

Detección de eventos transitorios en imágenes astronómicas

Author: Bruno Sánchez

Mariano Domínguez

Marcelo Lares

Mario Díaz

Martín Beroiz

IATE - CONICET - Universidad Nacional de Córdoba (UNC)

April 7, 2017



Contents

La variabilidad en astronomía es referida al estudio del cambio del brillo de un objeto astronómico en el tiempo

La variabilidad en astronomía es referida al estudio del cambio del brillo de un objeto astronómico en el tiempo (*serie temporal-curvas de luz*)

Las series temporales en astronomía sirven mayormente para:

- ▶ Clasificación de objetos
- ▶ Estudio de NEO's
- ▶ Estudio de sistemas estelares
- ▶ Determinación de distancias
- ▶ Estudiar expansión del universo

Hay varios tipos de variabilidad:

- ▶ Variabilidad Periódica

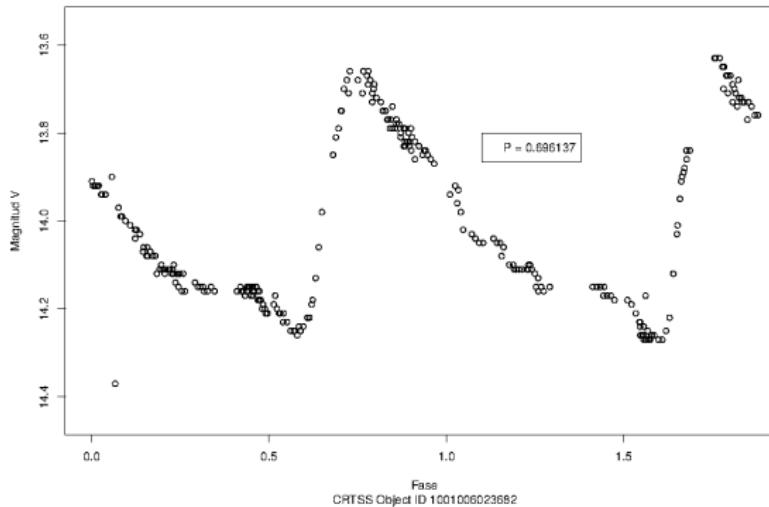


Figure: Curva de Luz de una estrella Variable Pulsante Clase RR Lyra. Realizado con datos de CRTSS.

Hay varios tipos de variabilidad:

- ▶ Variabilidad Periódica
- ▶ Variabilidad Transitoria

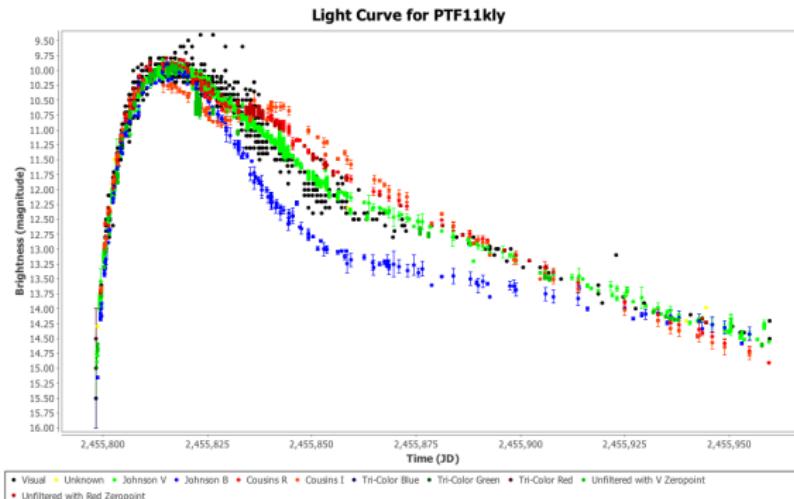


Figure: Curva de luz de Supernova Tipo Ia, correspondiente al evento PTF11kly. Datos de PTF.

Hay varios tipos de variabilidad:

- ▶ Variabilidad Periódica
- ▶ Variabilidad Transitoria

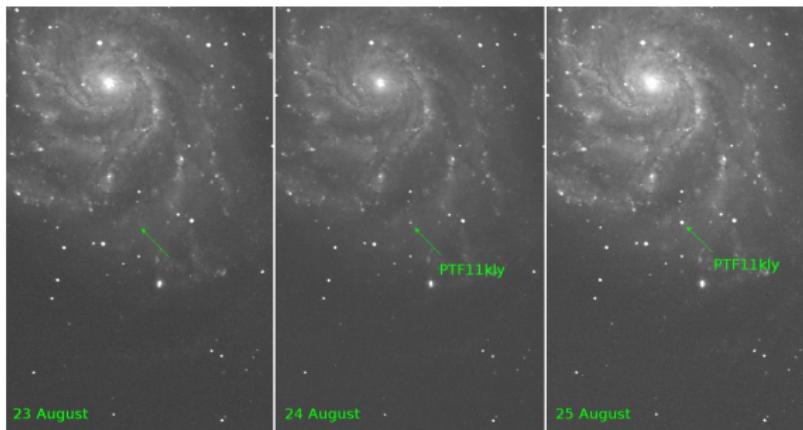


Figure: Curva de luz de Supernova Tipo Ia, correspondiente al evento PTF11kly. Datos de PTF.

La teoría de la Relatividad General (RG) admite en sus ecuaciones la existencia de **radiación gravitatoria**.

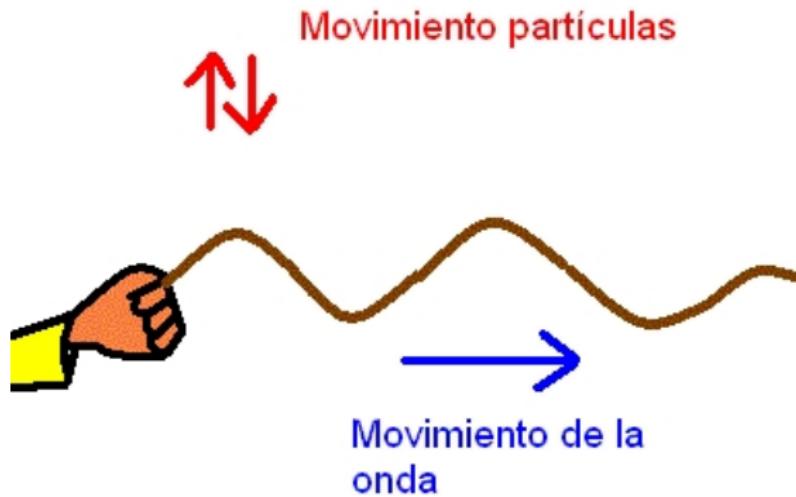
Estas son llamadas **Ondas Gravitacionales** (GW por sus siglas en inglés).

