**Surf-Forecaster**

Un enfoque para predicción del oleaje basado en análisis de regresión con técnicas de aprendizaje supervisado

Por

**Maximiliano Paolucci y Esteban Wagner**

Trabajo final de la carrera de grado de

Ingeniero de Sistemas

de la

Universidad Nacional del Centro de la Provincia de Buenos Aires



Directora: Dra. Daniela Lis Godoy

Tandil, Argentina. Abril de 2010

**Agradecimientos**

A nuestros padres y demás familiares que nos brindaron su afecto y colaboración durante estos años de estudio.

Un especial agradecimiento a nuestro directora Dra. Daniela Lis Godoy, por su colaboración, disposición y confianza brindadas durante el desarrollo de este trabajo.

A nuestros amigos y compañeros, los cuales supieron estar a nuestro lado en forma incondicional durante todo este tiempo, y de los cuales nos sentimos muy contentos y orgullosos.

¡Gracias!

**Índice General**

[Capítulo 1 - Introducción 7](#_Toc262075996)

[1.1 Análisis de regresión con técnicas de aprendizaje supervisado 7](#_Toc262075997)

[1.2 Caso de estudio 8](#_Toc262075998)

[1.3 Trabajos relacionados 9](#_Toc262075999)

[1.4 Trabajo realizado 9](#_Toc262076000)

[1.5 Estructura del trabajo 12](#_Toc262076001)

[Capitulo 2 - Estado del arte 14](#_Toc262076002)

[2.1 Aprendizaje supervisado 14](#_Toc262076003)

[2.2.1 Regresión Lineal Simple 16](#_Toc262076004)

[2.2.2 Redes neurales para regresión 18](#_Toc262076005)

[2.2.3 Árboles modelos y de regresión 21](#_Toc262076006)

[2.2.4 Maquinas de Soporte Vectorial para regresión 23](#_Toc262076007)

[2.3 Problema de predicción de oleaje 25](#_Toc262076008)

[2.4 Trabajos relacionados a la predicción de oleaje 27](#_Toc262076009)

[Capitulo 3 – Datos para predicción de oleaje 29](#_Toc262076010)

[3.1 Olas – Conceptos generales 29](#_Toc262076011)

[3.2 Observaciones Visuales 31](#_Toc262076012)

[3.3 Modelo WAVEWATCH III 33](#_Toc262076013)

[3.4 Pre procesamiento 35](#_Toc262076014)

[3.5 Modelos de instancia 36](#_Toc262076015)

[3.6 Conjuntos de entrenamiento 37](#_Toc262076016)

[Capítulo 4 - Experimentación 38](#_Toc262076017)

[4.1 Consideraciones generales 38](#_Toc262076018)

[4.1.1 Parametrización 38](#_Toc262076019)

[4.1.2 Medidas de evaluación utilizadas 38](#_Toc262076020)

[4.2 Modelos de instancia 39](#_Toc262076021)

[4.3 Resultados generales 39](#_Toc262076022)

[4.3.1 Resultados generales en Ala Moana 39](#_Toc262076023)

[4.3.2 Resultados generales en Diamond Head 40](#_Toc262076024)

[4.3.3 Resultados generales en North Shore 42](#_Toc262076025)

[4.3.4 Resultados generales en Windward 43](#_Toc262076026)

[4.3.5 Resultados generales en West Shore 44](#_Toc262076027)

[4.4 Algoritmo de aprendizaje y modelo de instancia seleccionado 45](#_Toc262076028)

[Capitulo 5 - Desarrollo de la aplicación 46](#_Toc262076029)

[5.1 Descripción de la aplicación 46](#_Toc262076030)

[5.1.1 Que diferencia a Surf-Forecaster de los demás sistemas de pronósticos ya existentes? 46](#_Toc262076031)

[5.1.2 Como logra pronosticar Surf-Forecaster? 46](#_Toc262076032)

[5.2 Tipos de usuario 48](#_Toc262076033)

[5.2.1 Usuario no registrado u anónimo 48](#_Toc262076034)

[5.2.2 Usuario registrado 48](#_Toc262076035)

[5.2.3 Administrador 48](#_Toc262076036)

[5.3 Secciones de la aplicación 49](#_Toc262076037)

[5.3.1 Pronóstico 49](#_Toc262076038)

[5.3.2 Comparación de olas 53](#_Toc262076039)

[5.3.3 Nueva ola 60](#_Toc262076040)

[5.3.4 Mis olas 63](#_Toc262076041)

[Capítulo 6 - Conclusiones 66](#_Toc262076042)

[6.1 Resumen 66](#_Toc262076043)

[6.2 Conclusiones del trabajo 66](#_Toc262076044)

[6.3 Ventajas del enfoque 67](#_Toc262076045)

[6.4 Limitaciones 68](#_Toc262076046)

[Referencias 70](#_Toc262076047)

**índice de figuras**

[Fig. 2.1: Proceso de clasificación. (Adaptado de [8]) 15](#_Toc262076054)

[Tabla 2.1: Relación años de experiencia, salario percibido ([14]). 17](#_Toc262076055)

[Fig. 2.2: Gráfico de los datos de la tabla 2.1 junto con la recta de regresión calculada utilizando el método de regresión lineal simple 17](#_Toc262076056)

[Fig. 2.3: Modelo de una neurona artificial. 19](#_Toc262076057)

[Fig. 2.4: Arquitectura de red neuronal Multilayer Feedforward. 19](#_Toc262076058)

[Fig. 2.5: Árbol modelo ([1]). 23](#_Toc262076059)

[Fig. 2.6: Regresión con soporte de vectores: (a) ɛ = 1, (b) ɛ = 2, y (c) ɛ = 0.5 25](#_Toc262076060)

[Fig. 2.7: Puntos de pronóstico del modelo de alta mar. 27](#_Toc262076061)

[Tabla 2.2: Modelos de regresión utilizados en trabajos relacionados a la predicción de la altura de ola. 28](#_Toc262076062)

[Fig. 3.1: Partes de la ola. 29](#_Toc262076063)

[Fig. 3.2: Grafico polar del espectro de olas ([36]). 30](#_Toc262076064)

[Fig. 3.3: Lugares de la isla de Oahu, Hawái con reportes de surf disponibles en la base de datos Goddard-Caldwell. 32](#_Toc262076065)

[Fig. 3.4: Mediciones de olas, HSF (Hawaiian Scale Feet) vs base-cresta. 32](#_Toc262076066)

[Fig. 3.5: Transformación de HSF a Base-cresta tomando como referencia la altura del surfista. 33](#_Toc262076067)

[Fig. 3.6: Vista general de las entradas y salidas de un modelo de olas genérico ([36]). 34](#_Toc262076068)

[Fig. 3.7: Puntos de la grilla global (1⁰ x 1. 25⁰) del sistema NOAA WAVEWATCH III 2.22 alrededor de Oahu, Hawái. 35](#_Toc262076069)

[Tabla 3.1: Conjuntos de entrenamiento generados para la playa North-Shore. 37](#_Toc262076070)

[Tabla 4.1: Algoritmos de aprendizaje de maquina estudiados 38](#_Toc262076071)

[Tabla 4.2: Resultados generales de pruebas sobre la ola “Ala Moana” 40](#_Toc262076072)

[Tabla 4.3: Resultados generales de pruebas sobre la ola “Diamond Head” 41](#_Toc262076073)

[Tabla 4.4: Resultados generales de pruebas sobre la ola “North Shore” 42](#_Toc262076074)

[Tabla 4.5: Resultados generales de pruebas sobre la ola “Windward” 44](#_Toc262076075)

[Tabla 4.6: Resultados generales de pruebas sobre la ola “West Shore” 45](#_Toc262076076)

[Fig. 5.1: Panel de localización de olas 49](#_Toc262076077)

[Fig. 5.2: Información de ola seleccionada 49](#_Toc262076078)

[Fig. 5.3: Condiciones actuales de la ola seleccionada. 50](#_Toc262076079)

[Fig. 5.4: Condiciones a partir de las próximas 3 horas de la ola seleccionada. 51](#_Toc262076080)

[Fig. 5.5: Fragmento de tabla de pronósticos detallados para la ola Pipeline en Oahu – Hawái. 52](#_Toc262076081)

[Fig. 5.6: Fragmento de tabla de pronósticos detallados minimizada. 53](#_Toc262076082)

[Fig. 5.7: Vista de pantalla de generación de comparaciones – Usuario no registrado. 54](#_Toc262076083)

[Fig. 5.8: Menú de localización de olas en pantalla de generación de comparaciones 54](#_Toc262076084)

[Fig. 5.9: de pantalla de generación de comparaciones – Usuario registrado u Administrador. 55](#_Toc262076085)

[Fig. 5.10: Guardar comparación. 56](#_Toc262076086)

[Fig. 5.11: Mensaje de confirmación para sobrescribir una comparación. 56](#_Toc262076087)

[Fig. 5.12: Mensaje de confirmación para eliminar una comparación. 57](#_Toc262076088)

[Fig. 5.13: Selección del pronosticador deseado para cada ola. 57](#_Toc262076089)

[Fig. 5.14: Altura de las olas en las próximas horas. 58](#_Toc262076090)

[Fig. 5.15: Captura de animación con la altura de cada ola a lo largo de las 180hs de pronóstico. Vista de gráfico de barras. 58](#_Toc262076091)

[Fig. 5.16: Gráfico con la altura de cada ola a lo largo de las 180hs de pronóstico. Vista de gráfico de líneas. 59](#_Toc262076092)

[Fig. 5.17: Fragmento de tabla de pronósticos detallado en sección de comparaciones. 59](#_Toc262076093)

[Fig. 5.18: Nueva ola, información básica 60](#_Toc262076094)

[Fig. 5.19: Nueva ola, ubicarla en el mapa. 61](#_Toc262076095)

[Fig. 5.20: Nueva ola, ya ubicada en el mapa. 62](#_Toc262076096)

[Fig. 5.21: Nueva ola, entrenar un pronosticador 62](#_Toc262076097)

[Fig. 5.22: Nueva ola, formato del archive de observaciones visuales. 63](#_Toc262076098)

[Fig. 5.23: Mis olas, tabla de olas pertenecientes al usuario actualmente logueado. 64](#_Toc262076099)

[Fig. 5.24: Confirmación de eliminación de la ola seleccionada. 64](#_Toc262076100)

[Fig. 5.25: Mis olas – Edición de ola, pronosticador especializado. 64](#_Toc262076101)

# Capítulo 1 - Introducción

## 1.1 Análisis de regresión con técnicas de aprendizaje supervisado

La Minería de Datos o Data Mining (DM) agrupa a un conjunto de herramientas y métodos orientados al descubrimiento de conocimiento a partir de bases de datos (KDD), del inglés Knowledge Discovery from Databases. KDD integra elementos como métodos estadísticos, aprendizaje de máquina y visualización de datos. Proporciona la capacidad para descubrir información nueva y significativa usando los datos existentes. KDD rápidamente excede la capacidad humana para analizar grandes cantidades de datos. La cantidad de datos que requieren procesamiento y análisis en grandes bases de datos exceden las capacidades humanas y la dificultad de transformar los datos con precisión es un conocimiento que va más allá de los límites de las bases de datos tradicionales. Por consiguiente, la utilización plena de los datos almacenados depende del uso de técnicas del descubrimiento del conocimiento.

Las siguientes son características básicas que comparten todas las técnicas KDD:

* Todos los enfoques tratan con grandes cantidades de datos
* Se requiere de eficiencia debido al volumen de datos
* La Exactitud es un elemento esencial
* Todos requieren el uso de un lenguaje de alto nivel
* Todos los enfoques usan alguna forma de aprendizaje automatizado
* Todos producen algunos resultados interesantes

La utilidad de aplicaciones futuras en KDD es de largo alcance. KDD puede usarse como un medio de recuperación de información, de la misma manera que los agentes inteligentes realizan la recuperación de información en el Web. Nuevos modelos o tendencias en los datos podrán descubrirse usando estas técnicas.

Los algoritmos de aprendizaje son una parte integral de KDD, dentro del área de aprendizaje de máquina el aprendizaje supervisado es una técnica que consiste en aprender una función a partir de un conjunto de datos de entrenamiento. El conjunto de entrenamiento consiste en pares de objetos de entrada y salida esperada. Utilizando estos datos se aprenderá una función que luego será capaz de predecir la salida esperada utilizando datos de entrada nunca vistos. Dedicado al desarrollo de algoritmos que aprenden a partir de ejemplos. Un ejemplo está representado por una instancia y una clase a la que esa instancia pertenece. Una instancia representa un conjunto de observaciones sobre un hecho determinado. Una clase representa la clasificación que se le da al hecho. A partir de un conjunto de ejemplos y un algoritmo de aprendizaje supervisado podemos entrenar un clasificador para que el mismo pueda predecir cuál será la clase de una instancia que desconocemos. (Ver capitulo 2 – Estado del arte)

Dentro del aprendizaje supervisado existen diferentes métodos de predicción, los cuales son más o menos eficientes de acuerdo a la naturaleza de los datos del problema en cuestión. Ejemplos de ellos son las maquinas de soporte vectorial, redes neuronales, árboles de decisión, etc. [1]. Estos algoritmos son utilizados en diversas áreas, incluyendo agricultura, finanzas, historia, industria, relaciones laborales y ciencias del medioambiente, entre otras.

Cuando la salida de la función aprendida es un valor continuo la tarea de aprendizaje se denomina regresión. El análisis de regresión es un conjunto de técnicas de modelado y análisis numérico la cual consiste en determinar el valor de una variable dependiente a partir de un conjunto de variables independientes. Esta técnica es utilizada para realizar pronósticos, inferencia, prueba de hipótesis y evaluación de causalidad entre otras cosas.

