Cas Kaggle: Mushroom classification

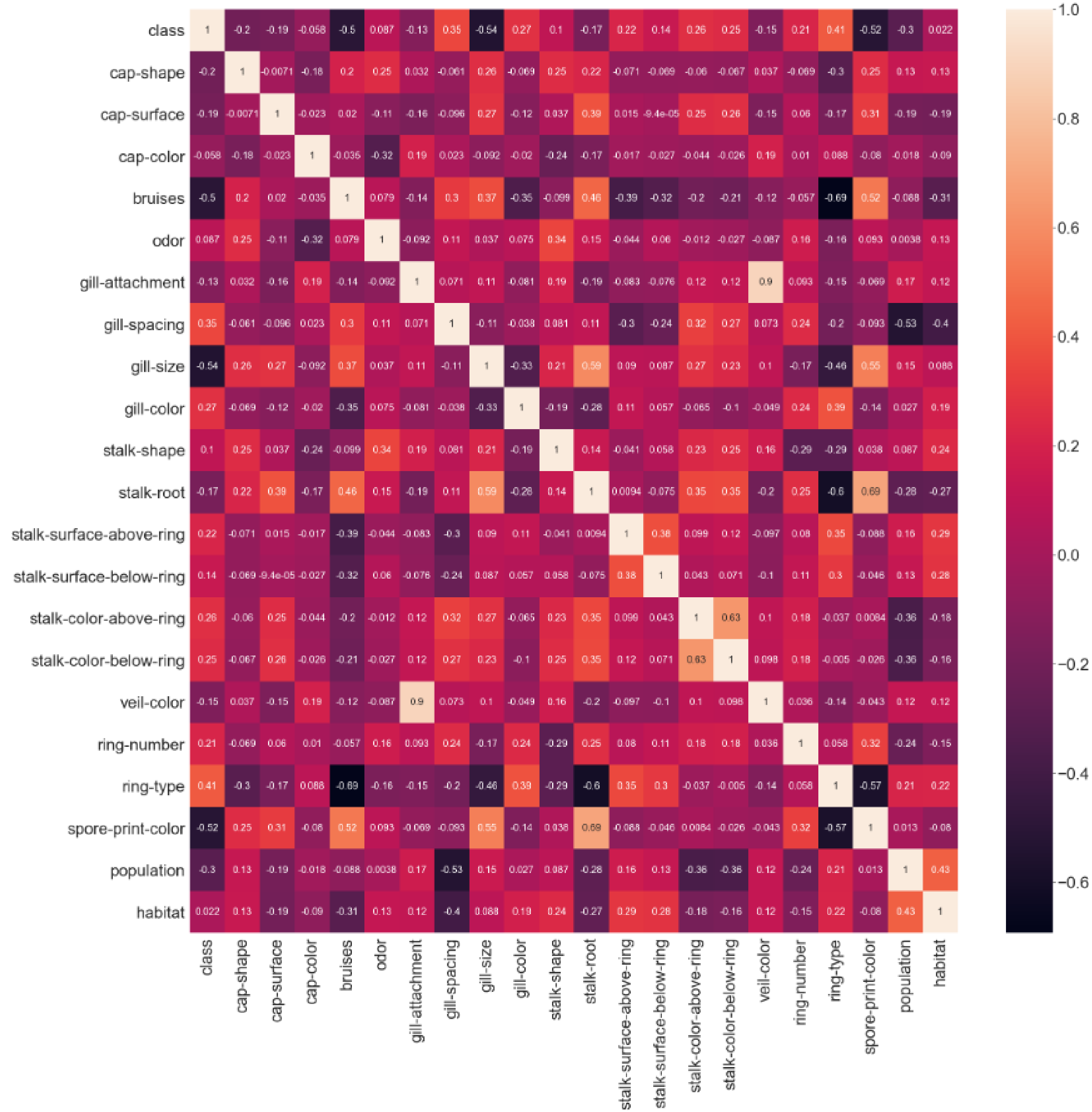
Rubén Simó Marín-1569391

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p	w	o	p	k	s	u
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p	w	o	p	n	n	g
2	e	b	s	w	t	l	f	c	b	n	...	s	w	w	p	w	o	p	n	n	m
3	p	x	y	w	t	p	f	c	n	n	...	s	w	w	p	w	o	p	k	s	u
4	e	x	s	g	f	n	f	w	b	k	...	s	w	w	p	w	o	e	n	a	g
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8119	e	k	s	n	f	n	a	c	b	y	...	s	o	o	p	o	o	p	b	c	l
8120	e	x	s	n	f	n	a	c	b	y	...	s	o	o	p	n	o	p	b	v	l
8121	e	f	s	n	f	n	a	c	b	n	...	s	o	o	p	o	o	p	b	c	l
8122	p	k	y	n	f	y	f	c	n	b	...	k	w	w	p	w	o	e	w	v	l
8123	e	x	s	n	f	n	a	c	b	y	...	s	o	o	p	o	o	p	o	c	l

```
Tipus d'atributs:
class                       object
cap-shape                   object
cap-surface                 object
cap-color                   object
bruises                     object
odor                        object
gill-attachment             object
gill-spacing                object
gill-size                   object
gill-color                  object
stalk-shape                 object
stalk-root                  object
stalk-surface-above-ring    object
stalk-surface-below-ring    object
stalk-color-above-ring      object
stalk-color-below-ring      object
veil-type                   object
veil-color                  object
ring-number                 object
ring-type                   object
spore-print-color           object
population                  object
habitat                     object
dtype: object
```