Las GW son **ondas transversales**: las perturbaciones que inducen son en el plano perpendicular a la dirección de propagación.

La teoría de la Relatividad General (RG) admite en sus ecuaciones la existencia de **radiación gravitatoria**.

Estas son llamadas **Ondas Gravitacionales** (GW por sus siglas en inglés).

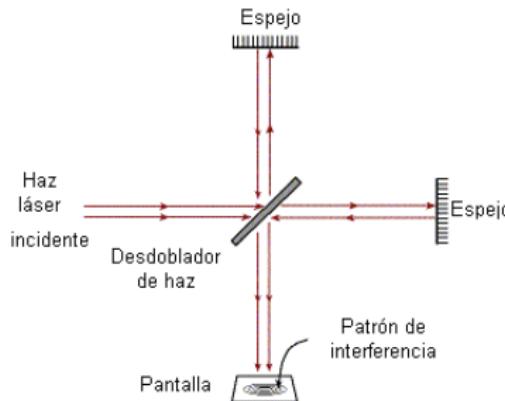
Las GW son **ondas transversales**: las perturbaciones que inducen son en el plano perpendicular a la dirección de propagación.



La detección se realiza mediante interferómetros de Michelson

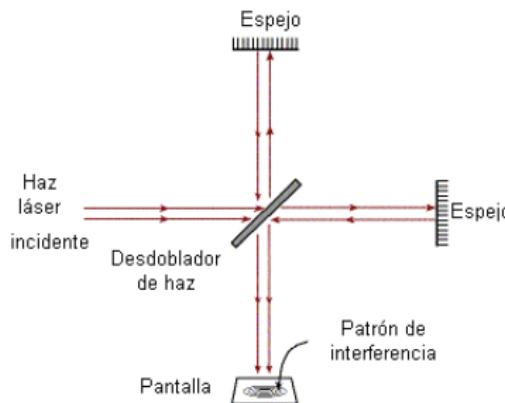
La detección se realiza mediante interferómetros de Michelson

Para incrementar la precisión de estos aparatos es necesario alargar la longitud de los brazos del interferómetro, y esto ha originado la construcción de enormes laboratorios con interferómetros con brazos de kilómetros de largo



La detección se realiza mediante interferómetros de Michelson

Para incrementar la precisión de estos aparatos es necesario alargar la longitud de los brazos del interferómetro, y esto ha originado la construcción de enormes laboratorios con interferómetros con brazos de kilómetros de largo



Otros detectores: Virgo, GEO 600, LCGT, AIGO, y LIGO.

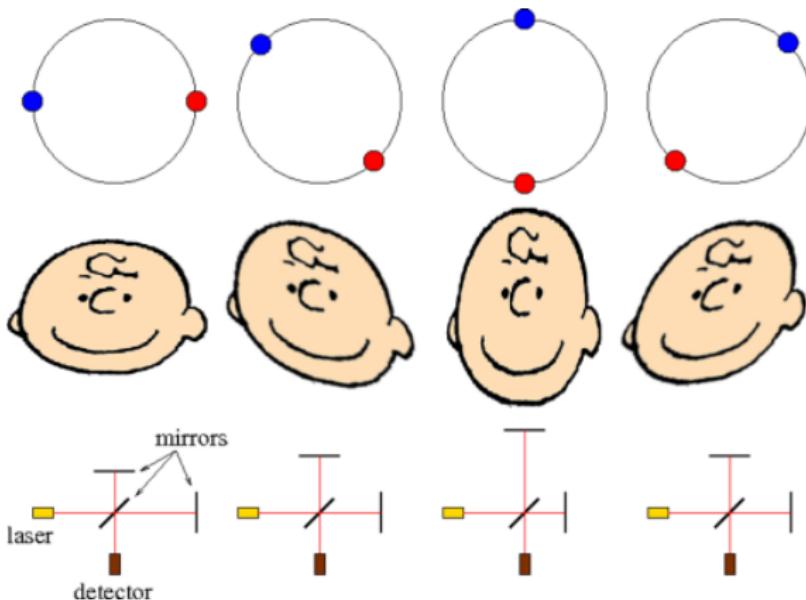


Figure: Esquema de las perturbaciones inducidas por las GW. Extraído de <http://www.einstein-online.info/spotlights/gravWav>.

Los problemas de localización de los eventos utilizando tan sólo detectores de GW son importantes.

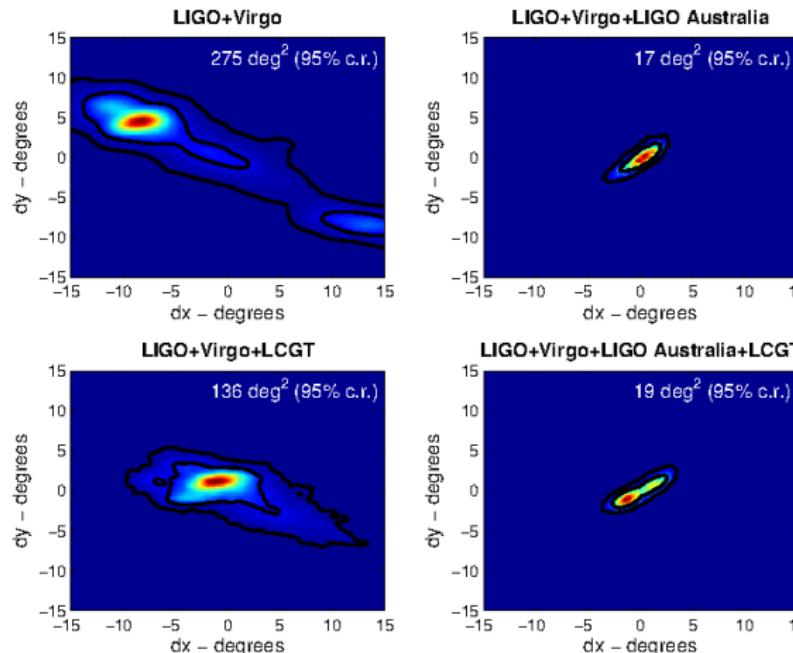


Figure: Localización en el cielo de eventos con S/N baja.