Usualmente se ha utilizado a este efecto la regresión lineal, pero los modelos obtenidos sólo operan con atributos numéricos e imponen una dependencia puramente lineal. Otros métodos de regresión son los provenientes del área de aprendizaje supervisado como los árboles de regresión y las redes neurales para regresión. Los árboles de regresión [3] se generan de forma similar a los de decisión, con valores promedio en cada hoja. El modelo neuronal de regresión generalizada (GRNN), que fue propuesto y desarrollado inicialmente por Specht [2], posee la propiedad deseable de no requerir ningún entrenamiento iterativo, es decir, puede aproximar cualquier función arbitraria entre vectores de entrada y salida, realizando la estimación de la función directamente a partir de los datos de entrenamiento.

Nuestra propuesta consiste en la evaluación de un conjunto de algoritmos de regresión provenientes del área de aprendizaje supervisado y su comparación con modelos de regresión clásicos en el marco de un caso de estudio específico. A partir de los resultados experimentales que se obtengan de estos algoritmos, se seleccionará el método de mejor performance para ser usado en un entorno real para el problema de predicción que se abordará.

## 1.2 Caso de estudio

En la actualidad existen una variedad de deportes acuáticos que se ven beneficiados por los pronósticos acerca del oleaje en una zona cercana a la costa. Ejemplos de estos pueden ser el surf, windsurf, kitesurf, kayak, etc. Los deportistas están sumamente atentos a las condiciones que presenta el mar diariamente para evaluar la playa en la que van a practicar su deporte. Utilizando un pronóstico de oleaje el deportista puede planificar sus actividades para lograr estar presente en el lugar que presenta las mejores condiciones. Eventualmente el deportista puede evitarse recorridas extensas buscando el sitio con las mejores condiciones del día.

Actualmente los pronósticos de oleaje son derivados a partir de la resolución de un modelo matemático global sumamente complejo cuyas predicciones son calculadas en zonas alejadas de la costa, varios kilómetros mar adentro. Este provee de la predicción de la altura de las olas, la dirección de las mismas y el período (tiempo entre una ola y la siguiente). Si bien este pronóstico tiene buen rendimiento en alta mar, no sucede lo mismo en la cercanía de la costa. Esto se debe a que la interacción de las olas con el lecho marino cercano no es contemplada por estos modelos. De acuerdo a la geografía de la costa puede suceder que en un rango de 15 km un lugar presente condiciones apropiadas para la práctica del deporte mientras que en otro las condiciones sean totalmente desfavorables. Por esta razón es que la utilización de un pronóstico preciso en las cercanías de la costa (exactamente donde el mar es lo suficientemente poco profundo para que las olas rompan) es de suma utilidad.

Existen modelos para contemplar el comportamiento de las olas en las cercanías de la costa. Los mismos reciben como entrada los datos de pronóstico de alta mar e información acerca del lecho marino. Utilizando estos datos los mismos proveen un refinamiento al pronóstico inicial. Si bien estos modelos poseen un buen comportamiento, su limitación está relacionada con la obtención de los datos del lecho marino. Usualmente la obtención de estos datos es costosa y no se encuentra disponible en todos los lugares.

## 1.3 Trabajos relacionados

Existen algunos trabajos previos que enfocan el tema de predicción de oleaje. Los trabajos de [4] y [5] se caracterizan por enfocarse en la predicción de la altura de la ola en la zona cercana a la costa, donde se realiza la práctica de surf, a partir de información obtenida en alta mar. Para lograr esto definen un modelo de regresión en donde las variables predictivas están dadas por los datos en alta mar y la variable de respuesta es la altura de la ola en la playa. [5] utiliza como variables predictivas los datos de alta mar recolectados por una boya marina, mientras que [4] utiliza las previsiones oceánicas brindadas por la resolución del modelo numérico WAVEWATCH III (Ver Capítulo 3 – Datos para predicción del oleaje). En ambos trabajos la variable de respuesta está representada por las observaciones visuales recolectadas por un observador experto el cual registra el tamaño de ola promedio en un determinado momento del día.

En los trabajos [6] y [7] la predicción de la altura de la ola se realiza en el mismo lugar donde se recolectan los datos. Estos utilizando el estado del mar de las últimas horas en un lugar determinado, infieren cual será la altura de la ola en ese mismo lugar con horas de anticipación. Para lograr esto se plantea un modelo de regresión donde las variables predictivas están dadas por las lecturas de una boya marina y la variable de respuesta está dada por la altura de la ola que se presentará en ese mismo lugar con N horas de anticipación.

Tanto en [4] y [6] como en [7] se desarrollaron modelos de regresión en base a redes neuronales, todas multicapa de propagación hacia adelante, con una función de transferencia del tipo sigmoidal. En contraste en [5] utilizaron un modelo de regresión polinomial.

En [7] se experimenta, obteniendo resultados positivos, con una técnica especial en la cual se conectan dos redes neuronales. Donde la salida de la primera red es la entrada a la segunda. Según los autores esto permite a la primera red concentrarse en aprender las interacciones complicadas, mientras que la segunda red realiza un refinamiento a las predicciones de la primera.

Diversas comparaciones entre los modelos desarrollados y modelos de predicción ya existentes son realizadas en los distintos trabajos. En [7] se compara el modelo de red neuronal desarrollado con un modelo estadístico auto regresivo obteniéndose mejores resultados con el primero. [4] realiza una comparación entre el modelo de red neuronal y el modelo SWAN (Booij et al., 1999), el cual es un modelo numérico para predicción de olas en la zona costera, obteniendo mejores resultados con el primero. También se realizan comparaciones entre un modelo de regresión lineal y uno no lineal, obteniendo mejores resultados con el modelo no lineal.

Todos los trabajos analizados presentan un desempeño satisfactorio en la predicción de olas, de esta manera dejando en evidencia que los métodos de regresión para la predicción de olas son una alternativa viable a los métodos de modelado numérico, y que además los mismos son útiles para refinar los pronósticos dados por los modelos de oleaje de alta mar en la cercanía de la costa.

## 1.4 Trabajo realizado

Nuestro estudio, alineado con los objetivos de los trabajos de [4] y [5], consistió en utilizar modelos de aprendizaje de máquina para aprender de casos pasados como un determinado pronóstico de alta mar se vio reflejado en las cercanías de la costa. De esta manera ante un nuevo pronóstico oceánico se logró predecir cuál fue el efecto del mismo en las cercanías de la costa. Nos centramos en la predicción de la altura de la ola en la cercanía de la costa.

Los métodos utilizados fueron:

* Regresión lineal
* Redes neuronales
* Maquinas de soporte vectorial
* Arboles modelos y de regresión

Evaluamos el desempeño de cada modelo sometiéndolos a un entrenamiento con datos reales (tanto de lecturas de altamar como de observaciones visuales costeras) y luego evaluando las predicciones con un conjunto datos diferentes a los utilizados para entrenar cada clasificador (datos nuevos que nunca antes han usado para entrenarse). Finalmente implementamos un sistema de predicción de olas en la cercanía de la costa en cualquier lugar del planeta utilizando como algoritmo de regresión aquel que presentó los mejores resultados en cada una de las diferentes pruebas a las que se sometieron en la etapa de experimentación (Ver capitulo 4 - Evaluaciones).

La evaluación experimental se realizó utilizando datos históricos de las condiciones de alta mar y de observaciones visuales correspondientes a la isla de Oahu, Hawái. Los datos de alta mar se obtuvieron de un archivo histórico de las salidas del modelo numérico WAVEWATCH III, provistos por la agencia americana NOAA (National Ocenographic and Atmospheric Administration), comprendiendo la altura, dirección y periodo de ola, dirección y velocidades de los vientos. Los datos de las observaciones visuales usados solo brindan la altura de la ola vista desde la costa. Las observaciones han sido relevadas en su mayoría por el servicio de guardavidas de la isla de Oahu y se encuentran disponibles para cinco playas diferentes: Sunset, Diamond Head, Makapu, Makaha y Ala Moana ubicadas alrededor de la isla. Tanto de la información de alta mar como de las observaciones visuales se cuenta con más de 6 años de registros con una periodicidad diaria.

Dado que las lecturas de WAVEWATCH III se actualizan cada 3 horas, tanto durante el día como la noche, y debido a que las observaciones visuales se han realizado en horarios en que se cuenta con luz solar, el primer filtro aplicado a la información que ofrece el modelo WAVEWATCH III fue eliminar todas las lecturas que no correspondían con un horario de luz natural en Oahu. Por otra parte, las observaciones visuales nos indicaban la altura de la ola más grande que arribo a la costa en el día, por esto el segundo filtro aplicado a las lecturas de WAVEWATCH III fue dejar únicamente la lectura diaria con la mayor altura de ola. De esta forma acoplamos el estado de altamar brindado por el modelo WAVEWATCH III y la observación de la altura de la ola registrada en la costa.

Para llevar a cabo el entrenamiento de los clasificadores utilizamos la herramienta WEKA. Esta herramienta provee los algoritmos de regresión (entre otros), utilizados en nuestro estudio durante la etapa de experimentación. WEKA recibe como entrada un conjunto de instancias descriptas a través de una serie de atributos. Cada instancia se compone de un conjunto de variables predictoras y uno o más variables a predecir. En nuestro estudio las variables predictoras son las lecturas del modelo WAVEWATCH III (altura de ola, periodo, velocidad del viento, dirección de ola) y el atributo de clase o variable a predecir fue la observación visual en la costa. Entonces dado un conjunto de instancias como entrada, y un algoritmo de regresión, WEKA logra entrenar un clasificador que es capaz de recibir únicamente instancias con las variables predictoras y devolver por cada una el valor del atributo de clase a predecir (altura de la ola en la costa).

La clave del estudio fue encontrar la forma de armar los datos de prueba en base a la información de la cual disponíamos y al mismo tiempo buscar y configurar el algoritmo de aprendizaje de máquina que mejores resultados entregaba al entrenarse con dichas instancias (es decir el que lograba predecir las olas en la ubicación del observador minimizando el margen de error y maximizando la correlación entre los datos de las lecturas de alta mar y la predicha por el clasificador).

En este trabajo se experimentó con numerosos conjuntos de datos obtenidos a partir del procesamiento de las observaciones reales. Para cada modelo de instancia se armaron diferentes conjuntos de entrenamiento, los cuales fueron utilizados como entrada para los distintos algoritmos de regresión. La combinación de variables predictoras (atributos) que mejores resultados mostraron en los experimentos fue la siguiente:

Entre los atributos considerados se encuentran:

* altura significante de ola actual (variable predictora WAVEWATCH III)
* dirección promedio actual (variable predictora WAVEWATCH III)
* período promedio actual (variable predictora WAVEWATCH III)
* altura significante de ola 1 día antes (variable predictora WAVEWATCH III)
* dirección promedio 1 día antes (variable predictora WAVEWATCH III)
* período promedio 1 día antes (variable predictora WAVEWATCH III)
* altura significante de ola 2 días antes (variable predictora WAVEWATCH III)
* dirección promedio 2 días antes (variable predictora WAVEWATCH III)
* periodo promedio 2 días antes (variable predictora WAVEWATCH III)
* observación visual (atributo de clase, histórico de observaciones de Oahu)

Este modelo utilizado se basó en armar cada instancia para un determinado GridPoint en una determinada fecha con las lecturas del WAVEWATCH III correspondientes a la ola más grande registrada ese día en un horario con luz solar, junto con la observación visual de altura de ola de ese mismo día, pero además con las lecturas de WAVEWATCH III correspondientes a la ola más grande registrada un día antes y dos días antes (ambas durante un horario con luz solar). De esta manera armamos instancias con nueve variables predictoras de WAVEWATCH III y un atributo de clase que representa la observación visual.

Por otra parte el algoritmo de regresión elegido en base al rendimiento del mismo fue Maquinas de Soporte Vectorial, cuyos resultados al ejecutarlos con las instancias descriptas anteriormente pueden observarse en el Capitulo 4 – Evaluaciones.

Con base en este clasificador entrenado desarrollamos un sistema web denominado “Surf-Forecaster” el cual es capaz, si se le suministra la información necesaria (histórico de observaciones visuales de cualquier lugar deseado), de dar un pronóstico con hasta una semana de anticipación en incrementos de tres horas, especificando la altura de las olas exactamente en la ubicación que al usuario le interesa, considerando que las variables restantes (dirección de ola, periodo de ola y dirección y velocidad del viento) no sufren cambios significativos entre el GridPoint evaluado y la costa dado que los cambios de las profundidades del mar y la geografía de la zona no las afecta, ofreciendo una alternativa distinta a los pronósticos existentes los cuales brindan esta información pero en zonas muy alejadas a la costa, y por lo general reportan el mismo pronostico para toda una región o localidad, lo cual pierden precisión en sitios con una geografía muy irregular o donde el tipo de fondo y profundidad del mar (entre otros aspectos) cambian vertiginosamente entre lugares separados por tan solo unos pocos kilómetros.

El sistema desarrollado puede ser de gran utilidad tanto para deportistas que diariamente recorren una localidad en busca de las mejores condiciones para su deporte como así también por ejemplo para tomar medidas de prevención anticipadas en lugares donde olas de gran tamaño pueden llegar a alcanzar a la población de esa zona.

