UNIVERSIDAD DE GRANADA

MÁSTER DE CIENCIA DE DATOS E INGENIERÍA DE COMPUTADORES

CURSO ACADÉMICO 2019-2020

EXTRACCIÓN DE CARACTERÍSTICAS EN IMÁGENES

Procesos gaussianos para clasificación

Desarrollo y análisis de modelos de basados en procesos gaussianos para clasificación de imágenes de tejido cancerígeno.

Nicolás Cubero Torres

10 de Marzo de 2020

Índice

Índice de figuras

Índice de tablas

1. ¿Qué es un Proceso Gaussiano?

Un **Proceso Gaussiano** es una colección de variables aleatorias de forma que cualquier conjunto finito de las mismas presenta una distribución Normal conjunta.

Un Proceso Gaussiano queda completamente especificado por una **función de media** y una **función de covarianza** (*kernel*) que permiten definir, de manera respectiva, la media y matriz de covarianza de la distribución conjunta de cualquiera de los conjuntos finitos de variables.

Para el caso de problemas de clasificación basados en modelos bayesianos, se combinan estos procesos gausianos con las funciones latentes de los modelos logísticos dando lugar a un nuevo tipo de modelo de clasificación que toma las propiedades de los procesos gausianos.

2. Software utilizado para la realización de la práctica

La ejecución de los experimentos por medio de *scripts* implementados en Matlab [1] haciendo uso de la librería *GPML* [2] para el desarrollo de modelos de clasificación basados en Procesos Gausianos.

Por su parte, se usaron los núcleos lineal y de exponenciación cuadrática isotrópica (Squared exponential isotropic). Para el segundo núcleo, se inicializaron los hiperparámetros varianza y longitud-escala a los valores 1 y 1.9 respectivamente.

Como modelo de probabilidad, se usó en todos los casos el modelo asociado a regresión logística. Por su parte, y dado que el *dataset* un desbalanceamiento entre imágenes entre clases, para cada *fold*, se particionaran los patrones de entrenamiento de la clase mayoritaria en una serie de conjuntos disjuntos cuyo tamaño sea aproximadamente similar al conjunto de patrones de entrenamiento de la clase minoritaria y se elaborarán un *bagging* de clasificadores GP, enfrentando los patrones de la clase minoritaria con cada conjunto.
La predicción dada para un patrón del conjunto de *test*, resultara en la predicción media del *bagging* de clasificadores.

[1] https://www.mathworks.com/products/matlab.html
[2] http://www.gaussianprocess.org/gpml/code/matlab/doc/index.html

Por último, en todos los ajustes se ejecutaron un total de 40 evaluaciones en la optimización de los hiperparámetros.

3. Resultados experimentales

A continuación, se procede a exponer y analizar los resultados de los experimentos realizados en cada una de las 5 particiones *fold* en las que se han dividido los datos.
Primeramente, se mostrarán los resultados obtenidos con el *kernel* lineal seguido de los resultados obtenidos con el *kernel* Gaussiano (exponenciación cuadrática) llevando a cabo una comparación entre ambas.

3.1. Kernel Gaussiano

Con el kernel Gaussiano, los resultados revelaron las siguientes métricas de rendimiento para cada *fold*:

1. **Fold nº 1**:

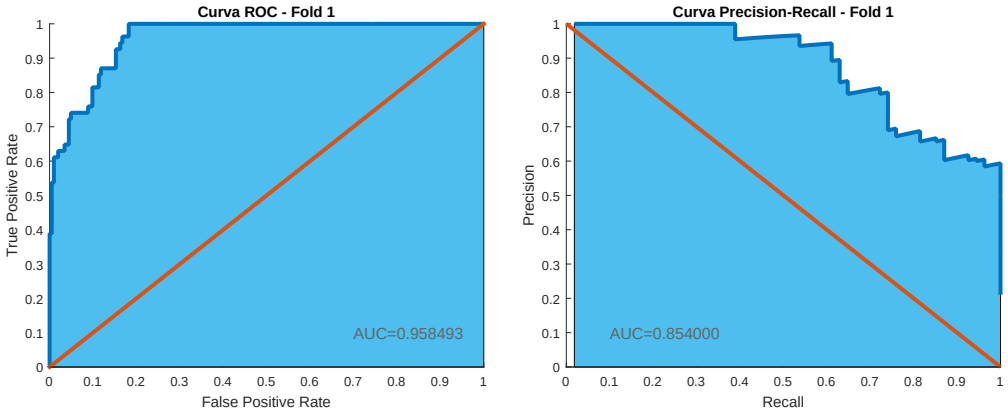

Figura 1: A la izquierda, curva ROC y área bajo la curva del modelo generado para el Fold 1 y a la derecha, su curva Precision-Recall y área bajo la curva.