La detección de estos eventos entonces necesita **confirmación** mediante un método independiente.

Existe un candidato firme: la colisión y fusión de objetos compactos, como estrellas de neutrones, o agujeros negros, denominado **Kilonova**.

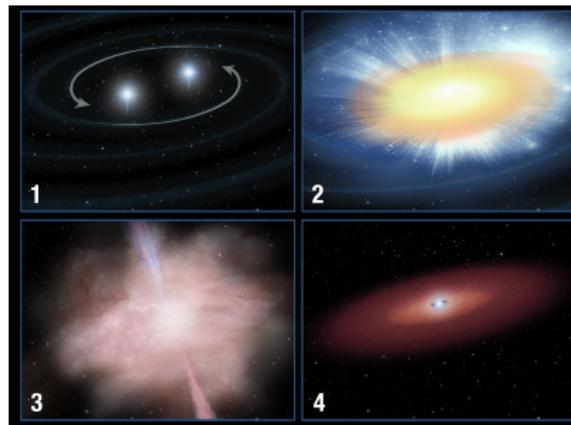


Figure: Modelo de fusión de objetos compactos: (1) Sistema binario de estrellas de neutrones; (2) concluida la etapa de acercamiento en espiral los objetos toman contacto; (3) los objetos se fusionan disparando un jet en la dirección perpendicular al plano orbital y expulsando material que emite radiación casi isotrópicamente; (4) el material eyectado decae radioactivamente y se apaga formando un disco. Extraído de <http://www.einstein-online.info/spotlights/gravWav>.

El evento de fusión posee las siguientes contrapartes electromagnéticas:

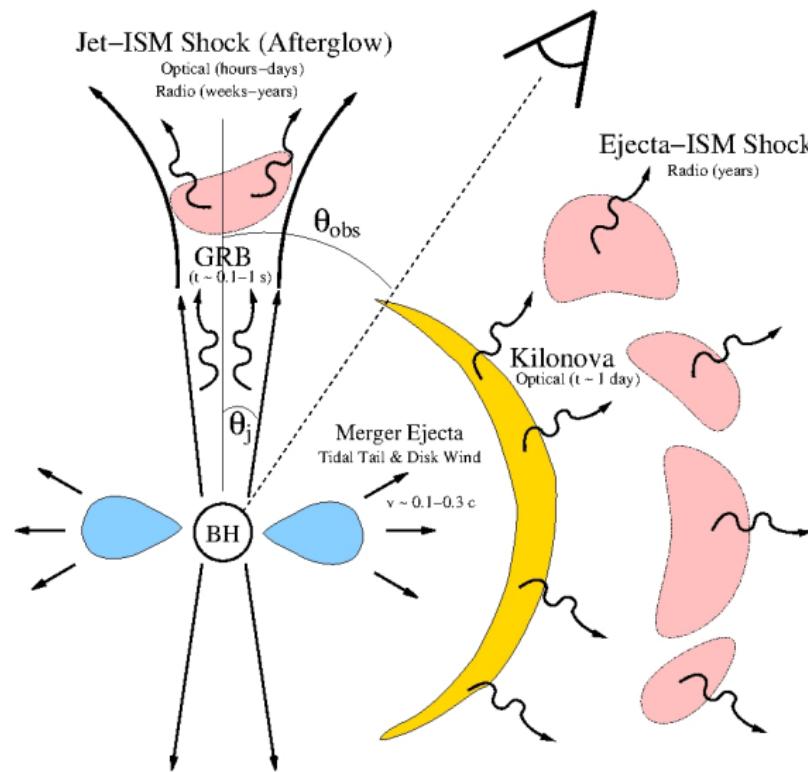


Figure: Esquema de Kilonova, de Metzger & Berger 2011.

Curva de luz del modelo de kilonova

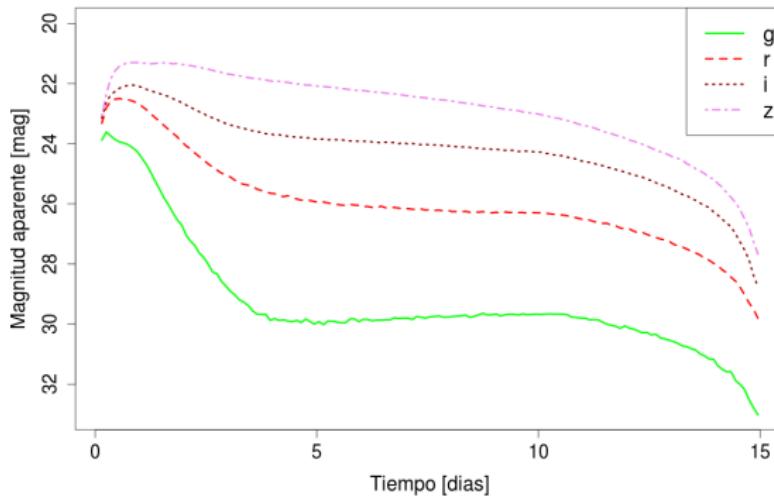


Figure: Curva de luz de Kilonova. Datos de Barnes & Kasen 2014 (de la colaboración de TOROS).

Resulta que Kilonovas observadas hay UNA.

Y es sólo un candidato.

Que hace falta para observar una Kilonova??

- ▶ Telescopio

Resulta que Kilonovas observadas hay UNA.

Y es sólo un candidato.

Que hace falta para observar una Kilonova??

- ▶ Telescopio
- ▶ Muchas observaciones

Resulta que Kilonovas observadas hay UNA.

Y es sólo un candidato.

Que hace falta para observar una Kilonova??

- ▶ Telescopio
- ▶ Muchas observaciones
- ▶ Detectar variabilidad de corta duración

Resulta que Kilonovas observadas hay UNA.

Y es sólo un candidato.

Que hace falta para observar una Kilonova??

- ▶ Telescopio
- ▶ Muchas observaciones
- ▶ Detectar variabilidad de corta duración
- ▶ Velocidad de procesado de datos

Resulta que Kilonovas observadas hay UNA.

Y es sólo un candidato.

Que hace falta para observar una Kilonova??

- ▶ Telescopio
- ▶ Muchas observaciones
- ▶ Detectar variabilidad de corta duración
- ▶ Velocidad de procesado de datos

Resulta que Kilonovas observadas hay UNA.

Y es sólo un candidato.

Que hace falta para observar una Kilonova??