Surf-Forecaster, es una aplicación interactiva que ofrece la posibilidad de obtener pronósticos detallados del oleaje, en cualquier lugar del planeta. Basta con crear una ola (Ver sección 5.3.3 - Nueva ola) en cualquier lugar del mundo para comenzar a recibir pronósticos detallados con 180hs de anticipación para la ola creada.

Adicionalmente el sistema cuenta con una sección de comparaciones entre diferentes olas, la cual brinda una herramienta muy práctica para ayudar al usuario a reconocer fácilmente en qué lugares romperán las mejores olas del día con una semana de anticipación, evitándole la necesidad, por ejemplo, de tener que recorrer toda la costa de su ciudad, diariamente, en busca de tales olas.

Surf-Forecaster ofrece además la posibilidad de registrase en el sistema (ver sección 5.2 – Tipos de usuario). Al ser un usuario registrado el sistema le permitirá dar de alta sus olas preferidas, generar comparaciones entre ellas y persistirlas, para no tener que volver a crearlas cada vez que quiere comparar las mismas olas.

## 1.5 Estructura del trabajo

En esta sección se detalla la estructura general del presente trabajo, brindando una breve descripción de los temas que se abordan en cada capítulo.

**Capitulo 2 – Estado del arte**

En este capítulo se describen los conceptos básicos del aprendizaje supervisado. Se desarrollan los métodos de regresión analizados para predicción de olas y finalmente se describen los trabajos relacionados existentes.

**Capitulo 3 – Datos para predicción del oleaje**

El objetivo de este capítulo es dar al lector una breve explicación de las olas y sus conceptos generales. Se describen además las observaciones visuales costeras usadas para el ejemplo del caso de estudio (Instanciado en la isla de Oahu - Hawái), el modelo WAVEWATCH III como fuente de pronósticos en altamar y se expone como se combinaron los datos y se pre-procesaron antes de su utilización.

**Capitulo 4 – Evaluaciones**

En el presente capítulo se detallan las pruebas realizadas para encontrar los mejores clasificadores que permitan predecir el tamaño que tendrá la ola al momento de romper en la costa, a partir de la información de alta mar brindada por el modelo de olas WAVEWATCH III. En especial se analizan los resultados obtenidos de experimentar con diferentes conjuntos de entrenamiento así como también con diferentes algoritmos de regresión.

**Capitulo 5 – Desarrollo de la aplicación**

En este capítulo se describe el sistema implementado, se exponen sus diferencias contra sistemas similares existentes y se explica cómo logra predecir Surf-Forecaster. Se describen las diferentes secciones que brinda la aplicación y como utilizarla.

**Capitulo 6 – Conclusiones**

En este capítulo se presentan las conclusiones obtenidas a partir de la utilización de la técnica propuesta, sus ventajas y limitaciones. Además, se describen algunas posibles mejoras en trabajos futuros.

**Capitulo 2 - Estado del arte**

**2.1 Aprendizaje supervisado**

El aprendizaje inductivo es aquel que a partir de la experiencia y observación permite al hombre comprender los fenómenos que lo rodean. Para realizar este tipo de aprendizaje es necesario recolectar las observaciones para luego ser analizadas y así deducir una hipótesis la cual explique un fenómeno en particular. Los medios para recolectar la información suelen ser la memoria humana, hojas de papel, archivos planos, bases de datos ([8]).

Cuando la cantidad de datos recolectados es grande, el análisis de los mismos para comprender un fenómeno determinado es una tarea dificultosa. Hoy en día los avances en las tecnologías, brindando una alta capacidad de almacenamiento y procesamiento, han hecho de la computadora un elemento muy importante en el proceso de descubrimiento. Esto implica no usar la computadora para que la misma nos indique gráficos o estadísticas, sino utilizarla para que ella misma aprenda de los datos.

El aprendizaje de maquina es una disciplina de la inteligencia artificial dedicada al desarrollo de algoritmos que permiten a la computadora aprender a partir de la experiencia. Ejemplo de tareas que abarca el aprendizaje de maquina pueden ser: descubrir que tratamientos se adaptaran mejor a futuros pacientes a partir del archivo de historias clínicas de un hospital; como hacer que un motor de búsqueda se adapte automáticamente a las preferencias de sus usuarios o como hacer que un robot móvil aprenda a navegar a partir de la experiencia ([18]).

El aprendizaje de maquina supervisado está dedicado al desarrollo de algoritmos que aprenden a partir de ejemplos. Un ejemplo está representado por una instancia y una clase a la que esa instancia pertenece. Una instancia representa un conjunto de observaciones sobre un hecho determinado. Una clase representa la clasificación que se le da al hecho. A partir de un conjunto de ejemplos y un algoritmo de aprendizaje supervisado podemos entrenar un clasificador para que el mismo pueda predecir cuál será la clase de una instancia que desconocemos.

En la figura 2-1 se muestra un conjunto de ejemplos los cuales son utilizados como entrada a un algoritmo de aprendizaje de máquina para generar un clasificador. Las instancias están compuestas por un conjunto de atributos correspondientes a las observaciones del ambiente en una cancha de golf. El atributo de clase representa si el día es apropiado o no para la práctica de golf en la cancha en cuestión. El usuario incorpora conocimiento acerca del problema seleccionando los datos a usar como entrenamiento así como también en la selección y parametrización de los algoritmos de aprendizaje de máquina.



Fig. 2.1: Proceso de clasificación. (Adaptado de [8])

La selección y preparación de los datos a utilizar tienen mucha importancia dado que el rendimiento del clasificador se verá sumamente condicionado por los mismos. Los datos suelen tener impurezas, como ruidos o valores faltantes, haciendo necesario definir una estrategia para tratar los mismos. Además, los problemas de aprendizaje de maquina suelen tener instancias con un gran número de atributos y no todos suelen ser relevantes para el problema en cuestión. La cantidad de estos afectan negativamente el desempeño, en cuanto a tiempo y precisión, de los algoritmos de clasificación. Para lidiar con este problema suelen ser utilizadas las técnicas de Selección de Atributos y Extracción de Atributos.

El resultado del proceso de aprendizaje es un clasificador, el cual representa un concepto aprendido a partir de los ejemplos. Cuando el formato de este concepto es comprensible por el humano se dice que el sistema de aprendizaje es orientado al conocimiento, de lo contrario el sistema se denomina de caja negra ([16]). Los sistemas orientados al conocimiento tienen la ventaja de poder ser utilizados no solo para la predicción de la clase de las instancias desconocidas sino que también permiten ser utilizados para comprender el fenómeno subyacente a los ejemplos. Estructuras con forma de árbol o de reglas de decisión son algunas de las usadas para describir los conceptos adquiridos.

El clasificador obtenido es resultado de un proceso inductivo y como tal su efectividad no está garantizada. Además de evaluar el clasificador con el sentido crítico del usuario es necesario establecer métricas claras del desempeño del mismo. Para esto se prueba el clasificador con instancias cuya clase se conoce previamente y se comparan la clase verdadera y la predicha.

En clasificación una métrica común consiste en contar la cantidad de instancias bien clasificadas y las mal clasificadas para luego obtener la taza de error. Cuando el atributo de clase es numérico otras medidas de error son tomadas. Medidas utilizadas frecuentemente son el error absoluto medio, el error cuadrático medio y el coeficiente de correlación ([1]).

Es importante que la evaluación sea realizada con un conjunto de datos independiente del usado para entrenamiento, de lo contrario la evaluación no sería representativa del desempeño del clasificador ante datos nuevos. El ideal es contar con tres conjuntos de datos: uno para entrenamiento, uno para validación y otro para pruebas. El conjunto de entrenamiento es utilizado para obtener el clasificador. El conjunto de validación para ajustar los parámetros del clasificador. El conjunto de prueba para obtener las medidas de performance finales del clasificador. En situaciones donde el número de ejemplos no resulta abundante para separar en estos tres conjuntos las técnicas de re muestreo resultan útiles. Ejemplo de ellas son: dejar-uno-afuera, validación cruzada de 10 particiones estratificada.

Cuando la clase que se desea aprender es discreta los problemas de aprendizaje supervisado se conocen como clasificación, mientras que cuando la clase es continua se los denomina problemas de regresión. Estos últimos utilizaremos en este trabajo y de describen a continuación.

**2.2 Métodos de regresión**

El análisis de regresión es un conjunto de técnicas de modelado y análisis numérico las cuales consisten en determinar el valor de un conjunto de variables de respuesta a partir de un conjunto de variables predictoras. Estas técnicas son utilizadas para realizar pronósticos, inferencia, prueba de hipótesis y evaluación de causalidad entre otras cosas.

**2.2.1 Regresión Lineal Simple**

El análisis de regresión de línea recta implica una variable de respuesta Y y una sola variable predictora X. Esta es la forma de regresión más simple que podemos encontrar, donde se modela Y como una función lineal de X. ( [14]) Esto es,

*y* = *b* + a*x*; (Ec. 2.1)

Donde la varianza de y es supuesta constante, b y a son los coeficientes de regresión que especifican la intercepción con el eje Y, y la pendiente de la recta respectivamente.

Estos coeficientes pueden ser resueltos mediante el método de mínimos cuadrados, el cual estima la recta de mejor ajuste como aquella que minimiza el error entre el dato verdadero y la estimación dada por la línea recta. Sea |D| el conjunto de entrenamiento que contiene puntos de datos de la forma (, ), (, ),…, (, ), podemos definir el error en función de los parámetros como:

(Ec. 2.2)

Donde , es el valor verdadero e el valor estimado por la función de regresión.

Queremos encontrar los valores de a y b que minimicen la función de error. Diferenciando esta función con respecto a cada parámetro hayamos las siguientes ecuaciones las cuales nos proporcionan los parámetros de la recta de mejor ajuste:

(Ec 2.3)

(Ec 2.4)

Donde es el promedio de x1, x2, ..., x|D|, y el correspondiente a y1, y2, ..., |D|.

***Ejemplo 2.1:***

Regresión lineal usando el método de mínimos cuadrados. La tabla 2-1 muestra los pares de datos: años de experiencia laboral de un graduado universitario (X) y el salario correspondiente (Y).

|  |  |
| --- | --- |
| **Años de experiencia (X)** | **Salario (Y) (miles)** |
| 3  8  9  13  3  6  11  21  1  16 | 30  57  64  72  36  43  59ª  90  20  83 |

Tabla 2.1: Relación años de experiencia, salario percibido ([14]).

Fig. 2.2: Gráfico de los datos de la tabla 2.1 junto con la recta de regresión calculada utilizando el método de regresión lineal simple

El grafico de la figura 2.2 sugiere una relación lineal entre las dos variables X e Y. De esta manera podemos modelar la relación entre el salario y la cantidad de años de experiencia mediante la ecuación de regresión lineal 2.1. Utilizando los datos de la tabla 1 junto con las ecuaciones (2.3) y (2.4) obtenemos el valor de los coeficientes:

a = 23,6

b = 3,5:

Reemplazando en 2.1 obtenemos la función de la recta de regresión:

Y = 23,6 + 3,5X. (Ec. 2.5)

Utilizando esta función podemos predecir el valor de salario para un nivel de experiencia de la cual no tenemos información. Por ejemplo podemos decir que se estima que un egresado universitario con X = 10 años de experiencia, posee un salario de: Y = 23,6 + 3,5 \* 10.= $58600.

La regresión lineal múltiple es una extensión de la regresión lineal simple de manera de poder incorporar más de una variable predictora. La misma permite modelar la variable de respuesta como una función lineal de N variables predictoras, formando una tupla, ( X = (x1, x2, : : : , xn)). Nuestro conjunto de entrenamiento D contiene datos de la forma (X1, y1), (X2, y2), : : : , (XjDj, yjDj), donde las Xi son tuplas de entrenamiento N dimensionales, con etiquetas de clases asociadas Yi. Un ejemplo de un modelo de regresión múltiple basado en dos variables predictoras es:

Y = w0 + w1x1+w2x2, (Ec. 2.6)

El método de cuadrados mínimos es extendido para encontrar los parámetros w0, w1, y w2 que minimicen la función de error asociada a este modelo. Las ecuaciones de regresión múltiple son tediosas para resolver a mano y este tipo de problemas son comúnmente resueltos usando paquetes estadísticos o de minería de datos.

La regresión lineal es un método simple pero poderoso para ser utilizado en la predicción numérica, el mismo ha sido usado ampliamente en aplicaciones estadísticas durante décadas. La desventaja que presenta este método es la linealidad. Cuando los datos exhiben una dependencia no lineal, la mejor recta de ajuste será encontrada. Esta recta no se ajustará demasiado bien a este tipo de datos. A pesar de esto los modelos lineales son interesantes ya que sirven como base para el desarrollo de métodos de aprendizaje más complejos ([1]).

**2.2.2 Redes neurales para regresión**

Una Red Neuronal Artificial (Red Neuronal o ANN), es un modelo computacional inspirado en las redes neuronales biológicas. La red está conformada de elementos de procesamiento (neuronas), conexiones entre los elementos de procesamiento y coeficientes (pesos) asociados a cada conexión. Todos estos elementos forman la estructura neuronal.

Las redes neuronales, en su mayoría, tienen la particularidad de ser sistemas adaptativos, es decir que adaptan su comportamiento de acuerdo al ambiente en donde se encuentran. Así la red es capaz de modificar su estructura, en la fase de aprendizaje, de acuerdo a la información que se le presenta a la misma.

En términos prácticos las redes neuronales son técnicas de modelamiento no lineales capaz de modelar funciones complejas. Estas pueden ser aplicadas a problemas de regresión, clasificación o control en un amplio espectro de campos como finanzas, neurociencia, medicina, ingeniería y física.