Considerando como clase positiva todas aquellas instancias con una probabilidad de pertenencia a la clase positiva (cancerígena) superior a 0.5, se calcularon las siguientes métricas de rendimiento:

Accuracy	0.6537
Specificity	0.5616
Sensitivity	1
Precision	0.3776
F-score	0.5482

Tabla 1: Métricas de accuracy, precission, sensitivity y F-score del modelo generado para el fold 1.

Por último, la matriz de confusión asociada a esta clasificación es la siguiente:

		Clase Real	
		+	-
Clase Predicha	+	54	0
	-	89	114

Tabla 2: Matriz de confusión del modelo con kernel Gaussiano (SE) generado para el fold 1

2. **Fold nº 2**:

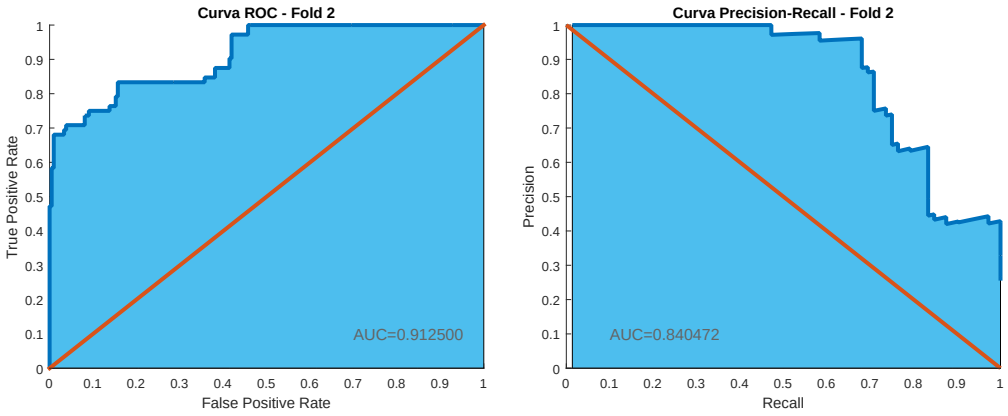

Figura 2: A la izquierda, curva ROC y área bajo la curva del modelo generado para el Fold 2 y a la derecha, su curva Precision-Recall y área bajo la curva.

De la misma forma que el anterior *fold*, las métricas evaluadas para los modelos generados y evaluados sobre este *fold* son:

Accuracy	0.7199
Specificity	0.6810
Sensitivity	0.8333
Precision	0.4724
F-score	0.6030

Tabla 3: Métricas de accuracy, precission, sensitivity y F-score del modelo generado para el fold 2.

Por último, la matriz de confusión asociada a esta clasificación es la siguiente:

		Clase Real	
		+	-
Clase Predicha	+	60	12
	-	67	143

Tabla 4: Matriz de confusión del modelo con kernel Gaussiano (SE) generado para el fold 2

3. **Fold nº 3**:

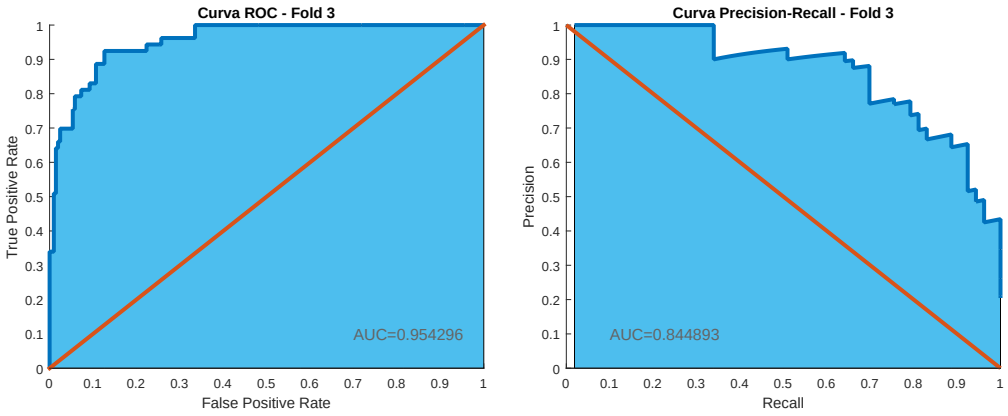

Figura 3: A la izquierda, curva ROC y área bajo la curva del modelo generado para el Fold 3 y a la derecha, su curva Precision-Recall y área bajo la curva.

De la misma forma que el anterior *fold*, las métricas evaluadas para los modelos generados y evaluados sobre este *fold* son:

Accuracy	0.7838
Specificity	0.7427
Sensitivity	0.9434
Precision	0.4854
F-score	0.6410

Tabla 5: Métricas de accuracy, precission, sensitivity y F-score del modelo generado para el fold 3.

Por último, la matriz de confusión asociada a esta clasificación es la siguiente:

		Clase Real	
		+	-
Clase Predicha	+	50	3
	-	53	153

Tabla 6: Matriz de confusión del modelo con kernel Gaussiano (SE) generado para el fold 3

4. **Fold nº 4**:

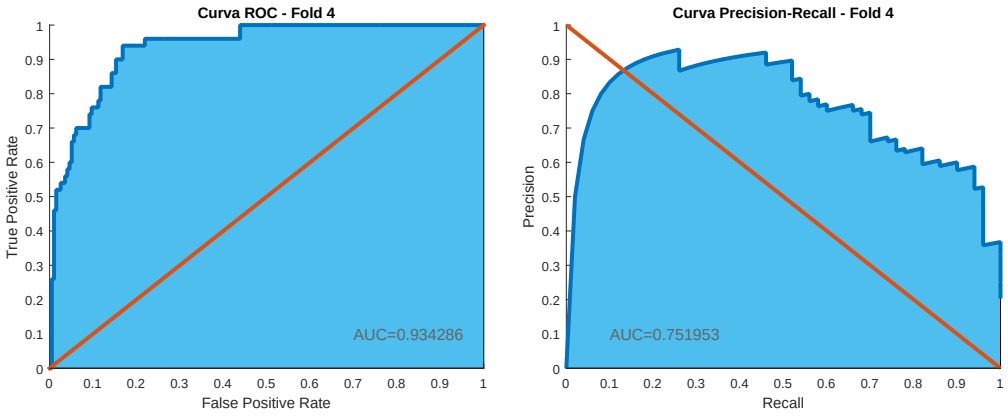

Figura 4: A la izquierda, curva ROC y área bajo la curva del modelo generado para el Fold 4 y a la derecha, su curva Precision-Recall y área bajo la curva.

De la misma forma que el anterior *fold*, las métricas evaluadas para los modelos generados y evaluados sobre este *fold* son:

Accuracy	0.7276
Specificity	0.6684
Sensitivity	0.9600
Precision	0.4248
F-score	0.5890

Tabla 7: Métricas de accuracy, precission, sensitivity y F-score del modelo generado para el fold 4.

Por último, la matriz de confusión asociada a esta clasificación es la siguiente:

		Clase Real	
		+	-
Clase Predicha	+	48	2
	-	65	131

Tabla 8: Matriz de confusión del modelo con kernel Gaussiano (SE) generado para el fold 4.