- ▶ Telescopio
- ▶ Muchas observaciones
- ▶ Detectar variabilidad de corta duración
- ▶ Velocidad de procesado de datos

Proyecto TOROS - TORITOS



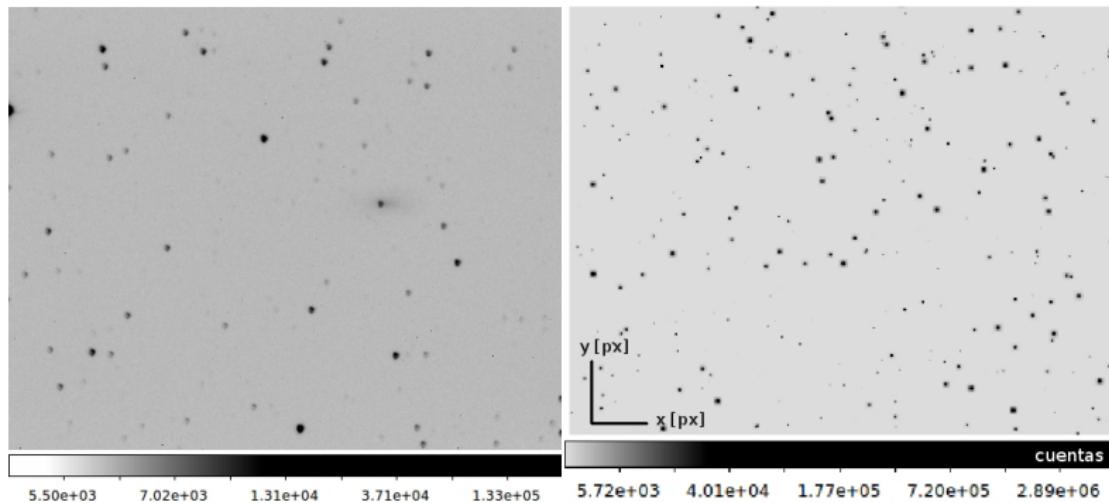
- ▶ TORITOS comenzaría a trabajar para el mes de septiembre, donde aLIGO va a tener una corrida científica de un mes aproximadamente.
- ▶ Para trabajar con un telescopio en el cerro necesitamos software que minimize la interacción con un ser humano.
- ▶ Junto con Martín Beroiz trabajamos en el desarrollo del *pipeline* de trabajo del instrumento.
- ▶ Aún así necesitamos saber de antemano sobre el *data product*

- ▶ TORITOS comenzaría a trabajar para el mes de septiembre, donde aLIGO va a tener una corrida científica de un mes aproximadamente.
- ▶ Para trabajar con un telescopio en el cerro necesitamos software que minimize la interacción con un ser humano.
- ▶ Junto con Martín Beroiz trabajamos en el desarrollo del *pipeline* de trabajo del instrumento.
- ▶ Aún así necesitamos saber de antemano sobre el *data product*
- ▶ La respuesta a esto son simulaciones de datos con cadencia temporal.

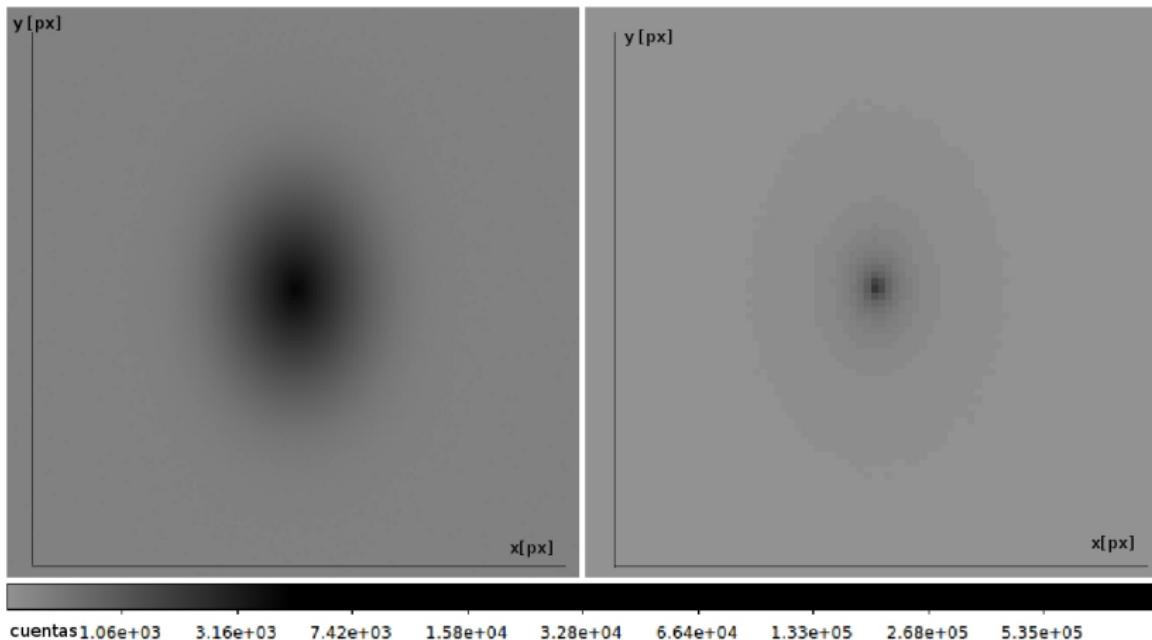
Para simular imágenes tuvimos que tener en cuenta detalles técnicos como:

- ▶ Difracción
- ▶ PSF
- ▶ Fondo del cielo
- ▶ Pixelizado
- ▶ Escalas fotométricas

¿Cuál es la falsa?



