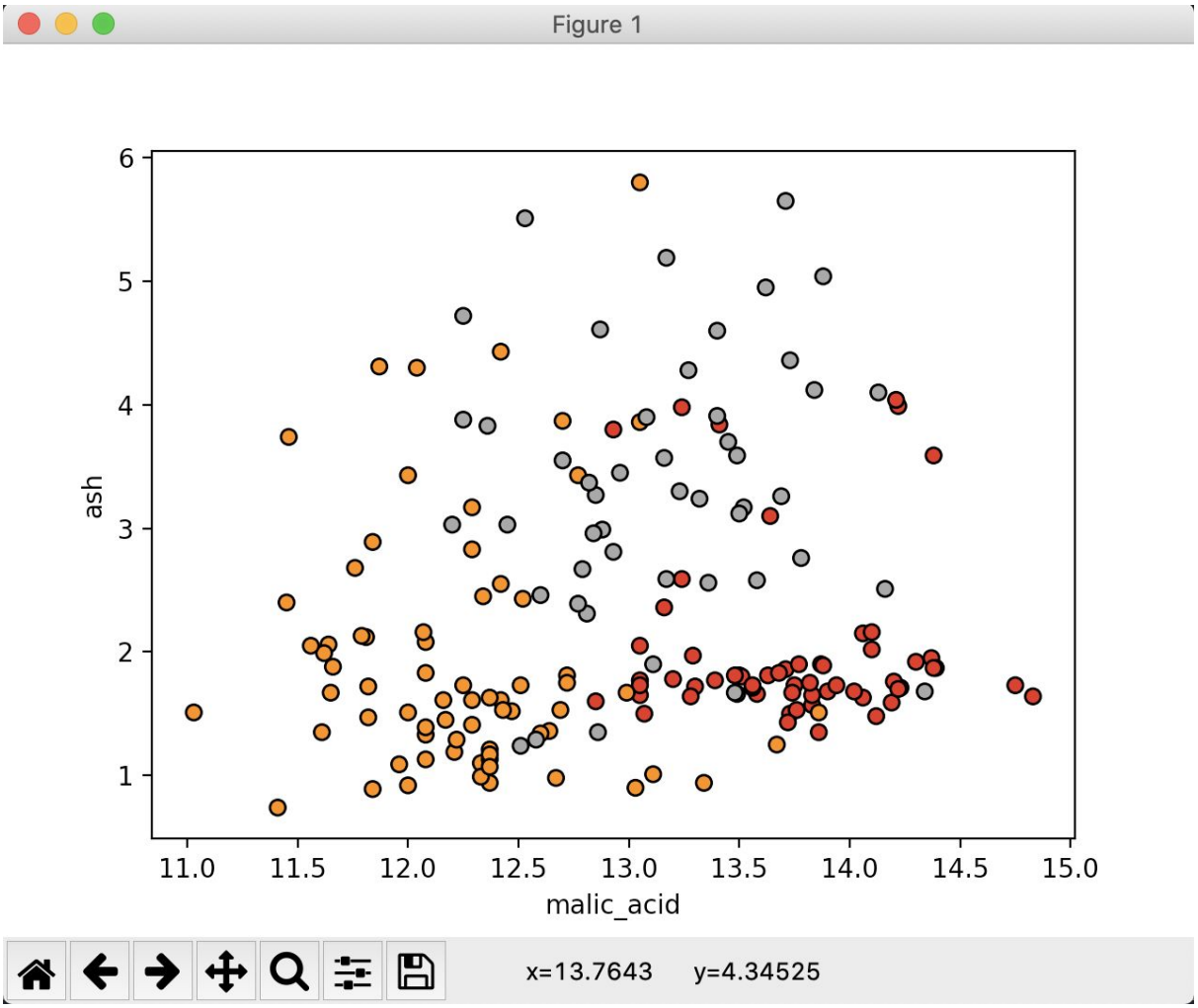


Aquí se muestran los datos de las 3 categorías graficados en conjuntos de dos variables mostrando que las categorías gris y naranja no son fáciles de diferenciar en algunos casos.

```
New Observation: [[6.038120556987795, 3.7634755721807838, 2.754347150706594, 2.697134600686991], [4.771811590486362, 4.5693564493241
Prediction for a new observation [1 2 1 1 1 2 1]
ACC =  0.9800000000000001
PRECISION1 =  1.0
PRECISION2 =  1.0
PRECISION3 =  0.9466666666666667
RECALL1 =  1.0
RECALL2 =  0.9349999999999999
RECALL3 =  1.0
```

Usando el algoritmo de SVM con kernel lineal se obtuvieron los resultados mostrados en la captura, estos resultados muestran un accuracy promedio muy alto al igual que la precisión y el recall, lo que es muy bueno.