5. **Fold nº 5**:

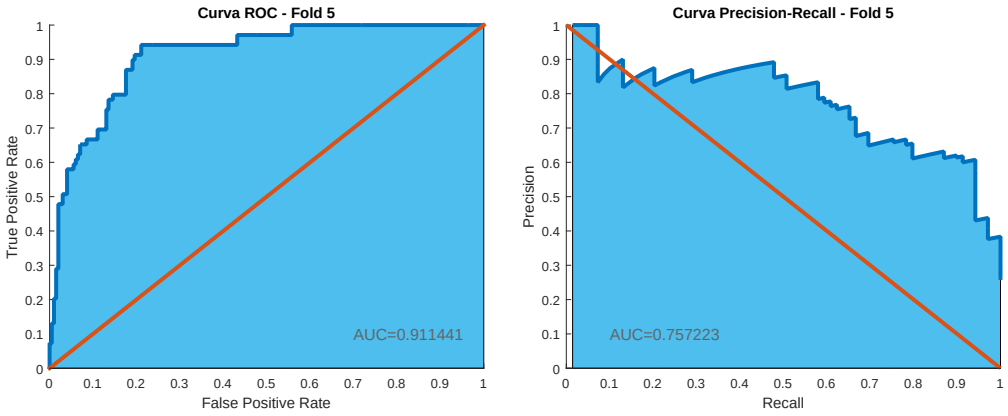

Figura 5: A la izquierda, curva ROC y área bajo la curva del modelo generado para el Fold 5 y a la derecha, su curva Precision-Recall y área bajo la curva.

De la misma forma que el anterior *fold*, las métricas evaluadas para los modelos generados y evaluados sobre este *fold* son:

Accuracy	0.7463
Specificity	0.6784
Sensitivity	0.9420
Precision	0.5039
F-score	0.6566

Tabla 9: Métricas de accuracy, precission, sensitivity y F-score del modelo generado para el fold 5.

Por último, la matriz de confusión asociada a esta clasificación es la siguiente:

		Clase Real	
		+	-
Clase Predicha	+	65	4
	-	65	139

Tabla 10: Matriz de confusión del modelo con kernel Gaussiano (SE) generado para el fold 5

El *accuracy* medio obtenido en todo el proceso es de 0.726250. En general, se puede observar que los modelos generados **presenta una mayor tendencia en clasificar patrones hacia la clase positiva**, lo que resulta en un número alto de falsos positivos.

Si bien el uso de *baggings* de clasificadores GP, ha evitado que los clasificadores resulten sesgados hacia la revela negativa, el uso repetido de los patrones de la clase minoritaria en el *bagging* de clasificadores, provoca la aparición de cierto sesgo a la clase positiva.

3.2. Kernel Lineal

por su parte, se repiten los anteriores experimentos usando un *kernel* lineal que aportó los siguientes resultados en cada *fold*:

1. **Fold nº 1**:

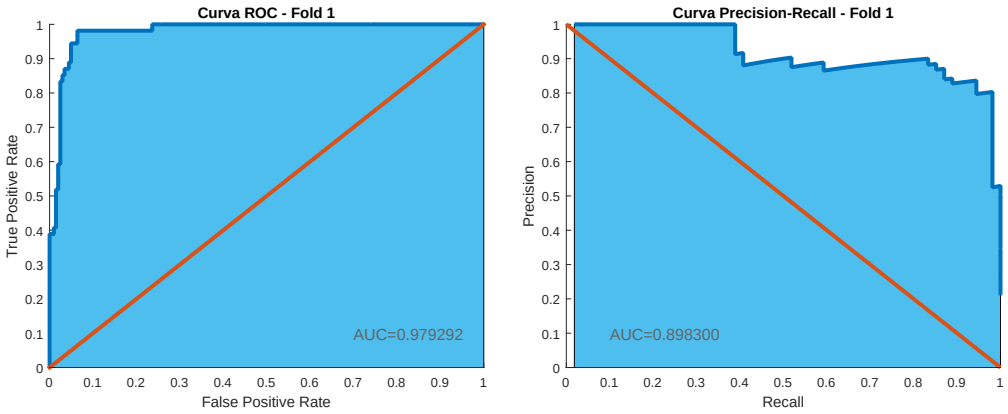

Figura 6: A la izquierda, curva ROC y área bajo la curva del modelo generado para el Fold 1 y a la derecha, su curva Precision-Recall y área bajo la curva.

Considerando como revela positiva todas aquellas instancias con una probabilidad de pertenencia a la revela positiva (cancerígena) superior a 0.5, se calcularon las siguientes métricas de rendimiento:

Accuracy	0.6965
Specificity	0.6158
Sensitivity	1
Precision	0.4091
F-score	0.5806

Tabla 11: Métricas de accuracy, precission, sensitivity y F-score del modelo generado para el fold 1.