Diferentes galaxias



Análisis de las imágenes

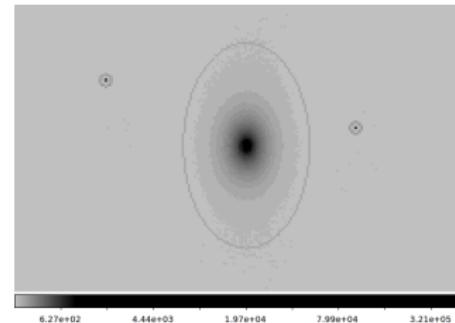
Para el análisis de las imágenes empleamos:

- ▶ Fotometría de Apertura

Análisis de las imágenes

Para el análisis de las imágenes empleamos:

- ▶ Fotometría de Apertura
 - ▶ Consta de utilizar “aperturas” para encerrar las fuentes astronómicas y calcular la energía recibida
 - ▶ $m_1 - m_2 = -2.5 \log_{10} \left(\frac{E_1}{E_2} \right)$

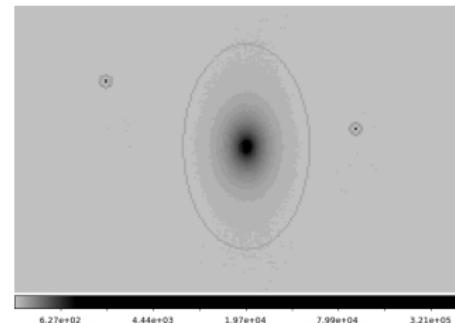


- ▶ Fotometría de Diferencias

Análisis de las imágenes

Para el análisis de las imágenes empleamos:

- ▶ Fotometría de Apertura
 - ▶ Consta de utilizar “aperturas” para encerrar las fuentes astronómicas y calcular la energía recibida
 - ▶ $m_1 - m_2 = -2.5 \log_{10} \left(\frac{E_1}{E_2} \right)$



- ▶ Fotometría de Diferencias
 - ▶ Consta de “restar” dos imágenes astronómicas
 - ▶ Formalmente involucra análisis de Fourier para calibrar ambas imágenes
 - ▶ Utilizamos resta directa ya que nuestras imágenes de por sí estaban calibradas

Algunos números

Resumiendo en números nuestras simulaciones:

- ▶ 3555 series temporales obtenidas
- ▶ 182490 imágenes simuladas
- ▶ Más de 28500 curvas de luz
- ▶ Se crearon otras 182490 imágenes de fotometría de diferencias
- ▶ 5,1 Terabytes de espacio total de memoria utilizado.

La Construcción de una Figura de Mérito

Se le llama Figura de Mérito a un estimador (o a un conjunto de estimadores) que caracterizan el rendimiento de un test estadístico. En el trabajo lidiamos con métodos que realizan múltiples pruebas por hipótesis. Todo proceso de este tipo tiene dos posibles errores:

-	Decisión	
Condición Real	Rechazar H_0	Mantener H_0
H_0 Verdadera	Error de Tipo I	Acierto
H_0 Falsa	Acierto	Error de Tipo II

La Construcción de una Figura de Mérito

Los estadísticos que miden el rendimiento de un test de decisión son FDR, TPR, y FPR.

- ▶ FDR (o *Tasa de Falsos Descubrimientos*) corresponde a la probabilidad de cometer un error de Tipo I habiendo obtenido como resultado el rechazo de H_0
- ▶ TPR (o *Tasa de Positivos Verdaderos*) corresponde a la probabilidad de rechazar la H_0 dado que es falsa, o bien uno menos la probabilidad de cometer un Error de Tipo I
- ▶ FPR (o *Tasa de Falsos Positivos*) el cual es la probabilidad de cometer un error de Tipo II

La Construcción de una Figura de Mérito

Las tasas de FPR y TPR dependen del valor umbral que se aplique a un dado método.

El valor umbral es quién procede a decidir si hubo o no detección y se asocia principalmente con el valor de confidencia que se pida a un test.

Las tasas de FPR y TPR poseen un balance que se caracteriza por la *Respuesta Característica del Operador*.

Y el estimador de rendimiento que usaremos será el AUC (*Área Bajo la Curva*)

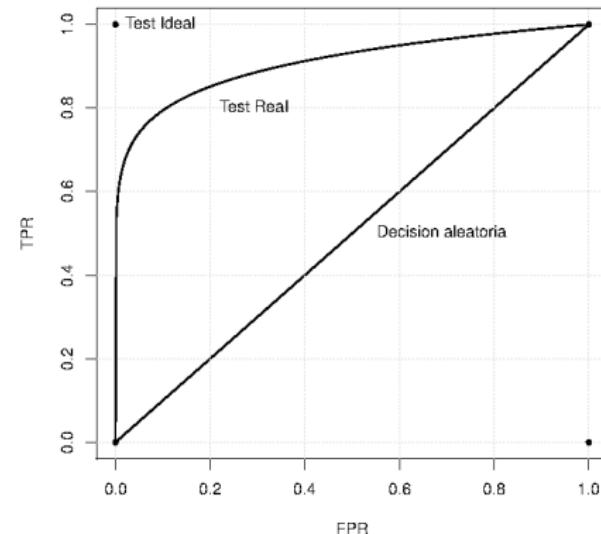


Figure: Curva ROC

La Construcción de una Figura de Mérito

Las tasas de FPR y TPR dependen del valor umbral que se aplique a un dado método.

El valor umbral es quién procede a decidir si hubo o no detección y se asocia principalmente con el valor de confidencia que se pida a un test.

Las tasas de FPR y TPR poseen un balance que se caracteriza por la *Respuesta Característica del Operador*.

Y el estimador de rendimiento que usaremos será el AUC (*Área Bajo la Curva*)