Problema 2 (Carga la base de datos Wine)



¿Cuántas variables y observaciones por clase hay en este conjunto de datos? Hay 3 clases las cuales tienen 13 variables, que tienen diferentes indicadores.

¿Que representan los predictores?

- Alcohol
- Malic acid
- Ash
- Alcalinity of ash
- Magnesium
- Total phenols
- Flavanoids
- Nonflavanoid phenols
- Proanthocyanins
- Color intensity

- Hue
- OD280/OD315 of diluted wines
- Proline

	Min	Max	Mean	SD
Alcohol:	11.0	14.8	13.0	0.8
Malic Acid:	0.74	5.80	2.34	1.12
Ash:	1.36	3.23	2.36	0.27
Alcalinity of Ash:	10.6	30.0	19.5	3.3
Magnesium:	70.0	162.0	99.7	14.3
Total Phenols:	0.98	3.88	2.29	0.63
Flavanoids:	0.34	5.08	2.03	1.00
Nonflavanoid Phenols:	0.13	0.66	0.36	0.12
Proanthocyanins:	0.41	3.58	1.59	0.57
Colour Intensity:	1.3	13.0	5.1	2.3
Hue:	0.48	1.71	0.96	0.23
OD280/OD315 of diluted wines:	1.27	4.00	2.61	0.71
Proline:	278	1680	746	315

Calcula la exactitud de los siguientes modelos de clasificación con k-fold cross validation (k = 5) para la base de datos Wine.

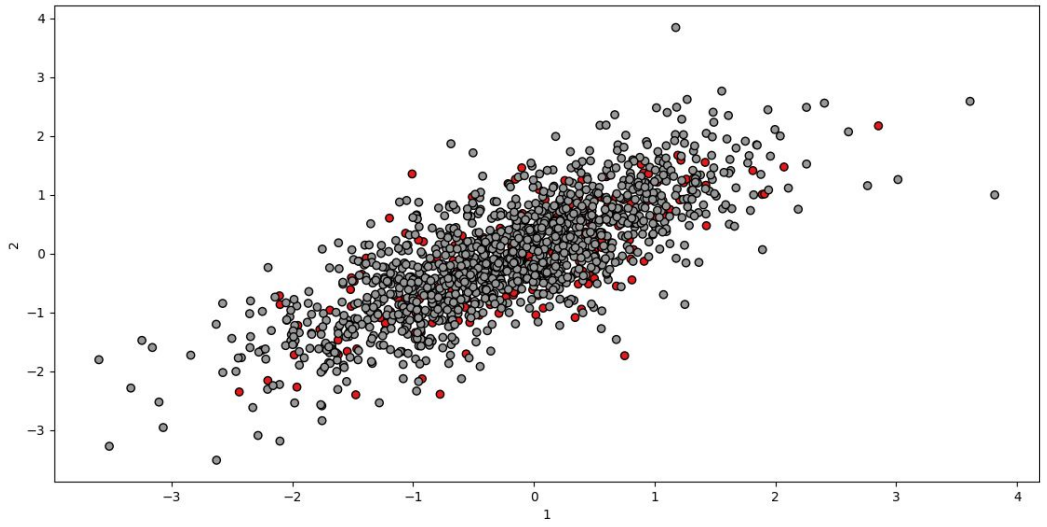
```
SVM - lineal

Prediction for a new observation [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2]
acc =  0.9722222222222222
Prediction for a new observation [0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2]
acc =  1.0
Prediction for a new observation [0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2]
acc =  0.9722222222222222
Prediction for a new observation [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
acc =  1.0
Prediction for a new observation [0 0 0 0 0 0 0 0 0 2 1 1 1 1 1 1 0 1 2 1 1 1 1 1 0 1 1 1 2 2 2 2 2 2 2]
acc =  0.8571428571428571
ACC =  0.9603174603174602

SVM de base radial

Prediction for a new observation [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1]
acc =  0.6666666666666666
Prediction for a new observation [0 0 0 0 0 2 0 0 0 0 0 0 2 2 2 1 2 1 2 2 1 2 1 2 1 1 1 1 1 1 2 2 1 2 2 1]
acc =  0.6944444444444444
Prediction for a new observation [2 0 0 0 0 2 0 0 0 0 0 0 0 2 2 0 0 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 2 1 1]
acc =  0.6111111111111112
Prediction for a new observation [0 0 1 0 0 0 0 0 1 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0]
acc =  0.6285714285714286
Prediction for a new observation [0 0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1]
acc =  0.6285714285714286
ACC =  0.6458730158730159
```

```
k-NN (para k = 3)

Prediction for a new observation [0 0 0 2 0 0 0 0 0 2 0 0 1 2 0 0 2 2 1 2 2 1 1 1 2 0 2 1 2 0 0 1 2 2 2 2]
acc =  0.5277777777777778
Prediction for a new observation [0 0 0 0 2 2 0 1 1 0 0 0 0 0 0 0 1 1 1 2 1 1 1 2 1 1 1 1 1 0 1 1 1 0 2 2]
acc =  0.6944444444444444
Prediction for a new observation [0 0 0 0 2 0 0 0 0 1 1 2 1 1 1 1 0 2 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1 1 2]
acc =  0.7777777777777778
Prediction for a new observation [1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 2 1 1 2 1 1 2 2 2 2 2 1 1 2 2 2 0]
acc =  0.7714285714285715
Prediction for a new observation [0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 1 1 2 1 1 2 2 2 1 2 1 1 2 1 2 1 0]
acc =  0.7142857142857143
ACC =  0.6971428571428572

Árbol de decisión

Prediction for a new observation [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2]
acc =  0.9722222222222222
Prediction for a new observation [0 0 0 0 0 0 1 0 0 0 2 0 1 1 1 1 0 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2]
acc =  0.8888888888888888
Prediction for a new observation [0 0 0 0 0 1 0 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2]
acc =  0.9166666666666666
Prediction for a new observation [1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 2 1 1 1 0 1 1 1 0 0 0 1 0 2 2 2 2 2 2 2]
acc =  0.7428571428571429
Prediction for a new observation [0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 2 1 1 1 0 1 1 1 2 1 1 2 1 2 2 2 2 2 2 2 2]
acc =  0.8
ACC =  0.8641269841269841
```