El bloque fundamental para la construcción de la red neuronal artificial es el modelo matemático de una neurona como se muestra en la figura 2.3. Los tres componentes básicos de una neurona artificial son:

* Las conexiones que proveen pesos Wj a los valores de entrada Xn.
* Una función de transferencia para computar el valor de entrada a la función de activación.
* Una función de activación g que mapea v a g(v), el valor de salida de la neurona.



Fig. 2.3: Modelo de una neurona artificial.

Mientras existen numerosas arquitecturas de redes neuronales, las aplicaciones con mejores resultados en el área de minería de datos han sido las redes multicapa de propagación hacia adelante (Multilayer Feedfoward) (MIT, 2009). Estas son redes en las cuales existe una capa de nodos que simplemente aceptan los valores de entrada y capas sucesivas que son neuronas como las de la figura 2.3. Las salidas de las neuronas en una capa son entrada a las neuronas en la capa sucesiva. La última capa es denominada capa de salida. Las capas entre la entrada y la salida son denominadas capas ocultas, ya que no interactúan con el medio externo. La figura 2.4 muestra un diagrama para esta arquitectura.



Fig. 2.4: Arquitectura de red neuronal Multilayer Feedforward.

Una de las características más importantes de las redes neuronales es su habilidad para aprender. Si consideramos una red neuronal como una función de mapeo F: X→Y, siendo X un vector de entrada a la red e Y un vector de salida de la misma. Dada una tarea específica para resolver y una clase de función F, aprender significa usar un conjunto de observaciones para encontrar un f\* F, que resuelva la tarea específica de manera óptima. La solución óptima será aquella que haga la función de costo mínima. Esta función representa el error del clasificador entre el valor de la predicción y el valor verdadero.



A pesar que la función de costo puede ser elegida de manera arbitraria, la elección de la misma suele ser realizada basada en las propiedades de la misma (convexidad) y también en las particularidades del problema que se intenta resolver. Finalmente la elección de la función de costo dependerá del tipo de tarea que intentemos resolver. La función de costo más utilizada en la práctica resulta ser la función de mínimos cuadrados. Esta resulta ser la misma función utilizada en la regresión lineal simple (Ver ecuación 2.2).

El problema de entrenamiento puede dividirse en dos: aprender la estructura de la red y aprender los pesos de las conexiones ([1]).

Existen numerosos algoritmos de entrenamiento, que resuelven de manera simple los valores de los pesos, dada una estructura de red fija. Ejemplo de ellos son: Propagación hacia atrás, Propagación rápida (ver [22], [15] y [13]), Gradiente Conjugado descendiente (ver [9]). Si bien existen algoritmos para encontrar una estructura de red adecuada este aspecto del problema suele ser resuelto a través de la experimentación.

Los algoritmos de estimación de pesos son, en su mayoría, aplicaciones de la teoría de optimización y de estimación estadística. Estos utilizan alguna variante de la técnica de optimización [Gradiente](http://en.wikipedia.org/wiki/Gradient_descent) descendente. El algoritmo más conocido es denominado “Propagación hacia atrás”

La técnica de “Propagación hacia atrás” se compone de un ciclo de dos fases (MIT, 2009): una fase hacia adelante, donde un dato de entrenamiento se introduce en la red y se calculan las salidas de todos los nodos hasta llegar al nodo final que produce el resultado de predicción. Una fase de retroceso en la cual se van actualizando los pesos de las conexiones desde los nodos de la capa de salida hasta la capa de entrada.

La técnica de propagación hacia atrás utiliza Gradiente Descendente, una técnica de optimización iterativa que usa la información de la derivada de primer orden de la función de costo para ajustar los pesos de la red. A partir del valor de las derivadas, las multiplica por una pequeña constante llamada **tasa de aprendizaje** y luego sustrae el resultado al valor actual del peso. Esto es repetido en cada ciclo hasta que el cambio en el valor del peso se torna muy pequeño, de esta manera hemos encontrado la configuración de los pesos que logran un mínimo de la función de costo elegida.

La **tasa de aprendizaje** determina el incremento en dirección al mínimo y por lo tanto que tan rápido la búsqueda converge. Si esta tasa es muy grande y la función tiene múltiples mínimos, la búsqueda puede pasar por alto algún mínimo, o puede oscilar fuertemente. Si la tasa es pequeña el progreso hacia un mínimo puede volverse demasiado lento. Cabe destacar que el método de Gradiente Descendente solo puede encontrar un mínimo local. Si la función de costo tiene varios mínimos puede ser que no se encuentre el mínimo óptimo. Para aliviar este problema suelen realizarse múltiples corridas inicializando los valores de pesos en forma aleatoria.

Como en cualquier otra técnica de aprendizaje de maquina en las redes neuronales podemos sufrir el problema de sobre entrenamiento (del inglés overfitting), es decir que la red puede reflejar una buen desempeño con los datos de entrenamiento, pero no así con datos nunca vistos.

**Detención temprana** es una modificación a la técnica de Gradiente Descendente la cual consiste en tener un conjunto de datos separado para verificar el desempeño de la red en cada iteración del ciclo de propagación hacia atrás. Cuando el desempeño medido con este conjunto de datos empieza a decaer, indicando sobre entrenamiento, el algoritmo es terminado.

Una pasada por todos los datos de entrenamiento se denomina una **Época**. La mayoría de las redes multicapa de propagación hacia adelante requieren de varias épocas antes de que los errores sean razonablemente pequeños.

El **momento** es una solución para minimizar el número de épocas necesarias para encontrar un mínimo aceptable. La misma consiste en agregar al peso que se está actualizando una proporción del incremento agregado en la iteración previa. Esto genera que el proceso de búsqueda sea más suave haciendo los cambios en dirección menos abruptos y favoreciendo una convergencia más rápida. Valores altos en el parámetro de momento forzaran a que los ajustes sucesivos sean en direcciones similares. Otra idea es variar el parámetro de **tasa de aprendizaje** para que este comience con un valor alto e ir decrementándolo a medida que se avanza de época ([1]).

Las redes neuronales pueden verse como una especie de sistema de procesamiento no lineal capaz de resolver un amplio espectro de problemas. Las redes neuronales son útiles cuando existen datos en abundancia pero se carece de una base teórica completa, es decir, no hay un modelo causal o una representación matemática. Los datos disponibles suelen ser no lineales, no estacionarios, o caóticos haciéndolos difíciles de modelar. Las redes neuronales no suponen ningún conocimiento previo acerca del espacio del problema, tampoco necesitan conocimientos previos en cuanto a la distribución estadística de los datos.

Las tareas en la cuales las redes neuronales son aplicadas se encuentran dentro de las siguientes categorías: [Aproximación de funciones](http://en.wikipedia.org/wiki/Function_approximation), o análisis de regresión, incluyendo predicción de series de tiempo y modelamiento. [Clasificación](http://en.wikipedia.org/wiki/Statistical_classification), incluyendo reconocimiento de patrones y reconocimiento de secuencias Procesamiento de datos, incluyendo filtrado, clusterización y compresión.

Las áreas de aplicación de las redes neuronales incluyen: Sistemas de control (control de vehículos), juegos (backgammon, ajedrez), reconocimiento de patrones (sistemas de radares, identificación de caras, reconocimiento de objetos), reconocimiento de secuencias (gestos, habla, escritura), diagnósticos médicos, aplicaciones financieras, descubrimiento de conocimiento en bases de datos, visualización y filtrado de email spam.

**2.2.3 Árboles modelos y de regresión**

La regresión lineal es un modelo global, donde existe una única ecuación predictiva que se mantiene para todo el espacio de datos. Cuando los datos tienen muchas características los cuales interactúan de una forma complicada, en formas no lineales, encontrar un único modelo global puede ser muy difícil ([26]). Inclusive una vez encontrado este modelo, el mismo suele ser confuso.

Una alternativa al enfoque no lineal es la de subdividir o particionar el espacio en regiones más pequeñas donde las interacciones son más accesibles. Luego estas particiones se vuelven a sub dividir y así sucesivamente. Finalmente obtenemos porciones del espacio en donde podemos utilizar modelos sencillos para encajar los datos. De esta manera el modelo global tiene dos partes: una consiste en la partición recursiva del espacio, la otra en aplicar un modelo simple a cada celda de la partición. La alternativa mencionada no es más que la aplicación de la estrategia “Divide y vencerás”. Este enfoque conlleva a adoptar un estilo de representación de los datos en forma de árbol. En cada nodo de un árbol se evalúa un atributo en particular. Generalmente se compara el nodo con un atributo constante. Los nodos hojas dan una clasificación que se aplica a todas las instancias que alcanzan la hoja. Para clasificar una instancia desconocida, la misma es encaminada desde la raíz del árbol hacia abajo de acuerdo a los valores de los atributos que se evalúan en cada nodo y cuando una hoja es alcanzada la instancia es clasificada de acuerdo a la clase asignada para esa hoja.

Existe dos clases de árboles utilizados para predicción numérica, los árboles de regresión ([3] citado en [1]) y los árboles modelo ([23] citado en [1]). La única diferencia entre ambos es que el primero almacena en sus nodos hojas el promedio de los datos que alcanzaron ese nodo, mientras que el segundo almacena un modelo de regresión lineal el cual es usado para predecir el valor de las instancias que alcanzan ese nodo ([1]). En lo que sigue describiremos los árboles modelos, ya que los árboles de regresión son un caso especial.

***Construcción del árbol modelo:***

La construcción del árbol modelo es un proceso recursivo. Comenzando del nodo raíz, se selecciona el atributo que mejor separa los datos de entrenamiento. Evaluando este atributo para cada instancia del conjunto de entrenamiento el mismo quedara separado en diferentes subconjuntos. Existirán tantos subconjuntos como posibles alternativas presente el nodo de decisión. Este proceso es repetido para cada subconjunto de los datos de entrenamiento hasta que todas las instancias que alcanzan un nodo tienen la misma clasificación.

Para determinar que atributo es el que mejor separa la porción T de los datos de entrenamiento que alcanzan un nodo en particular se utiliza el criterio de particionamiento. El mismo está basado en utilizar la desviación estándar de los valores de clase de T como una medida del error en ese nodo. El atributo que maximiza la reducción del error esperado es elegido para particionar los datos que llegan al nodo. La reducción del error esperado esta dado por la siguiente fórmula ([1]).

(Ec.2.5)

Donde T1, T2,. . . son los conjuntos que resultan de separar el nodo de acuerdo al atributo elegido. El proceso de particionamiento termina cuando el valor de clase de las instancias que alcanzan un nodo varían muy poco, es decir cuando su desviación estándar es solo una pequeña fracción (Ej.: %5) de la desviación estándar del conjunto de instancias original. El particionamiento también termina cuando quedan unas pocas instancias en un nodo, por ejemplo: 4 instancias. La experimentación indica que los resultados obtenidos no son muy sensibles al valor de estos parámetros.

Para predecir el valor de una instancia de prueba el árbol es atravesado hasta las hojas usando los valores de los atributos para decidir qué camino tomar en cada nodo. La hoja tendrá un modelo lineal el cual será utilizado para obtener el valor de la predicción. En vez de utilizar este valor directamente, resulta ser beneficioso realizar un proceso de suavizado para compensar las discontinuidades que inevitablemente ocurren entre los modelos adyacentes en las hojas del árbol. Este proceso es llevado a cabo implementando un modelo lineal en cada nodo interno del nodo, además del de las hojas. Luego una vez obtenida la predicción dada por el modelo en la hoja, este valor es filtrado durante el camino hacia el nodo raíz, suavizándolo en cada nodo que es atravesado mediante la combinación con el valor predicho por el modelo de cada nodo. Un modelo apropiado para calcular este suavizado esta dado por:

(Ec. 2.6)

*p* es la predicción pasada al nodo superior en el árbol, *p* es la predicción pasada al nodo actual proveniente del nodo inferior, *q es* el valor predicho por el modelo en el nodo actual, *n* es el número de instancias de entrenamiento que alcanzaron el nodo inferior, y *k* es una constante de suavizamiento. La experimentación muestra que realizando este proceso de suavización se mejora substancialmente la precisión de las predicciones.

*Poda:*

A pesar que los árboles construidos mediante el enfoque de “divide y vencerás” tienen un buen rendimiento sobre los datos de entrenamiento, los mismos suelen tener problemas de sobre entrenamiento y no pueden generalizar correctamente con conjuntos de pruebas independientes. Una solución a este problema es la realización de podas sobre el árbol.

Existen dos maneras para realizar la poda: **pre-poda** y **pos-poda**. En pre-poda cuando se va construyendo el árbol y se encuentra una estructura que es lo suficientemente compleja, se detiene la construcción en esa rama. En pos-poda primero se construye todo el árbol y luego las descripciones complejas son extraídas ([1]).

Pre-poda implica decidir, durante la construcción del árbol, cuando parar de desarrollar los sub-árboles- esta particularidad resulta interesante ya que, a diferencia de pos-poda, no se hace trabajo de mas desarrollando sub árboles que luego podrían ser desechados. A pesar de esto pos-poda ofrece ciertas ventajas. Por ejemplo, existen situaciones en la que dos atributos considerados individualmente no tienen ningún aporte significativo, mientras que los mismos considerados en conjunto resultan ser muy informativos. De esta manera para asegurarnos de poseer la mayor información posible es necesario construir el árbol completo para luego desechar las partes que provocan sobre entrenamiento. La mayoría de los algoritmos de construcción de árboles utilizan pos-poda. Actualmente es una pregunta abierta si las estrategias de pre-poda pueden ser desarrolladas para alcanzar el mismo rendimiento que las estrategias de pos-poda.