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
0	1	3	4	1	1	8	3	1	2	1	...	4	8	8	1	3	2	6	1	4	5
1	2	3	4	10	1	1	3	1	1	1	...	4	8	8	1	3	2	6	2	3	1
2	2	1	4	9	1	2	3	1	1	2	...	4	8	8	1	3	2	6	2	3	3
3	1	3	3	9	1	8	3	1	2	2	...	4	8	8	1	3	2	6	1	4	5
4	2	3	4	4	2	7	3	2	1	1	...	4	8	8	1	3	2	2	2	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8119	2	5	4	1	2	7	1	1	1	12	...	4	5	5	1	2	2	6	3	2	2
8120	2	3	4	1	2	7	1	1	1	12	...	4	5	5	1	1	2	6	3	5	2
8121	2	4	4	1	2	7	1	1	1	2	...	4	5	5	1	2	2	6	3	2	2
8122	1	5	3	1	2	4	3	1	2	3	...	3	8	8	1	3	2	2	8	5	2
8123	2	3	4	1	2	7	1	1	1	12	...	4	5	5	1	2	2	6	6	2	2

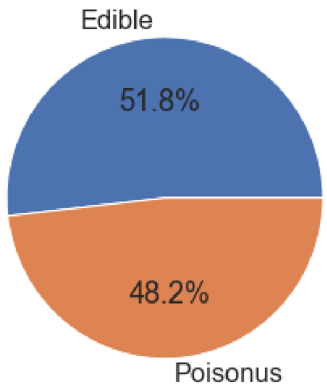
```
Tipus d'atributs:
class                       int64
cap-shape                   int64
cap-surface                 int64
cap-color                   int64
bruises                     int64
odor                        int64
gill-attachment             int64
gill-spacing                int64
gill-size                   int64
gill-color                  int64
stalk-shape                 int64
stalk-root                  int64
stalk-surface-above-ring    int64
stalk-surface-below-ring    int64
stalk-color-above-ring      int64
stalk-color-below-ring      int64
veil-type                   int64
veil-color                  int64
ring-number                 int64
ring-type                   int64
spore-print-color           int64
population                  int64
habitat                     int64
dtype: object
```



class
cap-shape
cap-surface
cap-color
bruises
odor
gill-attachment
gill-spacing
gill-size
gill-color
stalk-shape
stalk-root
stalk-surface-above-ring
stalk-surface-below-ring
stalk-color-above-ring
stalk-color-below-ring
veil-color
ring-number
ring-type
spore-print-color
population
habitat
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

```
2     4208
1     3916
Name: class, dtype: int64
```

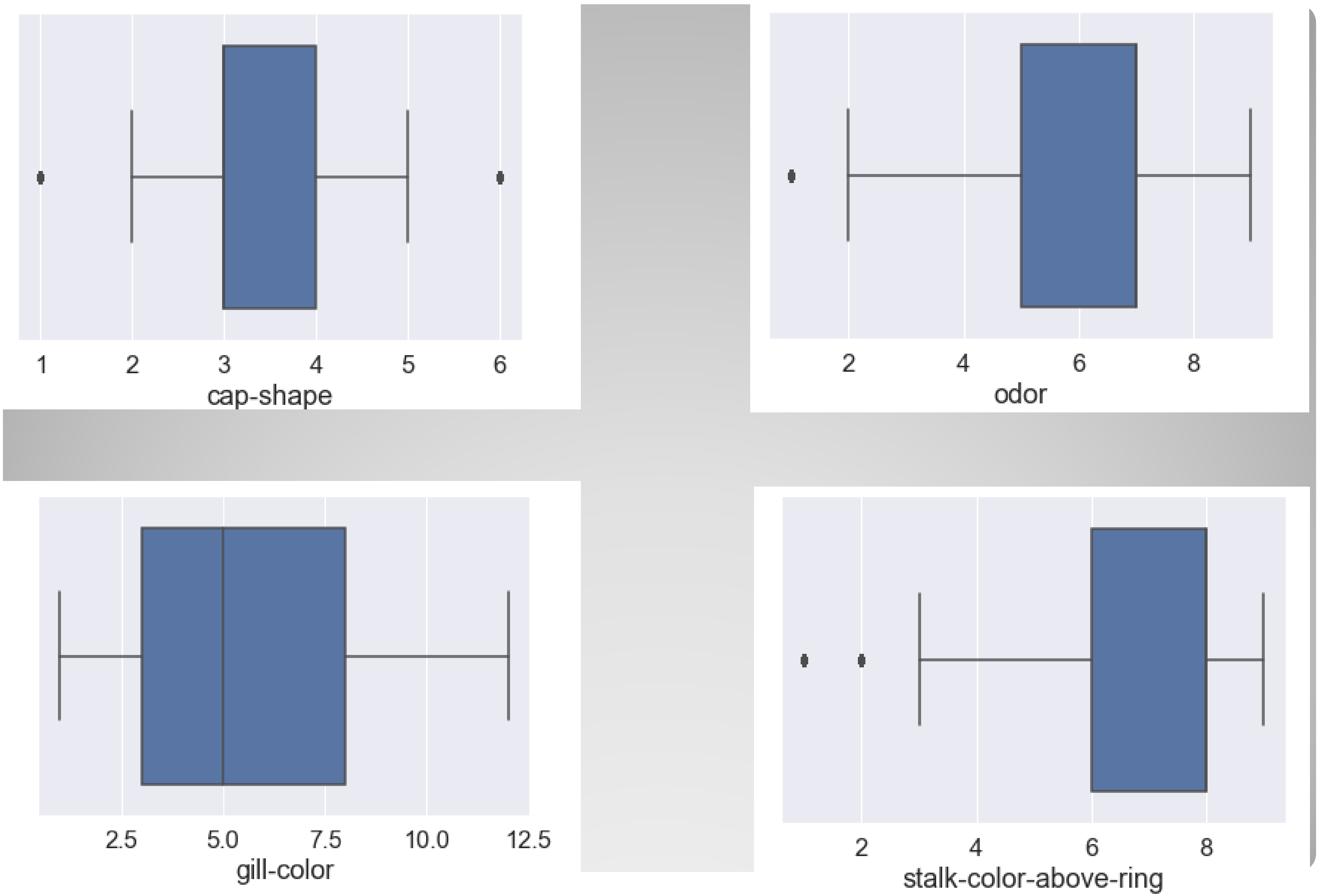
```
sns.set(font_scale = 1.5)
fig, ax = plt.subplots()
labels = 'Edible', 'Poisonus'
ax.pie(perc, labels = labels, autopct='%1.1f%%')
ax.axis('equal')

plt.show()
```