Indique con qué clasificador se obtuvieron los mejores resultados: Con SVM - lineal es el clasificador en el cual se obtuvieron los mejores datos. Muy cerca de él se encuentra el árbol de decisión.

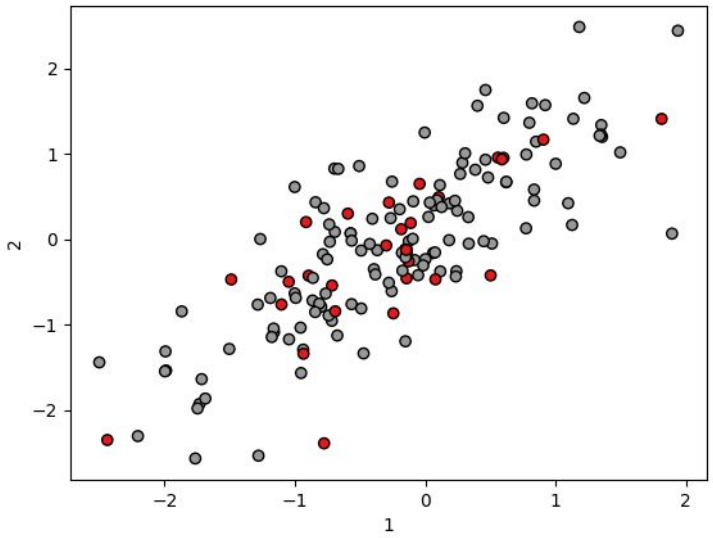
Problema 3

1. **Carga la base de datos misteriosa, y calcula la exactitud de un clasificador SVM lineal con k-fold cross validation (k = 5). En este archivo, la primera columna corresponde a la etiqueta o clase (1 o 2), y el resto de las columnas son variables o predictores.**



```
sample size 1587 ( 100 % of total data )
Prediction for a new observation [2.]
acc =  0.8867924528301887
acc =  0.8710691823899371
acc =  0.8958990536277602
acc =  0.8958990536277602
acc =  0.8958990536277602
AVG ACC =  0.8891117592206813
```

2. **Selecciona aleatoriamente el 10% de todas las observaciones de la base de datos misteriosa, y repite la evaluación del clasificador con k-fold cross validation pero con los datos seleccionados.**




```
sample size 159 ( 10.0 % of total data )
Prediction for a new observation [2.]
acc =  0.78125
acc =  0.875
acc =  0.75
acc =  0.84375
acc =  0.7741935483870968
AVG ACC =  0.8048387096774194
```