Por último, la matriz de confusión asociada a esta clasificación es la siguiente:

		revela Real	
		+	-
revela Predicha	+	54	0
	-	78	125

Tabla 12: Matriz de confusión del modelo con kernel Gaussiano (SE) generado para el fold 1

2. **Fold nº 2**:

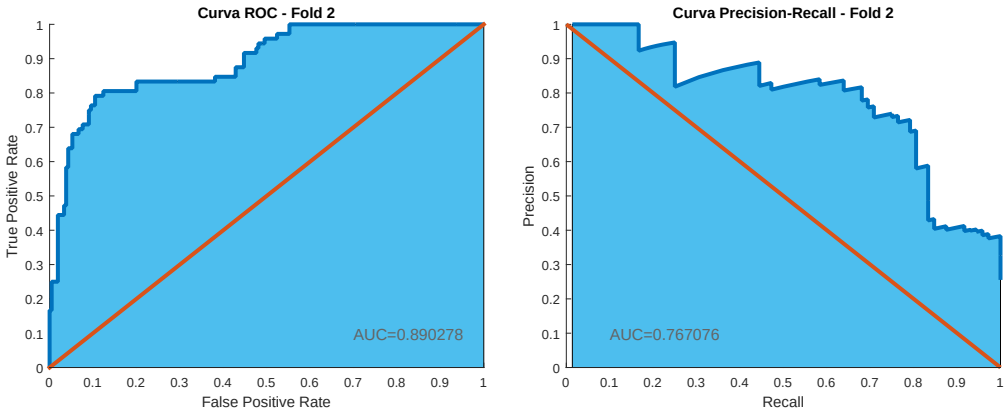

Figura 7: A la izquierda, curva ROC y área bajo la curva del modelo generado para el Fold 2 y a la derecha, su curva Precision-Recall y área bajo la curva.

De la misma forma que el anterior *fold*, las métricas evaluadas para los modelos generados y evaluados sobre este *fold* son:

Accuracy	0.6348
Specificity	0.5524
Sensitivity	0.8750
Precision	0.4013
F-score	0.5502

Tabla 13: Métricas de accuracy, precision, sensitivity y F-score del modelo generado para el fold 2.

Por último, la matriz de confusión asociada a esta clasificación es la siguiente:

		revela Real	
		+	-
revela Predicha	+	63	9
	-	94	116

Tabla 14: Matriz de confusión del modelo con kernel Lineal generado para el fold 2

3. **Fold nº 3**:

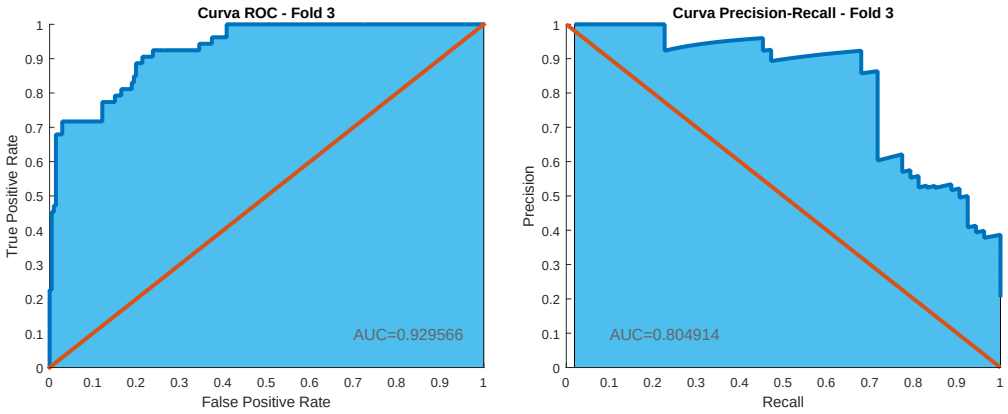

Figura 8: A la izquierda, curva ROC y área bajo la curva del modelo generado para el Fold 3 y a la derecha, su curva Precision-Recall y área bajo la curva.