Fig. 2.5: Árbol modelo ([1]).

En la figura 2.5, se muestra un ejemplo de un árbol modelo: En cada nodo se evalúa un atributo diferente y en las hojas se encuentra identificado el modelo lineal a utilizar.

**2.2.4 Maquinas de Soporte Vectorial para regresión**

Las maquinas de soporte vectorial fueron desarrolladas como se conocen hoy en día en los laboratorios AT&T por Vapnik y colaboradores ([11], [13], [24], [27], [12], citados en [25]). Debido a este contexto industrial la investigación fue orientada a aplicaciones del mundo real, específicamente al reconocimiento de caracteres. En un corto periodo de tiempo estos clasificadores se convirtieron en competidores de las mejores técnicas existentes del momento. Así también las maquinas de soporte vectorial utilizadas para clasificación fueron extendidas para soportar problemas de regresión, obteniéndose también muy buenos resultados ([21], [12], [27] citados en [25]). Actualmente los algoritmos de maquinas de soporte vectorial forman parte de cualquier herramienta estándar para minería de datos. A continuación explicaremos los conceptos detrás de las maquinas de soporte de vectores utilizadas para regresión.

Como con la regresión lineal, la idea básica es encontrar una función que aproxime los puntos de entrenamiento minimizando el error en la predicción. La diferencia crucial es que todas las desviaciones hasta un parámetro ɛ dado son descartadas ([1]).

Un parámetro ɛ especificado por el usuario define un tubo alrededor de la función de regresión en los cuales los errores son ignorados: para soporte de vectores lineal el tubo es un cilindro. Si todos los puntos de entrenamiento caben dentro de un tubo de 2ɛ, el algoritmo obtiene una función en el medio del tubo más horizontal que los encierra. En este caso el error percibido es cero. La figura 2.6 muestra un problema de regresión con un atributo, una clase numérica, y ocho instancias. En este caso el valor de ɛ fue configurado en 1, siendo el ancho del tubo alrededor de la función de regresión igual a 2. La figura 2.6 b muestra la salida del proceso de aprendizaje con el valor de ɛ configurado en 2. Como se puede apreciar un tubo más ancho hace posible aprender una función más horizontal.

El valor de ɛ controla que tan cerca la función encajaralos datos. Un valor demasiado grande producirá un predictor sin sentido – en el caso extremo, cuando 2ɛ excede el rango de valores de la clase de los datos de entrenamiento, la línea de regresión es horizontal y el algoritmo solo predice el valor promediode clase. Por otro lado, para valores pequeños de ɛ puede no haber un tubo que encierre todos los datos. En este caso algunos puntos de entrenamiento tendrán un error diferente de cero, y existirá habrá que buscar un equilibrio entre el error de predicción y la horizontalidad del tubo. En la figura 2.6aɛ fue configurado en 0.5 y no existe ningún tubo de ancho 1 que pueda encerrar todos los datos.

Para el caso lineal, la función de regresión con soporte de vectores puede ser escrita:

(Ec. 2.7)

Los vectores de soporte son aquellos puntos que no caen estrictamente dentro del tubo –o sea, los puntos afuera del tubo y sobre el borde (Ver figura 2.6a). Todos los puntos dentro del tubo se les asigna coeficiente 0 y pueden ser eliminados de los datos de entrenamiento sin cambiar la salida del proceso de aprendizaje.

En la mayoría de los casos los datos no son linealmente separables, son problemas no lineales, las MVS proveen soporte para este tipo de casos. Cuando los datos no son linealmente separables en el espacio de entradas, son problemas no lineales; En este caso se puede realizar una transformación no lineal del espacio de entradas, en un espacio de características. Este nuevo espacio permite que los datos puedan ser separados linealmente de manera que se pueden aplicar los mismos razonamientos que para las MVS lineales. Así podemos utilizar la ecuación de regresión 1, reemplazando el producto escalar por una función de núcleo. Una función núcleo o kernel se puede definir como aquella que permite realizar la separación y el traslado de los datos al espacio de características. Existen diversos kernels predeterminados conocidos entre los cuales se destaca el lineal, el RBF (Función de Base Radial), el polinomial, el sigmoidal.

El algoritmo de SVM para regresión funciona buscando simultáneamente una minimización del error y una maximización de la horizontalidad de la función de regresión. Sin embargo, cuando los datos no son linealmente separables incluso luego de aplicar una transformación, algunos puntos no encajan en el tubo (figura 2.6 c)y no existe un tubo con error igual a 0. Luego deberá realizarse una compensación entre el error en la predicción y la horizontalidad del tubo. Esta compensación es controlada forzando un límite superior C en el valor absoluto de los coeficientes . El límite superior restringe la influencia de los vectores de soporte en la forma de la función de regresión y es un parámetro que el usuario debe especificar en adición a ɛ. Mientras más grande sea C lo más cerca la función encajaralos datos. En el caso degenerado (ɛ=0) el algoritmo simplemente realiza una regresión de *error absoluto mínimo* utilizando la restricción del coeficiente y todas las instancias de entrenamiento se transforman en vectores de soporte. Contrariamente si ɛ es suficientemente grande como para que el tubo acomode todos los datos, el error se vuelve 0, no hay compensación para hacer y el algoritmo obtiene el tubo más horizontal que encierra a los datos indiferentemente del valor de C.

Comparado a otros métodos, como los árboles de decisión, incluso los algoritmos más veloces para soporte de vectores son lentos cuando son aplicados en un contexto no lineal. Por otro lado, suelen producir clasificadores muy precisos debido a los detalladosy complejos limites de decisión que pueden ser obtenidos.

1. **
2. **
3. **

Fig. 2.6: Regresión con soporte de vectores: (a) ɛ = 1, (b) ɛ = 2, y (c) ɛ = 0.5

**2.3 Problema de predicción de oleaje**

En la actualidad existen una variedad de deportes acuáticos que se ven beneficiados por los pronósticos acerca del oleaje en una zona cercana a la costa. Ejemplos de estos pueden ser el surf, windsurf, kitesurf, kayac, etc. Los deportistas están sumamente atentos a las condiciones que presenta el mar para evaluar la playa en la que van a practicar el deporte. Utilizando un pronóstico de oleaje el deportista puede planificar sus actividades para lograr estar presente en el lugar con mejores condiciones. Eventualmente el deportista puede evitarse recorridas extensas buscando el sitio con las mejores condiciones del día.

Actualmente los pronósticos de oleaje son derivados a partir de la resolución de un modelo matemático global sumamente complejo. Este provee de la predicción de la altura de las olas, la dirección de las mismas y el tiempo entre una ola y otra (período). Si bien este pronóstico tiene buen rendimiento en alta mar, no sucede lo mismo en la cercanía de la costa. Esto se debe a que la interacción de las olas con el lecho marino cercano no es contemplada por estos modelos. De acuerdo a la geografía de la costa puede suceder que en un rango de 15 km un lugar presente condiciones apropiadas para la práctica del deporte mientras que en otro las condiciones sean desfavorables. Se sabe que dentro de una misma costa expuesta a las mismas condiciones de mar las condiciones del lecho marino de una determinada playa pueden aumentar o disminuir el tamaño de las olas significativamente. Por esta razón es que la utilización de un pronóstico preciso en las cercanías de la costa es de suma utilidad.

Existen modelos numéricos para contemplar el comportamiento de las olas en las cercanías de la costa (Booij et al., 1999). Los mismos reciben como entrada los datos de pronóstico de alta mar e información acerca del lecho marino. Utilizando estos datos los mismos obtienen un refinamiento del pronóstico inicial. Si bien estos modelos poseen un buen comportamiento, su limitación está relacionada con la obtención de los datos del lecho marino. Usualmente la obtención de estos datos es costosa y no se encuentra disponible en todos los lugares.

Nuestra propuesta consiste en utilizar modelos de aprendizaje de máquina para aprender de casos pasados como un determinado pronóstico de alta mar se vio reflejado en las cercanías de la costa. De esta manera ante un nuevo pronóstico de alta mar se podrá predecir cuál será el efecto del mismo en las cercanías de la costa. El pronóstico de alta mar está representado por una grilla que cubre todo el planeta, donde cada punto posee la información de los parámetros de la ola en ese lugar. En la figura 2.7 se muestra la porción del pacifico de dicha grilla y resaltados los puntos más cercanos a la isla de Oahu, Hawái, los cuales pueden ser utilizados para realizar el pronóstico de olas de las diferentes playas de la isla.



Fig. 2.7: Puntos de pronóstico del modelo de alta mar.

Nos centraremos en la predicción de la altura de la ola en la cercanía de la costa, mejorando la predicción del pronóstico de alta mar. Para realizar esta predicción utilizaremos métodos de regresión los cuales tendrán como datos de entrada: el pronóstico de alta mar y la observación de la altura en la cercanía de la costa.

**2.4 Trabajos relacionados a la predicción de oleaje**

A continuación se describen las características de los principales trabajos relacionados a la predicción de la altura de la ola mediante el uso de técnicas de regresión.

Los trabajos [4] y [5] se caracterizan por enfocarse en la predicción de la altura de la ola en la zona cercana a la costa donde se realiza la práctica de surf. Estos a partir de la información del estado del mar en una zona de alta mar(X) infieren el tamaño que presentara este mismo estado en la zona de surf (Y). Es decir el objetivo es inferir todo el proceso de transformación que experimenta un estado del mar desde el punto X, hasta el punto Y. Para lograr esto definen un modelo de regresión en donde las variables predictivas están dadas por los datos en alta mar (X), y la variable de respuesta es la altura de la ola en la playa (Y). ([5]) utiliza como variables predictivas los datos de alta mar recolectados por una boya marina, mientras que ([4]) utiliza las previsiones oceánicas brindadas por la resolución del modelo numérico WAVEWATCH III (ver Capítulo 3 – Datos para predicción de oleaje). En ambos trabajos la variable de respuesta está representada por las observaciones visuales recolectadas por un observador experto el cual registra el tamaño de ola promedio en un determinado momento del día.

En los trabajos [6] y [7] la predicción de la altura de la ola en el mismo lugar donde se recolectan los datos. Estos utilizando el estado del mar de las últimas horas en un lugar X, infieren cual será la altura de la ola en ese mismo lugar con horas de anticipación. Para lograr esto se plantea un modelo de regresión donde las variables predictivas están dadas por las lecturas de una boya y la variable de respuesta está dada por la altura de la ola que se presentara en ese mismo lugar con ‘n’ horas de anticipación. Estos trabajos difieren del objetivo que presenta nuestro caso de estudio de predecir la altura de la ola en una zona diferente de donde son leídos los datos del mar. A pesar de esta diferencia los trabajos mencionados son validos como referencia por estar relacionados con el mismo fenómeno natural.

En [4], [6], y [7] se desarrollan modelos de regresión en base a redes neuronales, mientras que [4] utilizan un modelo de regresión polinomial.

Las redes neuronales desarrolladas, son todas multicapa de propagación hacia adelante. En todas ella se usa una función de transferencia del tipo sigmoidal. Los algoritmos de aprendizaje utilizados son diversos. [6] evalúa los algoritmos de Propagación hacia atrás, Gradiente conjugado y Correlación en cascada, obteniendo similares resultados en términos de correlación entre la altura observada y la altura predicha, aunque se observa un comportamiento superior en términos de tiempo de convergencia mediante el algoritmo de correlación en cascada. [4] experimenta con el algoritmo de aprendizaje de Levenburg-Marqhardt, el mismo es una forma más robusta del algoritmo de Gauss-Newton. [7] utiliza el algoritmo de “Resilient backpropagation”, el cual junto con Correlación en cascada y Levenburg-marqhardt es uno de los más rápidos para hallar los pesos óptimos. La cantidad de nodos a utilizar en la capa oculta (h) es obtenida mediante la fórmula empírica: h = (2z + 1), siendo z el número de entradas a la red ([19]). ([4]) utiliza una red con 6 nodos en la capa oculta, mientras que [6] utiliza el algoritmo de correlación en cascada el cual además de obtener los pesos óptimos se encarga de la construcción de la red en la capa oculta para hallar la estructura de red optima.

En [7] se experimenta, obteniendo resultados positivos, con una técnica especial en la cual se conectan dos redes neuronales. Donde la salida de la primera red es la entrada de la segunda. Según los autores esto permite a la primera red concentrarse en aprender las interacciones complicadas, mientras que la segunda red realiza un refinamiento a las predicciones de la primera.

Diversas comparaciones entre los modelos desarrollados y modelos de predicción ya existentes son realizadas en los trabajos. En [7] se compara el modelo de red neuronal desarrollado con un modelo estadístico auto regresivo obteniéndose mejores resultados con el primero. [4] realiza una comparación entre el modelo de red neuronal y el modelo SWAN ([10]), el cual es un modelo numérico para predicción de olas en la zona costera, obteniendo mejores resultados con el primero. También se realizan comparaciones entre un modelo de regresión lineal y uno no lineal, obteniendo mejores resultados con el modelo no lineal.

Todos los trabajos analizados presentan un desempeño satisfactorio en la predicción de olas, de esta manera dejando en evidencia que los métodos de regresión para la predicción de olas son una alternativa viable a los métodos de modelado numérico, y que además los mismos son útiles para refinar los pronósticos dados por los modelos de oleaje de alta mar en la cercanía de la costa.