3. **Repite el paso anterior, pero con 20%, 30%, 40%, ..., 90% de los datos.**

```
sample size 317 ( 20.0 % of total data )
Prediction for a new observation [2.]
acc =  0.796875
acc =  0.765625
acc =  0.7936507936507936
acc =  0.8571428571428571
acc =  0.7936507936507936
AVG ACC =  0.8013888888888889
```

```
sample size 476 ( 30.0 % of total data )
Prediction for a new observation [1.]
acc =  0.8020833333333334
acc =  0.8105263157894737
acc =  0.8631578947368421
acc =  0.8526315789473684
acc =  0.8736842105263158
AVG ACC =  0.8404166666666667
```

```
sample size 635 ( 40.0 % of total data )
Prediction for a new observation [2.]
acc =  0.8503937007874016
acc =  0.8740157480314961
acc =  0.8503937007874016
acc =  0.8110236220472441
acc =  0.8110236220472441
```

```
sample size 794 ( 50.0 % of total data )
Prediction for a new observation [2.]
acc =  0.8679245283018868
acc =  0.8805031446540881
acc =  0.9182389937106918
acc =  0.8742138364779874
acc =  0.8481012658227848
AVG ACC =  0.8777963537934879
```

```
sample size 952 ( 60.0 % of total data )
Prediction for a new observation [2.]
acc =  0.8848167539267016
acc =  0.8743455497382199
acc =  0.8578947368421053
acc =  0.8894736842105263
acc =  0.8473684210526315
AVG ACC =  0.870779829154037
```

```
sample size 1111 ( 70.0 % of total data )
Prediction for a new observation [2.]
acc =  0.8968609865470852
acc =  0.9054054054054054
acc =  0.8153153153153153
acc =  0.8738738738738738
acc =  0.8693693693693694
AVG ACC =  0.8721649901022099
```

```
sample size 1270 ( 80.0 % of total data )
Prediction for a new observation [2.]
acc =  0.8818897637795275
acc =  0.8976377952755905
acc =  0.8622047244094488
acc =  0.905511811023622
acc =  0.905511811023622
AVG ACC =  0.8905511811023622
```

```
sample size 1428 ( 90.0 % of total data )
Prediction for a new observation [2.]
acc =  0.8881118881118881
acc =  0.8916083916083916
acc =  0.9090909090909091
acc =  0.8666666666666667
acc =  0.8982456140350877
AVG ACC =  0.8907446939025887
```

4. **De acuerdo a los resultados del apartado anterior, ¿cuántas observaciones creen que son necesarias para entrenar el modelo para el tipo de datos probados?**

   Entre el 50% y el 70% la variación de exactitud era mínima dada la aleatoriedad de los datos aunque que a partir del 70% en adelante la exactitud llegó a sus valores máximos con variación mínima. Por lo que escoger el 70% de los datos en adelante es lo necesario para entrenar el modelo para el tipo de datos probados.

Problema 4

```
→  Problemas de clasificación git:(practica_2_problema4) ✗ python3 problema4.py
'K' Neighbors
{1: {'ACC': 0.7996210543023233, 'RECALL': array([0.3099404 , 0.89705141])},
 2: {'ACC': 0.7637045413963455, 'RECALL': array([0.4853375, 0.8195445])},
 3: {'ACC': 0.8481320556316092, 'RECALL': array([0.23567386, 0.9704787 ])},
 4: {'ACC': 0.838073130567625, 'RECALL': array([0.36421605, 0.93265905])},
 5: {'ACC': 0.85130051782632, 'RECALL': array([0.19241507, 0.98340807])},
 6: {'ACC': 0.8575799059579786, 'RECALL': array([0.25832528, 0.97739405])},
 7: {'ACC': 0.8532071503680336, 'RECALL': array([0.15435909, 0.99321218])},
 8: {'ACC': 0.8626411126321847, 'RECALL': array([0.23544992, 0.98794069])},
 9: {'ACC': 0.8512846457552129, 'RECALL': array([0.12532396, 0.99625143])},
 10: {'ACC': 0.8544590599765888, 'RECALL': array([0.164583  , 0.99163416])}}
```

Dados los siguientes resultados donde el número de vecinos (rango del 1 al 10) va cambiando, podemos visualizar que se mantiene una accuracy constante desde ‘K=3’ hasta ‘K=8’ donde después se mantiene en una tasa de ~.85.

Dado este problema podemos ver que en el rango de k {3~8} es una buena medida.

El recall de la misma manera llega a mantener una buena tasa constante en ese mismo rango del valor de K.

The following recall responds to the metrics taken, its visualized the high metric with the data obtained with ‘k’ greater than 3 maintaining a constant value from .97 to .99.