Prepocessing (normalitzation, outlier removal, feature selection..)

```
dataset.isnull().sum()
```

```
class                       0
cap-shape                   0
cap-surface                 0
cap-color                   0
bruises                     0
odor                        0
gill-attachment             0
gill-spacing                0
gill-size                   0
gill-color                  0
stalk-shape                 0
stalk-root                  0
stalk-surface-above-ring    0
stalk-surface-below-ring    0
stalk-color-above-ring      0
stalk-color-below-ring      0
veil-color                  0
ring-number                 0
ring-type                   0
spore-print-color           0
population                  0
habitat                     0
dtype: int64
```


1
2
3
4
5
6
cap-shape
2
4
6
8
odor
2.5
5.0
7.5
10.0
12.5
gill-color
2
4
6
8
stalk-color-above-ring

```
      class  cap-shape  cap-surface  cap-color   bruises      odor  \
0    -1.036613  -0.545782     1.065669  -1.255302 -1.185917  1.115027
1     0.964680  -0.545782     1.065669   1.357802 -1.185917 -2.413988
2     0.964680  -2.764967     1.065669   1.067457 -1.185917 -1.909843
3    -1.036613  -0.545782     0.217892   1.067457 -1.185917  1.115027
4     0.964680  -0.545782     1.065669  -0.384267  0.843230  0.610882
...        ...        ...          ...        ...       ...       ...
8119  0.964680   1.673403     1.065669  -1.255302  0.843230  0.610882
8120  0.964680  -0.545782     1.065669  -1.255302  0.843230  0.610882
8121  0.964680   0.563811     1.065669  -1.255302  0.843230  0.610882
8122 -1.036613   1.673403     0.217892  -1.255302  0.843230 -0.901553
8123  0.964680  -0.545782     1.065669  -1.255302  0.843230  0.610882

      gill-attachment  gill-spacing  gill-size  gill-color  ...  \
0            0.162896     -0.438864   1.494683   -1.415071  ...
1            0.162896     -0.438864  -0.669038   -1.415071  ...
2            0.162896     -0.438864  -0.669038   -1.115866  ...
3            0.162896     -0.438864   1.494683   -1.115866  ...
4            0.162896      2.278612  -0.669038   -1.415071  ...
...               ...           ...        ...         ...  ...
8119        -6.138869     -0.438864  -0.669038    1.876178  ...
8120        -6.138869     -0.438864  -0.669038    1.876178  ...
8121        -6.138869     -0.438864  -0.669038   -1.115866  ...
8122         0.162896     -0.438864   1.494683   -0.816662  ...
8123        -6.138869     -0.438864  -0.669038    1.876178  ...

      stalk-surface-above-ring  stalk-surface-below-ring  \
0                     0.615908                  0.660796
1                     0.615908                  0.660796
2                     0.615908                  0.660796
3                     0.615908                  0.660796
4                     0.615908                  0.660796
...                        ...                       ...
8119                  0.615908                  0.660796
8120                  0.615908                  0.660796
8121                  0.615908                  0.660796
8122                  0.615908                 -0.488242
8123                  0.615908                  0.660796
```

```
      stalk-surface-above-ring  stalk-surface-below-ring  \
0                     0.615908                  0.660796
1                     0.615908                  0.660796
2                     0.615908                  0.660796
3                     0.615908                  0.660796
4                     0.615908                  0.660796
...                        ...                       ...
8119                  0.615908                  0.660796
8120                  0.615908                  0.660796
8121                  0.615908                  0.660796
8122                  0.615908                 -0.488242
8123                  0.615908                  0.660796

      stalk-color-above-ring  stalk-color-below-ring  veil-color  ring-number  \
0                   0.724622                0.732112    0.142037    -0.256132
1                   0.724622                0.732112    0.142037    -0.256132
2                   0.724622                0.732112    0.142037    -0.256132
3                   0.724622                0.732112    0.142037    -0.256132
4                   0.724622                0.732112    0.142037    -0.256132
...                      ...                     ...         ...          ...
8119               -0.674783               -0.634961   -3.979055    -0.256132
8120               -0.674783               -0.634961   -8.100146    -0.256132
8121               -0.674783               -0.634961   -3.979055    -0.256132
8122                0.724622                0.732112    0.142037    -0.256132
8123               -0.674783               -0.634961   -3.979055    -0.256132

      ring-type  spore-print-color  population   habitat
0      0.948081          -1.083856   -0.514389  0.307811
1      0.948081          -0.729891   -1.313108 -1.272882
2      0.948081          -0.729891   -1.313108 -0.482535
3      0.948081          -1.083856   -0.514389  0.307811
4     -1.272216          -0.729891   -2.910546 -1.272882
...         ...                ...         ...       ...
8119   0.948081          -0.375925   -2.111827 -0.877709
8120   0.948081          -0.375925    0.284330 -0.877709
8121   0.948081          -0.375925   -2.111827 -0.877709
8122  -1.272216           1.393903    0.284330 -0.877709
8123   0.948081           0.685972   -2.111827 -0.877709

[8124 rows x 22 columns]
```