Nuestro trabajo estará alineado con los objetivos de los trabajos de [4] y [5], es decir predecir la altura de la ola en la zona de surf a partir de datos de alta mar, donde además de los métodos de regresión lineal y redes neuronales evaluaremos métodos no tenidos en cuenta como los árboles de regresión y las maquinas de soporte vectorial. Luego utilizando el clasificador más eficiente construiremos un sistema que permita al usuario, a partir de un pronóstico más preciso, decidir cuál será la playa que mejor se adapte a sus preferencias.

En la siguiente tabla se resumen los diferentes trabajos analizados.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Trabajo** | **Modelo de regresión** | **Algoritmo de Aprendizaje** | **Variables predictoras** | **Variables de respuesta** |
| [6] | Red neuronal Multilayer Feedforward | Propagación hacia atrás;  Correlación en cascada;  Gradiente conjugado | Lecturas de la altura de la ola en la boyad de las ultimas N horas. | Valor de la altura de la ola con tiempo de anticipación de N horas. |
| [7] | Red neuronal Multilayer Feedforward | Resilient back- propagation | Lecturas de las ultimas 48 hs de la altura de la ola en la boya | 24 salidas cada una representando el valor de la altura de la ola en un periodo de anticipación. |
| [4] | Red neuronal Multilayer Feedforward | Levenburg-marqhardt | Parámetros del modelo WAVEWATCH III | Observación visual desde la playa de la altura de la ola. |
| [5] | Regresión polinomial | - | Lectura del período y tamaño de ola en una boya en alta mar. | Observación visual desde la playa de la altura de la ola. |

Tabla 2.2: Modelos de regresión utilizados en trabajos relacionados a la predicción de la altura de ola.

# Capitulo 3 – Datos para predicción de oleaje

## 3.1 Olas – Conceptos generales

Las olas de la superficie del océano son el resultado de las fuerzas que actúan sobre el océano. Las fuerzas naturales predominantes son la presión de la atmosfera (especialmente a través de los vientos), terremotos, la gravedad de la tierra, los cuerpos celestiales (la Luna y el Sol), la fuerza de Coriolis (debido a la rotación de la tierra), y la tensión de la superficie ([37]).

Las olas generadas por la acción del viento están presentes casi en cualquier lugar del océano. Estas olas son generadas por vientos en algún lugar del océano, puede ser localmente o a miles de kilómetros de distancia. Estas afectan un diverso rango de actividades como la navegación, pesca, recreación, industria costera y de tierra, la administración de las costas (defensas costeras) y el control de la polución. También son muy importantes en los procesos climáticos ya que participan ampliamente en los intercambios de calor, energía, gases y partículas entre el océano y la atmosfera. En este trabajo nos centraremos en la predicción de este tipo de olas. Especialmente nos centraremos en la predicción de las olas en la zona de surf, es decir donde las olas rompen, siendo las principales actividades afectadas por estas las de recreación (deportes de tabla, bañistas, kayacs etc.), navegación y administración costera.

La presión que ejerce el viento sobre una determinada área del océano **genera** olas. Mientras mayor sea la intensidad, el área de cobertura y la duración del viento mayores serán las olas producidas. Las olas que se encuentran en la zona de generación son denominadas olas de viento, y a medida que las mismas se **dispersan** y salen de esta zona estas se denominan oleaje. Usualmente estas olas viajan largas distancias, hasta encontrarse con aguas poco profundas donde finalmente rompen y **disipan** toda su energía.

Una ola puede ser caracterizada por su **altura**, **dirección** y **periodo**. La altura de la ola es la distancia vertical entre la cresta y la base. El largo de la ola es la distancia entre dos crestas sucesivas. El periodo de la ola es el tiempo entre dos crestas sucesivas. (Ver figura 3.1)



Fig. 3.1: Partes de la ola.

Ya que no es posible observar dos olas iguales, las medidas utilizadas para describirlas son de naturaleza estadística. A continuación se explican las medidas comúnmente utilizadas:

* = Altura promedio de ola.
* = Altura máxima de ola.
* = Periodo promedio de las olas.
* = Periodo promedio de las 1/n olas más altas.
* = Altura promedio de las 1/n olas mas altas (Ej. Si todas las Alturas de las olas medidas son ordenadas de manera descendiente, la 1/n parte conteniendo las olas más altas. Debería ser tomada y H1/n es el promedio de la altura de estas olas.

Un valor comúnmente usado es N = 3:

* = Altura de ola significante, (este valor se aproxima al valor de la altura que reportaría un observador visual)

En un punto dado del mar en cualquier momento existen una variedad de trenes de olas tanto de viento como de swell, cada tren tiene una energía, un periodo y una dirección dada. Esta compleja combinación de trenes de olas puede ser descripta mediante el espectro de olas El **espectro de olas** es una expresión matemática para representar los diferentes trenes de olas que pasan por un punto – desde todas las direcciones y de todos los periodos de olas diferentes. En la figura 3.2 se muestra un grafico polar de este espectro para un punto del océano. La energía de cada tren de olas se grafica junto con la dirección y el periodo. El periodo disminuye del centro hacia el borde, el color indica el nivel de energía del tren de olas y el ángulo indica dirección. El concepto de espectro es ampliamente utilizado por los diferentes modelos de olas.



Fig. 3.2: Grafico polar del espectro de olas ([36]).

Los datos acerca de las olas son necesarios para el estudio del comportamiento del mar, para los pronósticos de olas y para propósitos climáticos. Existen tres tipos de datos diferentes: datos observados, datos medidos, información de hindcast (en español, retro-análisis) ([37]). Los datos observados suelen provenir de reportes de personas arriba de barcos o desde lugares costeros, por ejemplo puestos de observación de guardavidas. Los datos medidos provienen de las lecturas de instrumentos como boyas marinas. Los datos de hindcast corresponden a datos del estado del mar producidos por un modelo de olas en donde las entradas al modelo corresponden a datos de viento archivados. Como estos datos de viento son reales y no son un pronóstico, se espera que el estado del mar producido por el modelo sea también una aproximación bastante real de cómo se encontraba el mar en ese momento (si el modelo es bueno). En este trabajo se utilizan datos del mar provenientes de **observaciones visuales** y de **información de hindcast** resultante del modelo WAVEWATCH III. A continuación explicaremos los detalles de cada uno de estos.

## 3.2 Observaciones Visuales

Con el desarrollo del surf en 1960 en el North Shore de la isla de Oahu (Hawái) las observaciones del estado de las olas hechas por los surfistas se volvieron rutinarias. En 1970 se sumaron las de los guardavidas y los reportes de surf comerciales.

Las observaciones son reportadas como un rango de alturas. Los observadores ignoran las olas más pequeñas. El límite superior del rango es equivalente a la medida , es decir al promedio del 10% de las olas más altas observadas en un lapso de tiempo. El límite inferior es equivalente al promedio del %33 de las olas más altas observadas ().

Una base de datos digital[[1]](#footnote-2) con los reportes de surf para la zona de Hawái fue creada por Larry Goddard y Patrick Caldwell. Las observaciones datan desde el año 1968 hasta el 2004. La altura de las olas se encuentra en la medida HSF (Hawaiian Scale Feet). Los reportes provienen de observaciones realizadas por los autores así como también de los guardavidas, reportes comerciales, y surfistas profesionales de la época. El valor de altura en la base de datos corresponde al límite superior del reporte de surf de mayor altura del día, es decir la mayor registrada durante el día. En la figura 3.3 se pueden apreciar los lugares para los que se encuentran disponibles las observaciones.

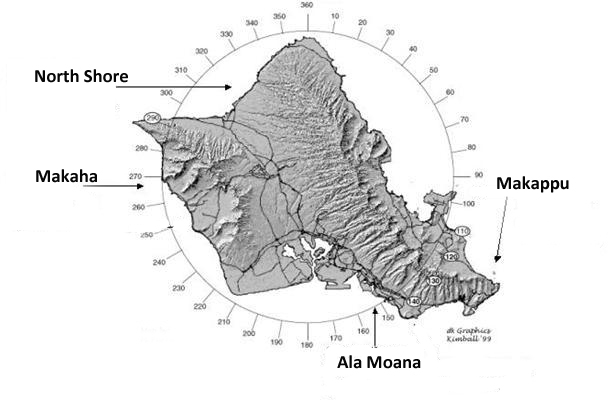


Fig. 3.3: Lugares de la isla de Oahu, Hawái con reportes de surf disponibles en la base de datos Goddard-Caldwell.

Se realizaron comparaciones de los reportes desde 1982 al 2002 con las lecturas de una boya marina perteneciente a NOAA (National Oceanic and Atmospheric Administration) la cual está ubicada a unos 400 km al oeste-noroeste de Oahu. A partir de las comparaciones se determinó que las observaciones son consistentes a través del tiempo con respecto a la estimación de la altura de la ola derivada de las lecturas de la boya ([29]).

La altura de las olas de la base de datos se encuentra en la escala HSF (Hawaiian Scale Feet). Esta forma de medición es originaria de Hawái, pero no es la medida estándar de medición de olas. La medida estándar es denominada base-cresta, y mide la distancia vertical de la cara de la ola desde la base hasta la cresta al momento de romper. En la figura 3.2 se pueden apreciar los dos tipos de mediciones.



Fig. 3.4: Mediciones de olas, HSF (Hawaiian Scale Feet) vs base-cresta.

La transformación de HSF a base-cresta para las islas de Hawái fue estudiada en ([5]), determinando que la altura de base-cresta representa el doble de la altura en el formato HSF con un margen de error dentro del 10-20%, para todo el rango de alturas encontradas en Hawái.

En la figura 3.5 se puede apreciar una fotografía de una ola vista de frente cuya altura fue estimada en 8 HSF. Se realiza la transformación a base-cresta utilizando como referencia la altura del surfista. Se considera que el surfista de pie y flexionado sobre la tabla posee una altura de 5 pies. A partir de esta referencia se estima que el tamaño base-cresta de la ola al momento de romper es de 15-16 pies, siendo el doble de la altura reportada en HSF.



Fig. 3.5: Transformación de HSF a Base-cresta tomando como referencia la altura del surfista.

## 3.3 Modelo WAVEWATCH III

WAVEWATCH III™ ([33], [34] y [35]) es un modelo de olas de tercera generación que contempla la generación, dispersión y disipación de las olas por la acción del viento. El modelo fue desarrollado por la entidad americana NOAA([National Oceanic and Atmospheric Administration](http://www.noaa.gov/) ) y es una mejora de los modelos WAVEWATCH, desarrollado en Delft University of Technology ([30] y [31]) y WAVEWATCH II, desarrollado en la NASA, Goddard Space Flight Center ([32]). WAVEWATCH III, difiere de sus predecesores en muchos puntos importantes como las ecuaciones gobernantes, la estructura del modelo, los métodos numéricos y las características físicas contempladas. Actualmente este modelo se encuentra en desarrollo continuo a cargo del EMC (Environmental Modelling Center), una delegación de NOAA.

La entrada al modelo consiste principalmente del pronóstico del viento sobre la superficie además de los datos específicos del lecho marino y de las costas. La estructura del modelo está representada por una grilla. Para cada pronóstico de viento disponible se calcula el correspondiente pronóstico de olas. El modelo calcula el espectro de olas (ver Sección 3.1) para cada punto de la grilla. A partir de este espectro se realiza una etapa de post-procesamiento para obtener los diversos parámetros del estado del mar en cada punto. Estos parámetros representan la salida del modelo. El espectro completo solo es guardado para ciertos puntos de la grilla, debido a la gran cantidad de información que el mismo involucra (Ver figura 3.6).



Fig. 3.6: Vista general de las entradas y salidas de un modelo de olas genérico ([36]).

A lo largo de la historia NOAA implementó diferentes sistemas de pronósticos de olas a partir del modelo WAVEWATCH. En la actualidad NOAA mantiene y distribuye públicamente las salidas de dos sistemas, NOAA WAVEWATCH III 2.22 y NOAA WAVEWATCH III 3.14. Ambos sistemas se corren 4 veces al día en intervalos de 6 horas. Cada corrida produce pronósticos en incrementos de 3 horas desde la hora inicial hasta las 180 horas (7 días y medio).

El sistema NOAA WAVEWATCH III 2.22, utiliza 5 modelos diferentes. Un modelo global con una resolución de grilla de 1 x 1.25 y 4 modelos regionales con una resolución de grilla de 0.5 X 0.5. Los parámetros de ola que se obtienen en la salida son: Altura Significante/Período/Dirección de ola dominante; Dirección y velocidad del viento. Las salidas del sistema se encuentran disponibles en forma de mapas, reportes de texto con el espectro completo para ciertas puntos de la grilla, y archivos binarios con la información de los parámetros calculados para cada punto de la grilla.

A partir de la versión 3.0 del modelo WAVEWATCH III es posible el anidamiento de grillas, haciendo posible plasmar en un mismo modelo, diferentes niveles de pronóstico a diferentes niveles de resolución. Además debido a mejoras en los algoritmos de particionamiento del espectro es posible distinguir entre las olas de viento y olas de swell (ver sección 3.1).

El sistema NOAA WAVEWATCH III 3.14 aprovecha estos avances definiendo un solo modelo compuesto. Este está conformado por una grilla de alcance global con una resolución de 0.5 x 0.5 grados, sub-regiones de mediana resolución de 0.16 X  0.16 grados y para las costas de Estados Unidos grillas de alta resolución de 0.06 X 0.06 grados (cuadrados de 4 millas náuticas). Los parámetros obtenidos a partir de este modelo tienen en cuenta la separación entre olas de viento y el oleaje y consisten en: *Altura Significante de ola Combinada* (Viento y swell), *Altura/Periodo/Dirección de ola dominante*, *Altura/Periodo/Dirección de ola de viento*, Altura/Periodo/Dirección del oleaje. *Dirección y velocidad del viento*. La salida del sistema se encuentra disponible en reportes de texto con la información del espectro completo para ciertos puntos de la grilla y en formato binario con los parámetros calculados para cada punto de la grilla.