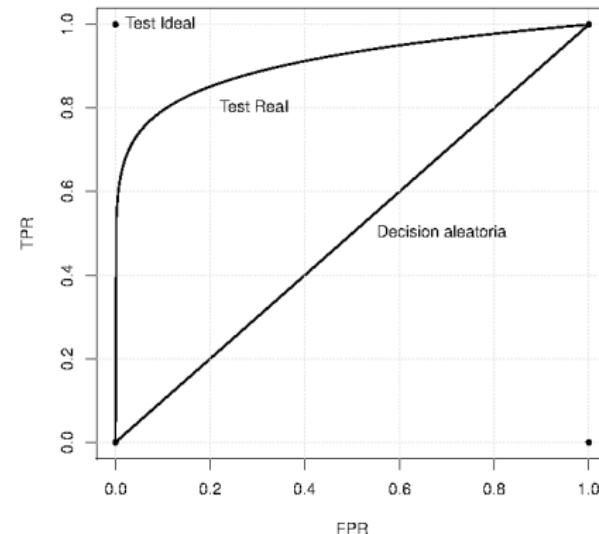


Figure: Curva ROC

Figura de Mérito para los Tests

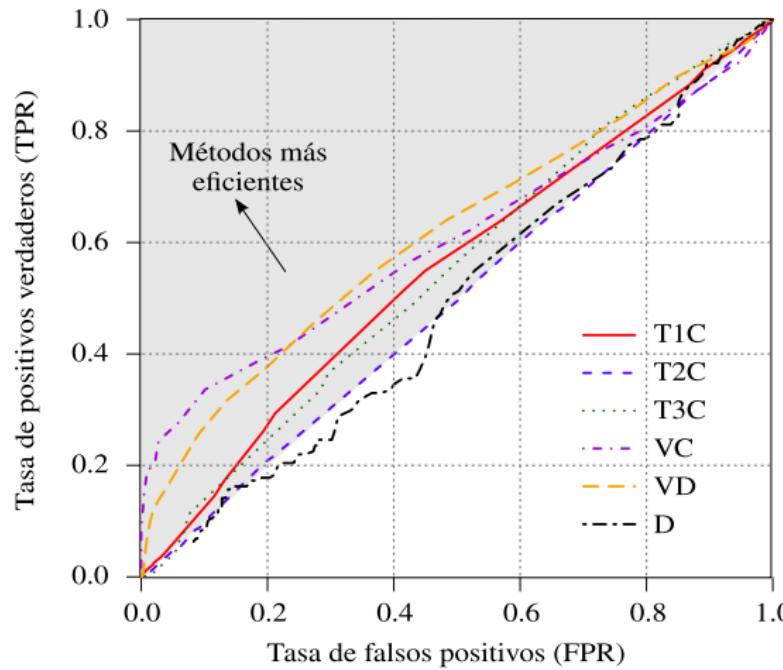


Figure: Curva ROC de nuestros tests de variabilidad.

Figura de Mérito para los Tests

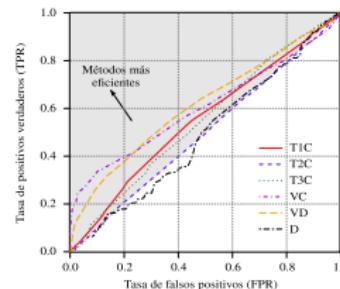


Figure: Curva ROC de nuestros tests de variabilidad.

Test	T1C	T2C	T3C	VC	VD	D
AUC	0.55	0.50	0.55	0.61	0.61	0.49

Resultados de la Fotometría de Diferencias

El nivel de corte fue de 2σ Hallamos:

- ▶ Para la Muestra de Control (sin kilonova)
 - ▶ 1231 detecciones sobre un total de 1440 series temporales
 - ▶ 85% de falsas detecciones
- ▶ Para la Muestra con Kilonova
 - ▶ 1862 detecciones en un total de 2115 series temporales
 - ▶ 90% de detecciones certeras
 - ▶ 68% de objetos espúreos.

Se puede ver que le erramos por bastante.

El problema es que el nivel de corte es muy bajo, pero un corte más alto nos haría perder las Kilonovas que nos están llamando por ahí

Resultados de la Fotometría de Diferencias

El nivel de corte fue de 2σ Hallamos:

- ▶ Para la Muestra de Control (sin kilonova)
 - ▶ 1231 detecciones sobre un total de 1440 series temporales
 - ▶ 85% de falsas detecciones
- ▶ Para la Muestra con Kilonova
 - ▶ 1862 detecciones en un total de 2115 series temporales
 - ▶ 90% de detecciones certeras
 - ▶ 68% de objetos espúreos.

Se puede ver que le erramos por bastante.

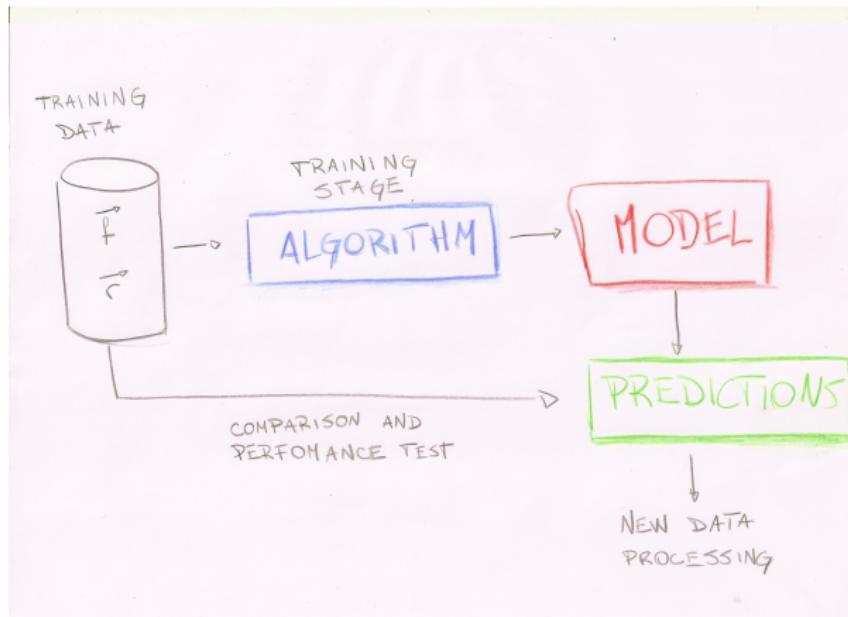
El problema es que el nivel de corte es muy bajo, pero un corte más alto nos haría perder las Kilonovas que nos están llamando por ahí
Que hacer??

Statistical Learning

Hoy en día mucha ciencia se hace a través de lo que *los datos dicen*:

Extraer ese contenido de información es lo que significa **aprender de los datos** con Machine Learning (ML):

Supervised learning



Algoritmos implementados

Lo que queremos entonces es aprender a diferenciar un transitorio real de uno falso.

Estos problemas se atacan utilizando las llamadas **features** (en la jerga de ML) los cuales son los observables asociados a un set de datos -llamado **set de entrenamiento**- desde el cual el algoritmo debe construir un **modelo**.

Nosotros usamos tres algoritmos clásicos del ML:

- ▶ **Naive Bayes**
- ▶ **Logistic Regression**
- ▶ **Random Forest**

Algoritmos implementados

Lo que queremos entonces es aprender a diferenciar un transitorio real de uno falso.

Estos problemas se atacan utilizando las llamadas **features** (en la jerga de ML) los cuales son los observables asociados a un set de datos -llamado **set de entrenamiento**- desde el cual el algoritmo debe construir un **modelo**.

Nosotros usamos tres algoritmos clásicos del ML:

- ▶ **Naive Bayes**
- ▶ **Logistic Regression**
- ▶ **Random Forest**

Algoritmos implementados

Lo que queremos entonces es aprender a diferenciar un transitorio real de uno falso.

Estos problemas se atacan utilizando las llamadas **features** (en la jerga de ML) los cuales son los observables asociados a un set de datos -llamado **set de entrenamiento**- desde el cual el algoritmo debe construir un **modelo**.