Cross-validation

- Cross-validation k-fold
- Cross-validation sufflesplit
- Leave-one-out

```python
# escollir la millor K per al nostre problema

folds = range(2,15)

def evaluatemodel(cv, standar):

    x = standar[:,[8]]
    y = standar[:,9]
    lab = preprocessing.LabelEncoder()
    y_transformed = lab.fit_transform(y)

    logReg = LogisticRegression()
    scores = cross_val_score(logReg, x, y_transformed, scoring='accuracy', cv=cv, n_jobs=-1)
    return mean(scores), scores.min(), scores.max()

for k in folds:
    cv = KFold(n_splits=k, shuffle=True, random_state=10)
    k_mean, k_min, k_max = evaluatemodel(cv, standardized)
    print('-> folds=%d, accuracy=%.3f (%.3f,%.3f)' % (k, k_mean, k_min, k_max))
```

```
-> folds=2, accuracy=0.374 (0.366,0.382)
-> folds=3, accuracy=0.374 (0.363,0.385)
-> folds=4, accuracy=0.374 (0.364,0.382)
-> folds=5, accuracy=0.374 (0.363,0.385)
-> folds=6, accuracy=0.374 (0.360,0.387)
-> folds=7, accuracy=0.374 (0.357,0.397)
-> folds=8, accuracy=0.374 (0.360,0.390)
-> folds=9, accuracy=0.374 (0.353,0.405)
-> folds=10, accuracy=0.374 (0.347,0.400)
-> folds=11, accuracy=0.374 (0.352,0.410)
-> folds=12, accuracy=0.374 (0.340,0.399)
-> folds=13, accuracy=0.374 (0.342,0.403)
-> folds=14, accuracy=0.374 (0.349,0.407)
```