NOAA mantiene archivos de Hindcast[[2]](#footnote-3) (ver Sección 3.1) del sistema 2.22 desde el año 1997 al 2005. Mientras que del sistema 3.14 no se encuentra disponibles información de Hindcast pero sí la salidas diarias del modelo. Estas se encuentran disponibles desde el año 2007 al 2009.

Para entrenar y validar los clasificadores desarrollados en este trabajo se utilizó la información del archivo de hindcast del modelo NOAA WAVEWATCH III 2.22. Ya que es necesario acoplar la información del modelo WAVEWATCH III con las observaciones visuales y este archivo era el único contemporáneo a las observaciones visuales disponibles.

Para realizar nuevos pronósticos, los clasificadores recibirán los datos provenientes de la salida diaria del sistema NOAA WAVEWATCHIII 2.22, ya que los mismos han sido entrenados con datos provenientes de esa versión del sistema.

En la figura 3.7 se muestran los puntos de la grilla del modelo global del sistema WAVEWATCH III 2.22 que rodean a la isla de Oahu, Hawái. La información del estado del mar en estos puntos producido por el modelo fue utilizada para entrenar y validar los diferentes clasificadores desarrollados, así como también es utilizada para producir los nuevos pronósticos.



Fig. 3.7: Puntos de la grilla global (1⁰ x 1. 25⁰) del sistema NOAA WAVEWATCH III 2.22 alrededor de Oahu, Hawái.

## 3.4 Pre procesamiento

Para realizar el entrenamiento de los clasificadores es necesario definir las instancias que serán la entrada a los mismos. Las instancias estarán compuestas por el estado de altamar derivado de la información proveniente del modelo WAVEWATCH III, mientras que la clase de la instancia se corresponderá con la observación visual desde la playa de la altura de la ola correspondiente. A continuación se describe el filtrado general realizado a los datos para poder generar los conjuntos de entrenamiento.

Dado que las lecturas de WAVEWATCH III se actualizan cada 3 horas, tanto durante el día como la noche, y debido a que las observaciones visuales se han realizado en horarios en que se cuenta con luz solar, el primer filtro aplicado a la información que ofrece el modelo WAVEWATCH III fue eliminar todas las lecturas que no correspondían con un horario de luz natural en Oahu.

Por otra parte, las observaciones visuales nos indican la altura de la ola más grande que arribo a la costa en el día, por esto el segundo filtro aplicado a las lecturas de WAVEWATCH III fue dejar únicamente la lectura diaria con la mayor altura de ola. De esta forma acoplamos el estado de altamar brindado por el modelo WAVEWATCH III y la observación de la altura de la ola registrada en la costa.

Todas las alturas de la base de datos Goddard-Caldwell se encuentran en la medida HSF por lo tanto fue necesario convertir las mismas a la medida Base-Cresta. Se realizó la transformación utilizando la formula 1 HSF = 2 \* Base-Cresta (Caldwell y Aucan, 2004).

Los filtros anteriormente descriptos fueron los que finalmente se utilizaron como filtrado básico de los datos de entrada en todas las evaluaciones que se describirán en el capítulo siguiente.

Otros filtros generales que también tuvimos en cuenta para realizar pruebas (descartados por no mejorar los resultados) fueron por ejemplo, utilizar el promedio de las lecturas de olas más altas registradas por el WAVEWATCH III de cada día y filtrar las lecturas cuya dirección de ola no estaba dirigida a la costa a evaluar.

## 3.5 Modelos de instancia

Como se mencionó en el Capítulo 2 – Estado del arte, al utilizar un algoritmo de aprendizaje de maquina es necesario incorporar conocimiento previo acerca del tema tanto en el modelo de instancia como en los conjuntos de entrenamiento a utilizar. Es necesario seleccionar los atributos más relevantes, así como también el atributo de clase, dependiendo del problema que se intenta resolver. En este trabajo se experimentó con diferentes modelos de instancias. Para cada modelo de instancia se armaron diferentes conjuntos de entrenamiento, los cuales fueron utilizados como entrada en los diferentes algoritmos de regresión. A continuación se detallan las instancias y conjuntos de entrenamiento generados.

Se armaron 3 modelos de instancias diferentes. Las mismas varían en la cantidad de atributos de utilizados, siendo el atributo de clase siempre el mismo. Los atributos de instancia pertenecen a los datos de alta mar provistos por el modelo WAVEWATCH III. El atributo de clase corresponde a la observación visual de la altura de la ola al momento de romper cerca de la playa.

***Instancia 1:***

*{altura significante de ola, dirección promedio, periodo promedio, observación visual}.*

Este modelo de instancia utiliza la información de un solo punto del modelo WAVEWATCH III y la observación visual de la altura en la playa.

***Instancia 2:***

*{Altura significante de ola X1, dirección promedio X1, periodo promedio X1, Altura significante de ola X2, dirección promedio X2, periodo promedio X2, observación visual}*

Este modelo de instancia utiliza la información de dos puntos, X1 y X2, del modelo WAVEWATCH III. Generalmente una playa está rodeada por más de un punto del modelo, por este motivo se diseño una instancia que contenga la información de los diferentes puntos que rodean una determinada playa.

***Instancia 3:***

*{altura significante de ola T1, dirección promedio T1, período promedio T1,…, altura significante de ola Tn, dirección promedio Tn, periodo promedio Tn, observación visual}*

Este modelo de instancia utiliza la información de un punto del modelo de WAVEWATCH III, pero con la información de ese punto de las últimas N lecturas. De esta manera se obtiene el último estado del mar, más la información de los últimos N-1 estados. Este modelo fue propuesto para experimentar si el clasificador mejoraba las predicciones, por el hecho de contar con información acerca de cómo es que el estado actual del mar fue desarrollándose los días anteriores.

## 3.6 Conjuntos de entrenamiento

Los datos disponibles para armar los conjuntos de entrenamiento pertenecían al periodo 1997-2004. Se probaron armar conjuntos de entrenamiento con las diferentes modelos de instancias y utilizando diferentes rangos de fechas (ver Tabla 3.1). Por ejemplo se armaron conjuntos de entrenamiento con información desde el año 1997 hasta el 2004. También se generaron conjuntos de entrenamiento, que abarcaban menos años, por ejemplo del 2002 al 2004. Inclusive se tuvo en cuenta los ciclos naturales de las olas de cada playa. Por ejemplo el North Shore posee dos épocas marcadas en cuanto a tamaño de olas. Así es que se generaron conjuntos de entrenamientos independientes para cada época de olas.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Modelo instancia** | **Numero**  **lecturas** | **Puntos WWIII** | **Fecha** | **#Instancias** | **Playa** |
| 1 | 1 | 1 | S1 | 1997-2004 | 2800 | North Shore |
| 2 | 1 | 1 | S4 | 1997-2004 | 2800 | North Shore |
| 3 | 2 | 1 | S1,S4 | 1997-2004 | 2800 | North Shore |
| 4 | 3 | 2 | S1 | 1997-2004 | 2800 | North Shore |
| 5 | 3 | 3 | S1 | 1997-2004 | 2800 | North Shore |
| 5 | 3 | 4 | S1 | 1997-2004 | 2800 | North Shore |
| 7 | 1 | 1 | S1 | 2002-2004 | 750 | North Shore |
| 8 | 1 | 1 | S1 | Noviembre - Marzo; 2002-2004 | 450 | North Shore |
| 9 | 1 | 1 | S1 | Abril – Octubre; 2002-2004 | 630 | North Shore |

Tabla 3.1: Conjuntos de entrenamiento generados para la playa North-Shore.

# Capítulo 4 - Experimentación

En el presente capítulo se detallan los experimentos realizados para encontrar el mejor clasificador y modelo de instancia para predecir el tamaño que tendrá la ola al momento al llegar a la costa, a partir de la información de alta mar brindada por el modelo de olas WAVEWATCH III. Básicamente se exponen todos los resultados obtenidos con diferentes clasificadores y modelos de instancias para argumentar la combinación optima (clasificador/modelo) elegida para el desarrollo del proyecto. Luego se detallan los resultados obtenidos en diferentes olas alrededor de la isla de Oahu – Hawái utilizando el algoritmo y modelo optimo.

## 4.1 Consideraciones generales

Todas las pruebas fueron realizadas utilizando el soporte de la herramienta Weka. La misma librería desarrollada en Java para realizar minería de datos, proveyendo soporte para pre-procesamiento de los datos, algoritmos de clasificación, y visualización de los resultados entre otras cosas.

Durante las pruebas se utilizaron diversos algoritmos con el objetivo de evaluar su desempeño en el problema en cuestión, los mismos son listados en la siguiente tabla.

|  |  |
| --- | --- |
| **Algoritmo** | **Clase** |
| Regresión Lineal | weka.classifiers.functions.LinearRegression |
| Red Neuronal Multicapa | weka.classifiers.functions.MultilayerPerceptron |
| Soporte de Vectores | weka.classifiers.functions.SVMreg |
| Arboles de Regresión | weka.classifiers.trees.M5P |

Tabla 4.1: Algoritmos de aprendizaje de maquina estudiados

### 4.1.1 Parametrización

Los algoritmos de regresión utilizados son parametrizables. Siendo que no existe una base teórica completa para determinar que parámetros son apropiados para cada tipo de problema, la configuración optima suele realizarse a prueba y error.

Para obtener los parámetros óptimos de cada clasificador se fueron probando diferentes combinaciones de los mismos, aquellas configuraciones que lograban indicadores destacados, resultantes de la validación cruzada, fueron utilizadas para el entrenamiento de los clasificadores de las diferentes playas.

### 4.1.2 Medidas de evaluación utilizadas

Una vez entrenado un clasificador es necesario evaluar su desempeño. Para este fin se utilizan instancias de prueba cuya clase se conoce, el clasificador realiza la predicción de la clase y luego se compara el valor predicho y el valor real. Este proceso es repetido para un conjunto de instancias significante y luego diversos indicadores estadísticos son extraídos.

En este trabajo se utilizó la técnica de validación cruzada de 10 conjuntos (Ver capítulo 2 – Estado del arte), para realizar la optimización de los parámetros (explicado a continuación) de los clasificadores. Mientras que para el entrenamiento y validación final de cada clasificador se utilizó un conjunto de entrenamiento independiente y se realizó 10 veces 10 fold-cross validation. En ambas casos se calcularon dos indicadores para medir el desempeño de los clasificadores: la correlación y el error absoluto promedio.

La correlación indica cual es el grado de relación entre el valor predicho y el valor verdadero. El valor de correlación se encuentra en el intervalo [-1,1], siendo los extremos indicadores de buen comportamiento del clasificador, no así los valores cercanos a 0.

El error absoluto promedio, indica el promedio entre la diferencia del valor predicho y el valor verdadero en cada instancia de prueba. Mientras menor se este indicador mejor clasificador tendremos.

## 4.2 Modelos de instancia

Como se mencionó en el Capítulo 3 – Datos para predicción de oleaje, la selección de atributos que conforman cada modelo de instancia es de vital importancia para cualquier problema de aprendizaje de máquina, siendo que estos afectan directamente la performance del clasificador obtenido, independientemente del algoritmo de aprendizaje utilizado.

Para determinar los atributos que generaban los mejores resultados, se experimentó generando conjuntos de entrenamiento a partir de los modelos de instancias descriptos en el capitulo anterior. Estos conjuntos fueron utilizados como entrada a un algoritmo de aprendizaje de máquina y los clasificadores resultantes fueron comparados.

## 4.3 Resultados generales

A continuación se detalla el desempeño de los clasificadores obtenidos a partir de los diferentes modelos de instancias. Se itero entre los distintos conjuntos de entrenamiento y se evaluó cada uno con los algoritmos especificados en la Tabla 4.1. Debido a la gran cantidad de experimentos los mismos fueron divididos por playa y los resultados son resumidos en las siguientes tablas.

Cada tabla muestra los resultados de una ola en particular, ordenada por valor de correlación creciente y se estructuró de la siguiente manera:

* Columna 1: Clasificador de Weka. Representa el algoritmo de aprendizaje utilizado.
* Columna 2: Estrategia. Representa el modelo de instancia utilizado, el cual define los atributos que conforman cada instancia.
* Columna 3: Correlación resultante en Weka con la parametrización por defecto del algoritmo
* Columna 4: MAE. Error absoluto medio resultante en Weka con la parametrización por defecto del algoritmo.
* Columna 5: Correlación resultante en Weka con la parametrización optima del algoritmo
* Columna 6: MAE. Error absoluto medio resultante en Weka con la parametrización optima del algoritmo.