Nosotros usamos tres algoritmos clásicos del ML:

- ▶ **Naive Bayes** Usa el teorema de Bayes para estimar probabilidades de pertenecer a cada clase:

$$P(c = c_i | \vec{f}) = \frac{P(\vec{f} | c = c_i) P(c = c_i)}{P(\vec{f})}$$

- ▶ **Logistic Regression**
- ▶ **Random Forest**

Algoritmos implementados

Lo que queremos entonces es aprender a diferenciar un transitorio real de uno falso.

Estos problemas se atacan utilizando las llamadas **features** (en la jerga de ML) los cuales son los observables asociados a un set de datos -llamado **set de entrenamiento**- desde el cual el algoritmo debe construir un **modelo**.

Nosotros usamos tres algoritmos clásicos del ML:

- ▶ **Naive Bayes**
- ▶ **Logistic Regression** Este algoritmo asume que las probabilidades de pertenecer a una dada clase puede ser linealmente interpolada luego de una transformación muy simple:

$$\text{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta \times \vec{f}_i$$

- ▶ **Random Forest**

Algoritmos implementados

Lo que queremos entonces es aprender a diferenciar un transitorio real de uno falso.

Estos problemas se atacan utilizando las llamadas **features** (en la jerga de ML) los cuales son los observables asociados a un set de datos -llamado **set de entrenamiento**- desde el cual el algoritmo debe construir un **modelo**.

Nosotros usamos tres algoritmos clásicos del ML:

- ▶ **Naive Bayes**
- ▶ **Logistic Regression**
- ▶ **Random Forest** Este algoritmo usa muchos árboles de decisión, y posee dos parámetros importantes: N_{tree} el número de árboles de decisión, y N_f el número de features al azar en cada árbol.
Después del entrenamiento la clasificación es calculada en una votación entre todos los árboles.

Construimos el set de entrenamiento

Simulamos nuevamente datos, produciendo un conjunto balanceado de 9202 *bogus* y 9120 *reals*.

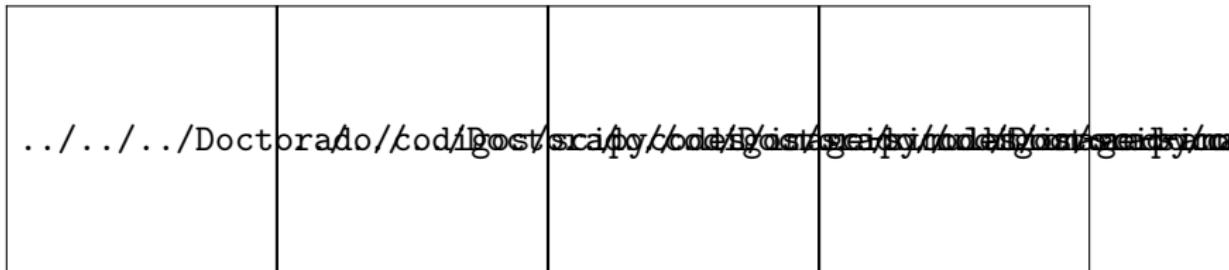
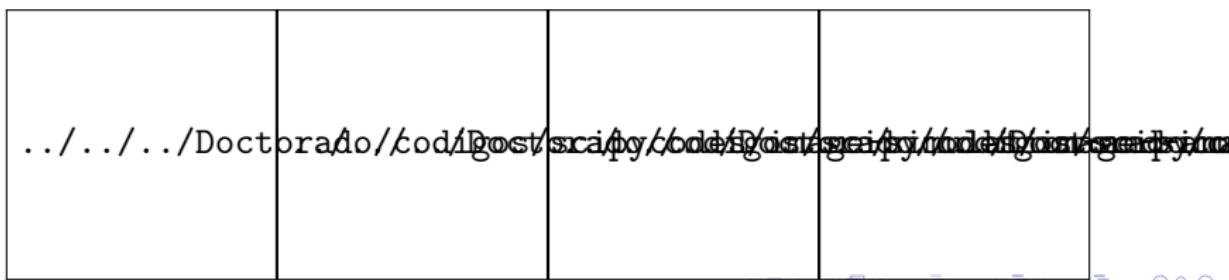


Figure: Estampillas de objetos *bogus* simulados.



Procedimiento

- ▶ Implementamos un **vectorizador**, el cual simplemente usa datos de los pixeles y extrae un enorme conjunto de features -llega hasta 3000 features-, y usamos un conjunto reducido de **1028 features**.

Procedimiento

- ▶ Implementamos un **vectorizador**, el cual simplemente usa datos de los pixeles y extrae un enorme conjunto de features -llega hasta 3000 features-, y usamos un conjunto reducido de **1028 features**.
- ▶ Luego usamos el **WEKA Explorer suit** y analizamos los features. El suit tiene muchas herramientas de ML para jugar, incluyendo **selección de features**.

Procedimiento

- ▶ Implementamos un **vectorizador**, el cual simplemente usa datos de los pixeles y extrae un enorme conjunto de features -llega hasta 3000 features-, y usamos un conjunto reducido de **1028 features**.
- ▶ Luego usamos el **WEKA Explorer suit** y analizamos los features. El suit tiene muchas herramientas de ML para jugar, incluyendo **selección de features**.
- ▶ Usamos dos rutinas: PCA, y *CfSubsetEval* de Hall (1998). Esto reduce el conjunto de 1028 features principalmente para descartar interdependencias.

Procedimiento

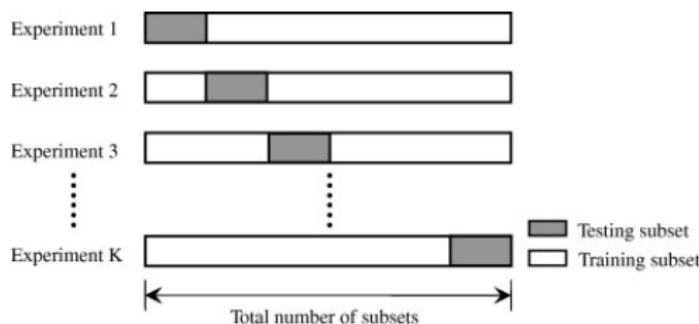
- ▶ Implementamos un **vectorizador**, el cual simplemente usa datos de los pixeles y extrae un enorme conjunto de features -llega hasta 3000 features-, y usamos un conjunto reducido de **1028 features**.
- ▶ Luego usamos el **WEKA Explorer suit** y analizamos los features. El suit tiene muchas herramientas de ML para jugar, incluyendo **selección de features**.
- ▶ Usamos dos rutinas: PCA, y *CfSubsetEval* de Hall (1998). Esto reduce el conjunto de 1028 features principalmente para descartar interdependencias.
- ▶ PCA arrojó 310 *principal components*, y nos proporcionó la matriz de transformación de los datos, y los vectores de features ya proyectados al espacio de las PC.