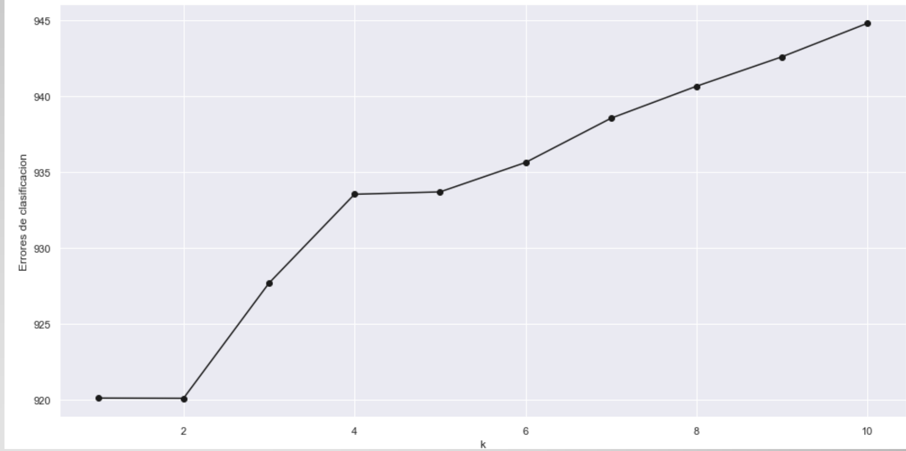
Metric Analysis

- **Accuracy_score**
- **F1_score**
- **Average_precision_score**

```python
# PR Curve and Roc Curve functions

from sklearn.metrics import f1_score, precision_recall_curve, average_precision_score, roc_curve, auc

def PRCurve(y_v,probs, n_classes):
    precision = {}
    recall = {}
    average_precision = {}
    plt.figure()
    for i in range(n_classes):
        precision[i], recall[i], _ = precision_recall_curve(y_v == i, probs[:, i])
        average_precision[i] = average_precision_score(y_v == i, probs[:, i])

        plt.plot(recall[i], precision[i],
        label='Precision-recall curve of class {0} (area = {1:0.2f})'
                               ''.format(i, average_precision[i]))
        plt.xlabel('Recall')
        plt.ylabel('Precision')
        plt.legend(loc="upper right")

def RocCurve(y_v, probs, n_classes):
    fpr = {}
    tpr = {}
    roc_auc = {}
    for i in range(n_classes):
        fpr[i], tpr[i], _ = roc_curve(y_v == i, probs[:, i])
        roc_auc[i] = auc(fpr[i], tpr[i])

    # Compute micro-average ROC curve and ROC area
    # Plot ROC curve
    plt.figure()
    for i in range(n_classes):
        plt.plot(fpr[i], tpr[i], label='ROC curve of class {0} (area = {1:0.2f})' ''.format(i, roc_auc[i]))
    plt.legend()
```

```
LOGISTIC REGRESION

Logistic Regression Score:  0.4073846153846154
Logistic Regression Cross Val Score:  0.42791292993776225

963 Errores de clasificacion de un total de 1625
662 Aciertos de clasificacion de un total de 1625

Confusion matrix:
[[  1  15   0   0   1   0   0  40   0   0  25   0]
 [  8  19   0   0   4   0   4  51   0   0 110   7]
 [  0   0 325   0   0   0   0   0   0   0   0   0]
 [  4   7   0  56  25   0   0  46   0   0   0   0]
 [  8  16   2  59  33   4   0  23   0   0  12   0]
 [  0   0   0   0   0   0   0   3   0   0   0   0]
 [  0   0   0   0   0   0   1   0   0   0   0  11]
 [ 10  24   4  59  40   0   0  84   0   0 106   0]
 [  3   3   1   0   2   0   0   9   0   0  83   0]
 [  0   0   0   0   0   0   0   0   0   8  14   0]
 [  6  31   2   1   4   4   0  59   0   6 123   0]
 [  1   0   0   4   0   0   1   1   0   0   0  12]]

Logistic Regression F1 Score: 0.4073846153846154
```

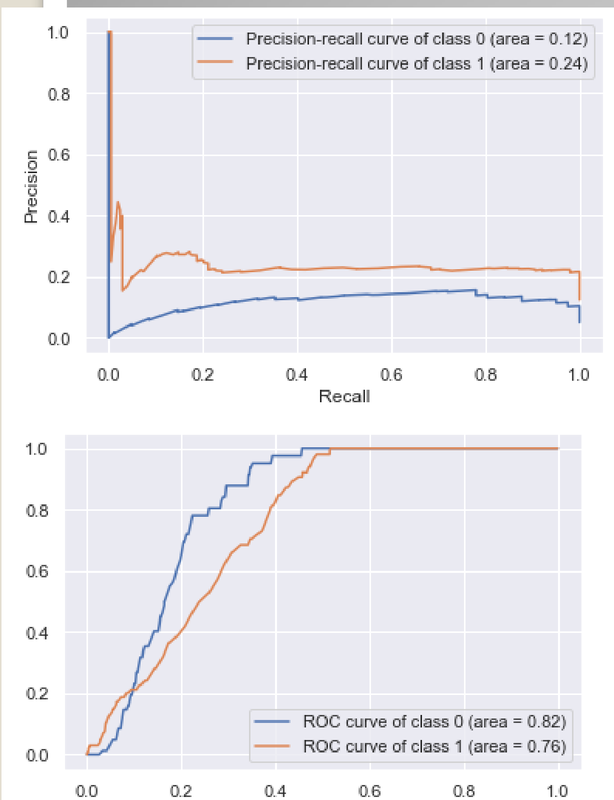
```
NEAREST K NEIGHBORS

k=1: 920.109 errores de clasificación de un total de 1625
k=2: 920.096 errores de clasificación de un total de 1625
k=3: 927.693 errores de clasificación de un total de 1625
k=4: 933.55 errores de clasificación de un total de 1625
k=5: 933.705 errores de clasificación de un total de 1625
k=6: 935.644 errores de clasificación de un total de 1625
k=7: 938.564 errores de clasificación de un total de 1625
k=8: 940.675 errores de clasificación de un total de 1625
k=9: 942.616 errores de clasificación de un total de 1625
k=10: 944.828 errores de clasificación de un total de 1625
```