De la misma forma que el anterior *fold*, las métricas evaluadas para los modelos generados y evaluados sobre este *fold* son:

Accuracy	0.6911
Specificity	0.6262
Sensitivity	0.9434
Precision	0.3937
F-score	0.5556

Tabla 15: Métricas de accuracy, precission, sensitivity y F-score del modelo generado para el fold 3.

Por último, la matriz de confusión asociada a esta clasificación es la siguiente:

		revela Real	
		+	-
revela Predicha	+	50	3
	-	77	129

Tabla 16: Matriz de confusión del modelo con kernel Gaussiano (SE) generado para el fold 3

4. **Fold nº 4**:

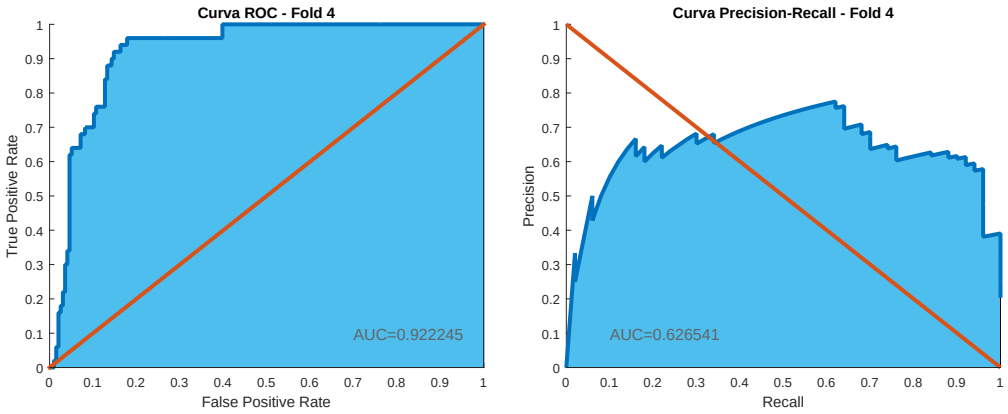

Figura 9: A la izquierda, curva ROC y área bajo la curva del modelo generado para el Fold 4 y a la derecha, su curva Precision-Recall y área bajo la curva.

De la misma forma que el anterior *fold*, las métricas evaluadas para los modelos generados y evaluados sobre este *fold* son:

Accuracy	0.6789
Specificity	0.6071
Sensitivity	0.9600
Precision	0.3840
F-score	0.5486

Tabla 17: Métricas de accuracy, precission, sensitivity y F-score del modelo generado para el fold 4.

Por último, la matriz de confusión asociada a esta clasificación es la siguiente:

		revela Real	
		+	-
revela Predicha	+	48	2
	-	77	119

Tabla 18: Matriz de confusión del modelo con kernel Lineal generado para el fold 4

5. **Fold nº 5**:

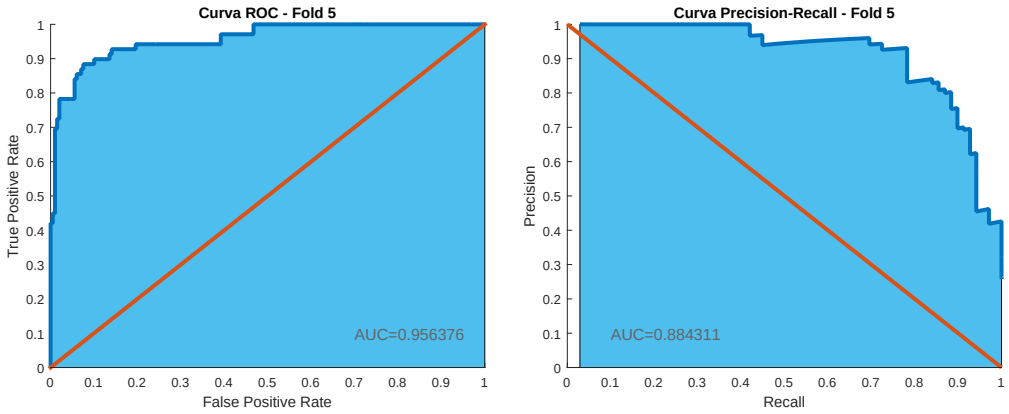

Figura 10: A la izquierda, curva ROC y área bajo la curva del modelo generado para el Fold 5 y a la derecha, su curva Precision-Recall y área bajo la curva.