Procedimiento

- ▶ Implementamos un **vectorizador**, el cual simplemente usa datos de los pixeles y extrae un enorme conjunto de features -llega hasta 3000 features-, y usamos un conjunto reducido de **1028 features**.
- ▶ Luego usamos el **WEKA Explorer suit** y analizamos los features. El suit tiene muchas herramientas de ML para jugar, incluyendo **selección de features**.
- ▶ Usamos dos rutinas: PCA, y *CfSubsetEval* de Hall (1998). Esto reduce el conjunto de 1028 features principalmente para descartar interdependencias.
- ▶ PCA arrojó 310 *principal components*, y nos proporcionó la matriz de transformación de los datos, y los vectores de features ya proyectados al espacio de las PC.
- ▶ El método de *CfSubsetEval* arrojó 39 features útiles.

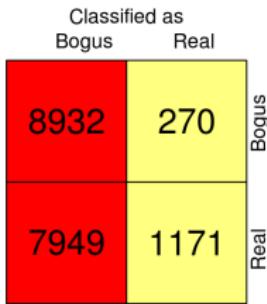
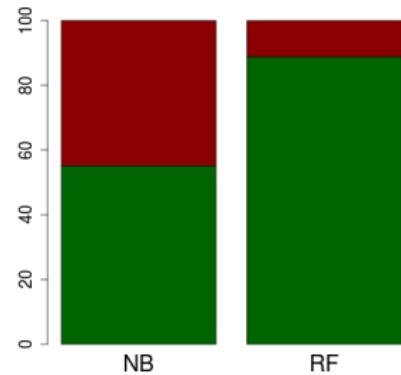
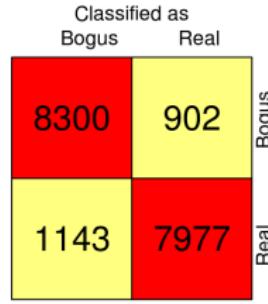
K-fold cross validation

La performance de los métodos se calculó usando *K-Fold Cross Validation*, esto es dividir los datos en **K porciones muestradas al azar**, entrenar en $K-1$ porciones y testear sobre la porción restante.



Esto se hace K veces, y esto arroja K medidas de performance, que es de donde se calculan los estadísticos TPR y FPR.

Resultados muestra completa de features

Naive Bayes**Random Forest**

Resultados para transformada con PCA

Naive Bayes

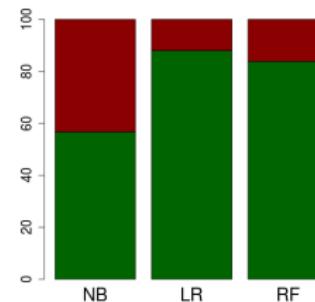
		Classified as
		Bogus
Bogus	Classified as	
	Bogus	8382
Real	Real	820
	Bogus	7110
		2010

Random Forest

		Classified as
		Bogus
Bogus	Classified as	
	Bogus	7801
Real	Real	1401
	Bogus	1569
		7551

Logistic Regression

		Classified as
		Bogus
Bogus	Classified as	
	Bogus	8136
Real	Real	1066
	Bogus	1122
		7998



Resultados features seleccionados con Cfs

Naive Bayes

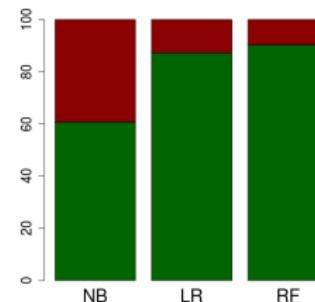
		Classified as
		Bogus
Classified as	Bogus	8871
	Real	331
		Bogus
Classified as	Real	6866
	Real	2254

Random Forest

		Classified as
		Bogus
Classified as	Bogus	8407
	Real	795
		Real
Classified as	Real	987
	Real	8133

Logistic Regression

		Classified as
		Bogus
Classified as	Bogus	8117
	Real	1085
		Bogus
Classified as	Real	1265
	Real	7855



Resumen y conclusiones

- ▶ Implementamos algoritmos de ML para clasificar imágenes, usando vectorizadores y métodos de selección de features.

Resumen y conclusiones

- ▶ Implementamos algoritmos de ML para clasificar imágenes, usando vectorizadores y métodos de selección de features.
- ▶ Los resultados son esperanzadores, y pueden ser aplicados de forma simple y con la posibilidad de determinar errores con confianza.

Resumen y conclusiones

- ▶ Implementamos algoritmos de ML para clasificar imágenes, usando vectorizadores y métodos de selección de features.
- ▶ Los resultados son esperanzadores, y pueden ser aplicados de forma simple y con la posibilidad de determinar errores con confianza.
- ▶ La flexibilidad que tiene esta solución (especialmente por el vectorizador) hace que sea expandible a otras situaciones nuevas.

Trabajo a futuro

- ▶ Correr estos clasificadores contra bases de datos públicas, tales como PTF (trabajo en preparación)

Trabajo a futuro

- ▶ Correr estos clasificadores contra bases de datos públicas, tales como PTF (trabajo en preparación)
- ▶ Hacer *fine tuning* de los clasificadores

Trabajo a futuro

- ▶ Correr estos clasificadores contra bases de datos públicas, tales como PTF (trabajo en preparación)
- ▶ Hacer *fine tuning* de los clasificadores
- ▶ Usarlo para TOROS/TORITOS (deployment on spring).

Trabajo a futuro

- ▶ Correr estos clasificadores contra bases de datos públicas, tales como PTF (trabajo en preparación)
- ▶ Hacer *fine tuning* de los clasificadores
- ▶ Usarlo para TOROS/TORITOS (deployment on spring).

Trabajo a futuro

- ▶ Correr estos clasificadores contra bases de datos públicas, tales como PTF (trabajo en preparación)
- ▶ Hacer *fine tuning* de los clasificadores
- ▶ Usarlo para TOROS/TORITOS (deployment on spring).

PREGUNTAS?? :D