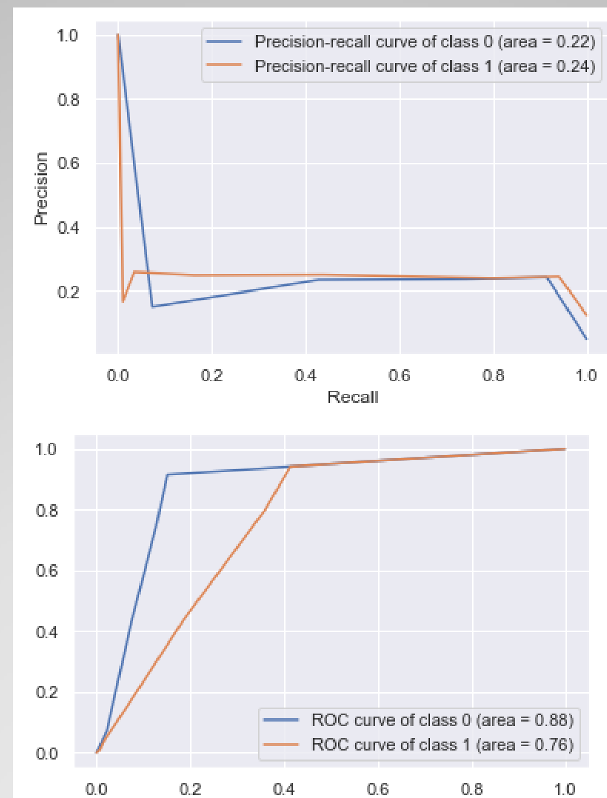


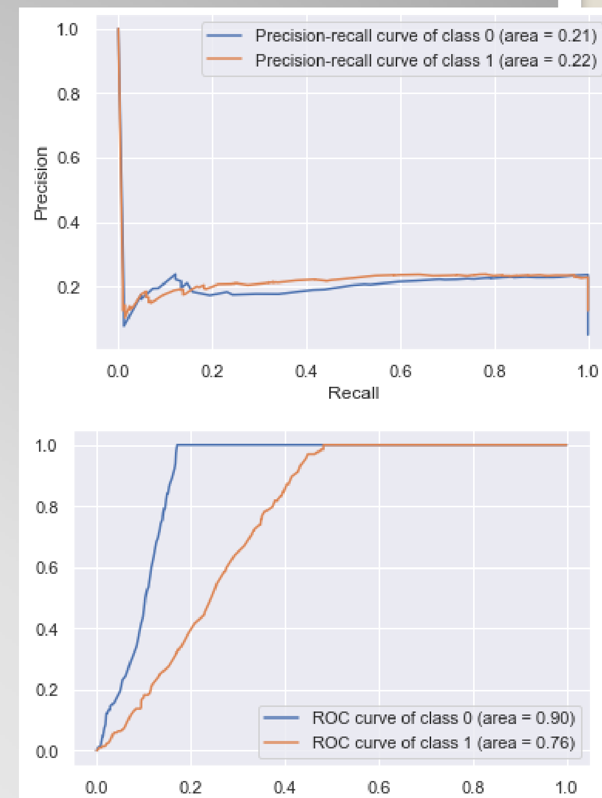

```
Nearest K Neighbour Score:  0.4006153846153846
Nearest K Neighbour Cross Val Score:  0.4136011889924456

974 Errores de clasificacion de un total de 1625
651 Aciertos de clasificacion de un total de 1625

Confusion matrix:
[[ 32  18   0  10   9   0   0   9   0   0   4   0]
 [ 28  51   0  19  12   0   2  44  19   0  24   4]
 [  0   0 325   0   0   0   0   0   0   0   0   0]
 [ 14   4   0  40  46   0   0  22   3   0   9   0]
 [ 17  11   0  52  46   4   0  16   0   0  11   0]
 [  0   0   0   0   1   0   0   0   0   0   2   0]
 [  0   3   0   0   0   0   4   0   0   0   0   5]
 [ 29  50   0  60  48   0   0  75  22   0  43   0]
 [  0  28   0   0   2   0   0  42  14   0  15   0]
 [  0   0   0   0   0   0   0   1   1   7  13   0]
 [ 13  35   2   3  21   4   0  71  17  16  54   0]
 [  0   4   0   2   1   0   2   1   0   0   6   3]]

Nearest K Neighbour F1 Score: 0.4006153846153846
```

```
SVM with rbf kernel Score:  0.4024615384615385
SVM with rbf kernel Cross Val Score:  0.4314517704002638

971 Errores de clasificacion de un total de 1625
654 Aciertos de clasificacion de un total de 1625

Confusion matrix:
[[  0  10   0   0   5   0   0  38   0   0  29   0]
 [  0   6   0   0   6   0   0  47   6   0 127  11]
 [  0   0 325   0   0   0   0   0   0   0   0   0]
 [  0   0   0  49  32   0   0  55   0   0   2   0]
 [  0  18   2  71  20   4   0  23   5   0  14   0]
 [  0   0   0   0   0   0   0   0   0   0   3   0]
 [  0   0   0   0   0   0   0   0   0   0   0  12]
 [  0   3   4  67  35   0   0  82   3   0 133   0]
 [  0   2   1   0   2   0   0   3   1   0  92   0]
 [  0   0   0   0   0   0   0   0   0   0  22   0]
 [  0  13   2   1   0   4   0  54   0   0 162   0]
 [  0   0   0   4   0   0   0   0   0   0   6   9]]

SVM with rbf kernel F1 Score: 0.4024615384615385
```

```
SVM Linear Score:  0.4086153846153846
SVM Linear Cross Val Score:  0.4312980189282621

961 Errores de clasificacion de un total de 1625
664 Aciertos de clasificacion de un total de 1625

Confusion matrix:
[[  0  14   0   0   0   0   0  36   0   0  32   0]
 [  1  20   4   0   0   0   4  48   0   0 119   7]
 [  0   0 325   0   0   0   0   0   0   0   0   0]
 [  0   7   0  63  18   0   0  50   0   0   0   0]
 [  7  12   2  67  23   4   0  23   0   0  19   0]
 [  0   0   0   0   0   0   0   3   0   0   0   0]
 [  0   0   0   0   0   0   1   0   0   0   0  11]
 [  8  29   5  71  27   0   0  78   0   0 109   0]
 [  3   3   3   0   0   0   0   0   0   0  92   0]
 [  0   0   0   0   0   0   0   0   0  10  12   0]
 [  5  27   2   1   4   4   0  49   0  12 132   0]
 [  0   0   0   4   0   0   1   2   0   0   0  12]]

SVM Linear F1 Score: 0.4086153846153846
```