De la misma forma que el anterior *fold*, las métricas evaluadas para los modelos generados y evaluados sobre este *fold* son:

Accuracy	0.7275
Specificity	0.6533
Sensitivity	0.9420
Precision	0.4851
F-score	0.6404

Tabla 19: Métricas de accuracy, precission, sensitivity y F-score del modelo generado para el fold 5.

Por último, la matriz de confusión asociada a esta clasificación es la siguiente:

		revela Real	
		+	-
revela Predicha	+	64	4
	-	69	130

Tabla 20: Matriz de confusión del modelo con kernel Gaussiano (SE) generado para el fold 5

El *accuracy* medio de los modelos generados se sitúa en 0.685769. Se aprecia que **el rendimiento de los modelos generados con este**

***kernel* es reducido en comparación con el *kernel* gausiano.**

Por su parte, en estos modelos se observa el mismo fenómeno analizado para el *kernel* Gaussiano: Los modelos presentan cierta tendencia a clasificar patrones en la revela positiva, volviendo a generar un número alto de falsos positivos.

3.3. Procedimiento para la clasificación de un nuevo patrón

Para la clasificación de un nuevo patrón una vez obtenidos todos los modelos con todos los hiperparámetros correspondientes, se procedería a tomar los mismos, junto con las funciones de media, varianza y el modelo probabilístico designado, se procedería a organizar el modelo para determinar la probabilidad de pertenencia a la revela positiva de cada instancia de test, fijando un umbral de probabilidad mínima.

Para la librería GPML de Matlab, una vez optimizados los hiperparámetros mediante *minimize*, se procedería a usar la función *gp*.

Siendo X_train e y_train el conjunto de datos de entrenamiento y las etiquetas de cada instancia respectivamente e X_test el conjunto de datos de test que se desea clasificar:

```
[a, b, c, d, lp] = gp(hyperparameters, <método inferencia>,
                      <función de media>, <función de covarianza>,
                      <modelo probabilístico>, X_train, Y_train,
                      X_test, ones(size(X_test,1),1));
```

Sabiendo que *ones(size(X_test,1),1)*, permite especificar a la función que debe de calcular la probabilidad de pertenencia de cada patrón a la revela positiva (representada por 1 en este problema), la anterior función, calcularía el logaritmo de la probabilidad (*lp*) de pertenencia a la revela positiva de cada instancia, por lo que las probabilidades reales se calcularían tomando $prob = e^{lp}$.

```
prob = exp(lp);
```

Por último, habiendo fijado un umbral mínimo de probabilidad θ, se procedería a clasificar cada instancia en la revela positiva (+1) o en la revela negativa (-1) siguiendo el siguiente razonamiento:

$$class = \begin{cases} 1 & : prob \geq \theta \\ -1 & : prob < \theta \end{cases}$$

3.4. Diseño de nuevos experimentos

En el presente proyecto se ha hecho uso de *bagging* de clasificadores GP para solventar el sesgo que los clasificadores puedieran sufir debido al desbalanceamiento de la revela positiva respecto de la revela negativa.

No obstante, otra de resolver el desbalanceamiento de revelas consistiría en sobremuestreo de patrones sintéticos de la revela positiva minoritaria, usando desde métodos más simples como *SMOTE* (*Synthetic Minority Oversampling TEchnique*) hasta otros métodos más robustos como una combinación de *SMOTE* con métodos de submuestreo sobre la revela mayoritaria (e.g. SMOTE + Tomek Link).

Con la finalidad de evaluar de la forma más objetiva el rendimiento de este método, se partiría de los 5 *folds* anteriores y, para cada *fold*, se aplicaría, en primer lugar, el sobremuestreo (o sobremuestreo combinado con submuestreo) de las instancias de entrenamiento de la revela minoritaria.

A continuación, se procedería a entrenar un único modelo GP con el conjunto de entrenamiento y se evaluaría el rendimiento de este modelo con la partición de test inalterada. El proceso se repetiría para cada *fold*.