Precision-recall curve of class 0 (area = 0.12)
Precision-recall curve of class 1 (area = 0.24)
Precision
Recall
ROC curve of class 0 (area = 0.82)
ROC curve of class 1 (area = 0.76)


Precision-recall curve of class 0 (area = 0.22)
Precision-recall curve of class 1 (area = 0.24)
Precision
Recall
ROC curve of class 0 (area = 0.88)
ROC curve of class 1 (area = 0.76)


Precision-recall curve of class 0 (area = 0.21)
Precision-recall curve of class 1 (area = 0.22)
Precision
Recall
ROC curve of class 0 (area = 0.90)
ROC curve of class 1 (area = 0.76)

Logistic Regression

	precision	recall	f1-score	support
class 0	0.02	0.01	0.02	82
class 1	0.17	0.09	0.12	203
class 2	0.97	1.00	0.99	325
class 3	0.31	0.41	0.35	138
class 4	0.30	0.21	0.25	157
class 5	0.00	0.00	0.00	3
class 6	0.17	0.08	0.11	12
class 7	0.27	0.26	0.26	327
class 8	0.00	0.00	0.00	101
class 9	0.57	0.36	0.44	22
class 10	0.26	0.52	0.35	236
class 11	0.40	0.63	0.49	19
accuracy			0.41	1625
macro avg	0.29	0.30	0.28	1625
weighted avg	0.38	0.41	0.38	1625

Nearest K Neighbour

	precision	recall	f1-score	support
class 0	0.24	0.39	0.30	82
class 1	0.25	0.25	0.25	203
class 2	0.99	1.00	1.00	325
class 3	0.22	0.29	0.25	138
class 4	0.25	0.29	0.27	157
class 5	0.00	0.00	0.00	3
class 6	0.50	0.33	0.40	12
class 7	0.27	0.23	0.25	327
class 8	0.18	0.14	0.16	101
class 9	0.30	0.32	0.31	22
class 10	0.30	0.23	0.26	236
class 11	0.25	0.16	0.19	19
accuracy			0.40	1625
macro avg	0.31	0.30	0.30	1625
weighted avg	0.40	0.40	0.40	1625

SVM

	precision	recall	f1-score	support
class 0	0.00	0.00	0.00	82
class 1	0.12	0.03	0.05	203
class 2	0.97	1.00	0.99	325
class 3	0.26	0.36	0.30	138
class 4	0.20	0.13	0.16	157
class 5	0.00	0.00	0.00	3
class 6	0.00	0.00	0.00	12
class 7	0.27	0.25	0.26	327
class 8	0.07	0.01	0.02	101
class 9	0.00	0.00	0.00	22
class 10	0.27	0.69	0.39	236
class 11	0.28	0.47	0.35	19
accuracy			0.40	1625
macro avg	0.20	0.24	0.21	1625
weighted avg	0.35	0.40	0.36	1625

Hyperparameter Search

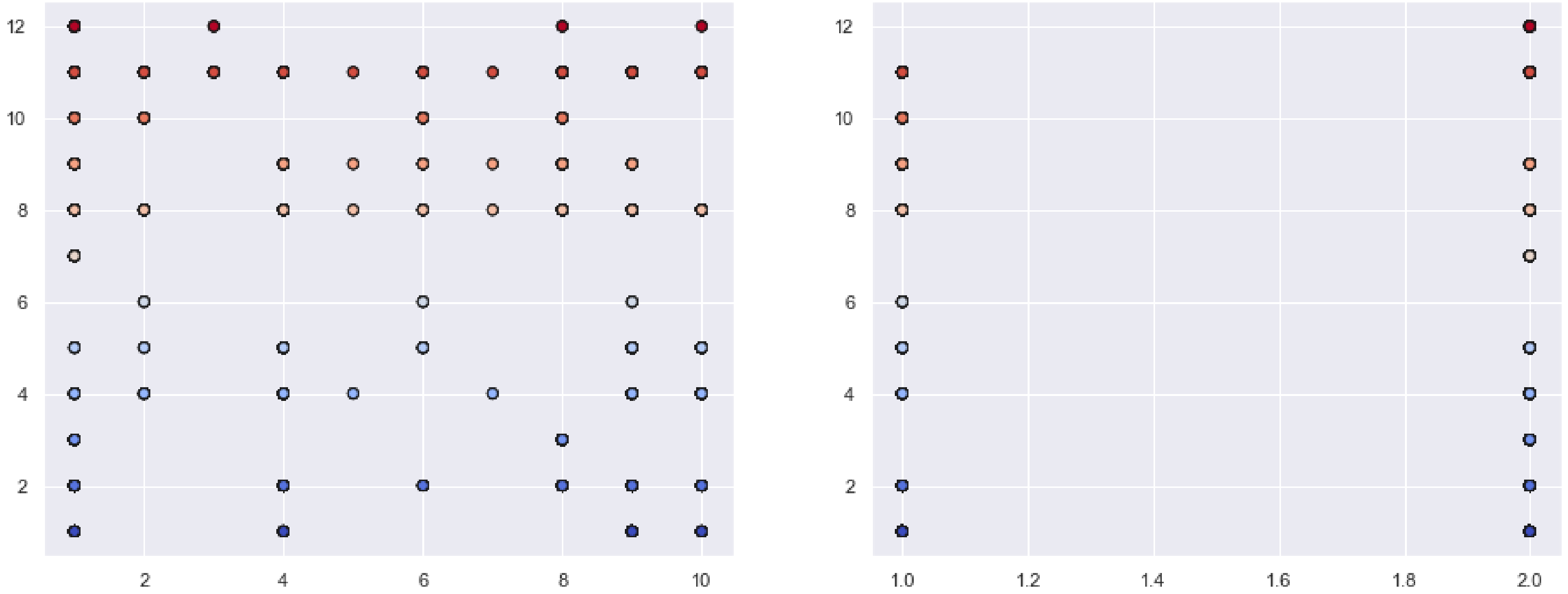
- **Exhaustive Grid Search**
- **Randomized Parameter Optimization**
- **Bayesian optimization search**

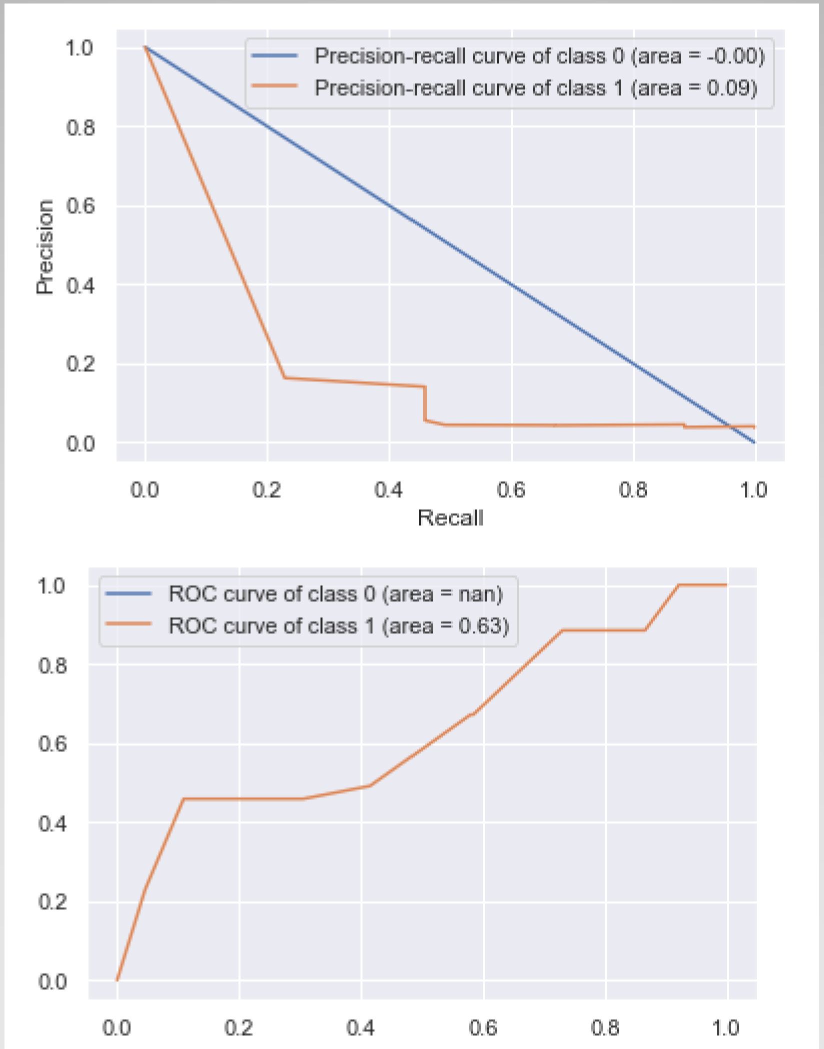
```python
# GridSearchCV
from sklearn.model_selection import GridSearchCV

parameters = {'kernel':('linear', 'rbf'), 'C':[0,0.2,0.5,0.75,1,1.25,1.5], 'gamma':[0.09,0.08,0.07,0.1]}
svc = svm.SVC()
clf = GridSearchCV(svc, parameters)
search = clf.fit(x_train,y_train)
print("GridSearchCV: ", search.best_params_)

# RandomizedSearchCV
from sklearn.model_selection import RandomizedSearchCV

clf = RandomizedSearchCV(svc, parameters, random_state=0)
search = clf.fit(x_train,y_train)
print("RandomizedSearchCV: ", search.best_params_)
```

```
GridSearchCV:  {'C': 0.5, 'gamma': 0.07, 'kernel': 'rbf'}
RandomizedSearchCV:  {'kernel': 'linear', 'gamma': 0.09, 'C': 1.25}
```

Comparativa de models

```
Correct classification Logistic  0.5 % of the data:  0.3380108321024126
Correct classification SVM       0.5 % of the data:  0.3380108321024126
Correct classification Logistic  0.7 % of the data:  0.33305988515176377
Correct classification SVM       0.7 % of the data:  0.33305988515176377
Correct classification Logistic  0.8 % of the data:  0.3686153846153846
Correct classification SVM       0.8 % of the data:  0.3464615384615385
```




Precision-recall curve of class 0 (area = -0.00)
Precision-recall curve of class 1 (area = 0.09)
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve of class 0 (area = nan)
ROC curve of class 1 (area = 0.63)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0