### 4.3.1 Resultados generales en Ala Moana

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Optimizados** | | **No Optimizados** | |
| **Clasificador** | **Estrategia** | **Correl.** | **MAE** | **Correl.** | **MAE** |
| weka.classifiers.functions.LinearRegression | WW3Last2LecturesStrategy | 0,26 | 0,54 | 0,25 | 0,54 |
| weka.classifiers.functions.LinearRegression | WW3Last3LecturesStrategy | 0,28 | 0,53 | 0,28 | 0,54 |
| weka.classifiers.functions.LinearRegression | WW3Last6LecturesStrategy | 0,28 | 0,53 | 0,28 | 0,53 |
| weka.classifiers.functions.LinearRegression | WW3Last8LecturesStrategy | 0,27 | 0,54 | 0,27 | 0,53 |
| weka.classifiers.functions.LinearRegression | WW3SimpleStrategy | 0,29 | 0,53 | 0,29 | 0,53 |
| weka.classifiers.functions.LinearRegression | TwoGridPointStrategy | 0,30 | 0,53 | 0,31 | 0,53 |
| weka.classifiers.functions.LinearRegression | WW3Last2DaysStrategy | 0,30 | 0,54 | 0,30 | 0,53 |
| weka.classifiers.functions.LinearRegression | WW3Last3DaysStrategy | 0,33 | 0,53 | 0,32 | 0,52 |
| weka.classifiers.functions.LinearRegression | WW3Last4DaysStrategy | 0,34 | 0,52 | 0,34 | 0,52 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last2LecturesStrategy | 0,41 | 0,51 | 0,41 | 0,56 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last3LecturesStrategy | 0,42 | 0,50 | 0,39 | 0,59 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last6LecturesStrategy | 0,41 | 0,50 | 0,30 | 0,61 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last8LecturesStrategy | 0,38 | 0,51 | 0,29 | 0,70 |
| weka.classifiers.functions.MultilayerPerceptron | WW3SimpleStrategy | 0,43 | 0,52 | 0,41 | 0,65 |
| weka.classifiers.functions.MultilayerPerceptron | TwoGridPointStrategy | 0,42 | 0,52 | 0,41 | 0,57 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last2DaysStrategy | 0,43 | 0,50 | 0,41 | 0,67 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last3DaysStrategy | 0,45 | 0,50 | 0,41 | 0,56 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last4DaysStrategy | 0,45 | 0,50 | 0,35 | 0,58 |
| weka.classifiers.functions.SMOreg | WW3Last2LecturesStrategy | 0,48 | 0,46 | 0,25 | 0,53 |
| weka.classifiers.functions.SMOreg | WW3Last3LecturesStrategy | 0,50 | 0,46 | 0,26 | 0,52 |
| weka.classifiers.functions.SMOreg | WW3Last6LecturesStrategy | 0,46 | 0,48 | 0,28 | 0,53 |
| weka.classifiers.functions.SMOreg | WW3Last8LecturesStrategy | 0,45 | 0,48 | 0,26 | 0,53 |
| weka.classifiers.functions.SMOreg | WW3SimpleStrategy | 0,47 | 0,47 | 0,29 | 0,52 |
| weka.classifiers.functions.SMOreg | TwoGridPointStrategy | 0,47 | 0,48 | 0,29 | 0,53 |
| weka.classifiers.functions.SMOreg | WW3Last2DaysStrategy | 0,48 | 0,47 | 0,31 | 0,52 |
| weka.classifiers.functions.SMOreg | WW3Last3DaysStrategy | 0,47 | 0,48 | 0,33 | 0,52 |
| weka.classifiers.functions.SMOreg | WW3Last4DaysStrategy | 0,45 | 0,50 | 0,34 | 0,52 |
| weka.classifiers.trees.M5P | WW3Last2LecturesStrategy | 0,46 | 0,48 | 0,46 | 0,49 |
| weka.classifiers.trees.M5P | WW3Last3LecturesStrategy | 0,47 | 0,49 | 0,46 | 0,48 |
| weka.classifiers.trees.M5P | WW3Last6LecturesStrategy | 0,45 | 0,50 | 0,44 | 0,50 |
| weka.classifiers.trees.M5P | WW3Last8LecturesStrategy | 0,43 | 0,50 | 0,42 | 0,51 |
| weka.classifiers.trees.M5P | WW3SimpleStrategy | 0,46 | 0,49 | 0,45 | 0,49 |
| weka.classifiers.trees.M5P | TwoGridPointStrategy | 0,49 | 0,49 | 0,48 | 0,48 |
| weka.classifiers.trees.M5P | WW3Last2DaysStrategy | 0,48 | 0,48 | 0,47 | 0,50 |
| weka.classifiers.trees.M5P | WW3Last3DaysStrategy | 0,46 | 0,49 | 0,47 | 0,47 |
| weka.classifiers.trees.M5P | WW3Last4DaysStrategy | 0,48 | 0,48 | 0,47 | 0,47 |

Tabla 4.2: Resultados generales de pruebas sobre la ola “Ala Moana”

Se puede ver en la tabla anterior, la fila marcada con verde resalta el clasificador y la estrategia o modelo de instancia que mejor performance mostró en base a las observaciones visuales suministradas. Por otra parte se puede observar (en esa misma línea) como el clasificador con los parámetros optimizados duplica la correlación de las predicciones respecto del que no fue optimizado. El algoritmo es el de SVM y la estrategia usada fue WWW3Last3LecturesStrategy (Ver capitulo 3 – Datos para predicción de oleaje – Modelos de instancias).

### 4.3.2 Resultados generales en Diamond Head

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Optimizados** | | **No Optimizados** | |
| **Clasificador** | **Estrategia** | **Correl.** | **MAE** | **Correl.** | **MAE** |
| weka.classifiers.functions.LinearRegression | WW3Last2LecturesStrategy | 0,21 | 0,55 | 0,22 | 0,55 |
| weka.classifiers.functions.LinearRegression | WW3Last3LecturesStrategy | 0,22 | 0,55 | 0,23 | 0,55 |
| weka.classifiers.functions.LinearRegression | WW3Last6LecturesStrategy | 0,22 | 0,55 | 0,23 | 0,55 |
| weka.classifiers.functions.LinearRegression | WW3Last8LecturesStrategy | 0,22 | 0,55 | 0,21 | 0,55 |
| weka.classifiers.functions.LinearRegression | WW3SimpleStrategy | 0,24 | 0,54 | 0,24 | 0,54 |
| weka.classifiers.functions.LinearRegression | TwoGridPointStrategy | 0,25 | 0,54 | 0,25 | 0,54 |
| weka.classifiers.functions.LinearRegression | WW3Last2DaysStrategy | 0,29 | 0,54 | 0,29 | 0,53 |
| weka.classifiers.functions.LinearRegression | WW3Last3DaysStrategy | 0,30 | 0,53 | 0,30 | 0,53 |
| weka.classifiers.functions.LinearRegression | WW3Last4DaysStrategy | 0,31 | 0,53 | 0,30 | 0,53 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last2LecturesStrategy | 0,34 | 0,53 | 0,32 | 0,55 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last3LecturesStrategy | 0,37 | 0,52 | 0,38 | 0,56 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last6LecturesStrategy | 0,38 | 0,51 | 0,33 | 0,56 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last8LecturesStrategy | 0,36 | 0,51 | 0,29 | 0,69 |
| weka.classifiers.functions.MultilayerPerceptron | WW3SimpleStrategy | 0,29 | 0,53 | 0,29 | 0,63 |
| weka.classifiers.functions.MultilayerPerceptron | TwoGridPointStrategy | 0,30 | 0,54 | 0,29 | 0,61 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last2DaysStrategy | 0,36 | 0,52 | 0,35 | 0,58 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last3DaysStrategy | 0,37 | 0,51 | 0,34 | 0,57 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last4DaysStrategy | 0,37 | 0,51 | 0,30 | 0,71 |
| weka.classifiers.functions.SMOreg | WW3Last2LecturesStrategy | 0,47 | 0,46 | 0,20 | 0,54 |
| weka.classifiers.functions.SMOreg | WW3Last3LecturesStrategy | 0,48 | 0,47 | 0,19 | 0,54 |
| weka.classifiers.functions.SMOreg | WW3Last6LecturesStrategy | 0,47 | 0,46 | 0,18 | 0,54 |
| weka.classifiers.functions.SMOreg | WW3Last8LecturesStrategy | 0,47 | 0,47 | 0,17 | 0,54 |
| weka.classifiers.functions.SMOreg | WW3SimpleStrategy | 0,40 | 0,49 | 0,23 | 0,54 |
| weka.classifiers.functions.SMOreg | TwoGridPointStrategy | 0,42 | 0,48 | 0,24 | 0,54 |
| weka.classifiers.functions.SMOreg | WW3Last2DaysStrategy | 0,42 | 0,48 | 0,26 | 0,53 |
| weka.classifiers.functions.SMOreg | WW3Last3DaysStrategy | 0,44 | 0,47 | 0,28 | 0,53 |
| weka.classifiers.functions.SMOreg | WW3Last4DaysStrategy | 0,43 | 0,48 | 0,29 | 0,52 |
| weka.classifiers.trees.M5P | WW3Last2LecturesStrategy | 0,37 | 0,51 | 0,37 | 0,52 |
| weka.classifiers.trees.M5P | WW3Last3LecturesStrategy | 0,35 | 0,52 | 0,35 | 0,52 |
| weka.classifiers.trees.M5P | WW3Last6LecturesStrategy | 0,34 | 0,51 | 0,34 | 0,53 |
| weka.classifiers.trees.M5P | WW3Last8LecturesStrategy | 0,34 | 0,55 | 0,31 | 0,53 |
| weka.classifiers.trees.M5P | WW3SimpleStrategy | 0,44 | 0,49 | 0,43 | 0,49 |
| weka.classifiers.trees.M5P | TwoGridPointStrategy | 0,39 | 0,51 | 0,38 | 0,52 |
| weka.classifiers.trees.M5P | WW3Last2DaysStrategy | 0,42 | 0,51 | 0,40 | 0,50 |
| weka.classifiers.trees.M5P | WW3Last3DaysStrategy | 0,41 | 0,51 | 0,39 | 0,52 |
| weka.classifiers.trees.M5P | WW3Last4DaysStrategy | 0,40 | 0,49 | 0,38 | 0,51 |

Tabla 4.3: Resultados generales de pruebas sobre la ola “Diamond Head”

Se puede ver en la tabla anterior, nuevamente, la fila marcada con verde mostrando el clasificador y estrategia que mejores resultados obtuvo. El algoritmo es el de SVM y la estrategia usada fue WWW3Last3LecturesStrategy (Ver capitulo 3 – Datos para predicción de oleaje – Modelos de instancias).

### 4.3.3 Resultados generales en North Shore

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Optimizados** | | **No Optimizados** | |
| **Clasificador** | **Estrategia** | **Correl.** | **MAE** | **Correl.** | **MAE** |
| weka.classifiers.functions.LinearRegression | WW3Last2LecturesStrategy | 0,87 | 0,97 | 0,87 | 0,95 |
| weka.classifiers.functions.LinearRegression | WW3Last3LecturesStrategy | 0,85 | 0,97 | 0,85 | 0,99 |
| weka.classifiers.functions.LinearRegression | WW3Last6LecturesStrategy | 0,87 | 0,93 | 0,87 | 0,94 |
| weka.classifiers.functions.LinearRegression | WW3Last8LecturesStrategy | 0,87 | 0,99 | 0,86 | 0,97 |
| weka.classifiers.functions.LinearRegression | WW3SimpleStrategy | 0,86 | 0,98 | 0,86 | 0,98 |
| weka.classifiers.functions.LinearRegression | TwoGridPointStrategy | 0,86 | 0,98 | 0,86 | 0,98 |
| weka.classifiers.functions.LinearRegression | WW3Last2DaysStrategy | 0,87 | 0,95 | 0,87 | 0,94 |
| weka.classifiers.functions.LinearRegression | WW3Last3DaysStrategy | 0,87 | 0,96 | 0,87 | 0,95 |
| weka.classifiers.functions.LinearRegression | WW3Last4DaysStrategy | 0,87 | 0,94 | 0,87 | 0,95 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last2LecturesStrategy | 0,88 | 0,91 | 0,88 | 1,08 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last3LecturesStrategy | 0,88 | 0,86 | 0,87 | 1,43 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last6LecturesStrategy | 0,84 | 0,88 | 0,81 | 1,25 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last8LecturesStrategy | 0,84 | 0,91 | 0,80 | 1,46 |
| weka.classifiers.functions.MultilayerPerceptron | WW3SimpleStrategy | 0,89 | 0,85 | 0,87 | 1,07 |
| weka.classifiers.functions.MultilayerPerceptron | TwoGridPointStrategy | 0,88 | 0,88 | 0,87 | 1,08 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last2DaysStrategy | 0,89 | 0,82 | 0,88 | 0,95 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last3DaysStrategy | 0,89 | 0,88 | 0,87 | 1,08 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last4DaysStrategy | 0,87 | 0,88 | 0,84 | 1,08 |
| weka.classifiers.functions.SMOreg | WW3Last2LecturesStrategy | 0,89 | 0,82 | 0,87 | 0,93 |
| weka.classifiers.functions.SMOreg | WW3Last3LecturesStrategy | 0,89 | 0,82 | 0,87 | 0,92 |
| weka.classifiers.functions.SMOreg | WW3Last6LecturesStrategy | 0,88 | 0,87 | 0,87 | 0,94 |
| weka.classifiers.functions.SMOreg | WW3Last8LecturesStrategy | 0,87 | 0,88 | 0,87 | 0,93 |
| weka.classifiers.functions.SMOreg | WW3SimpleStrategy | 0,89 | 0,83 | 0,86 | 0,96 |
| weka.classifiers.functions.SMOreg | TwoGridPointStrategy | 0,89 | 0,83 | 0,86 | 0,96 |
| weka.classifiers.functions.SMOreg | WW3Last2DaysStrategy | 0,90 | 0,80 | 0,87 | 0,92 |
| weka.classifiers.functions.SMOreg | WW3Last3DaysStrategy | 0,89 | 0,81 | 0,87 | 0,92 |
| weka.classifiers.functions.SMOreg | WW3Last4DaysStrategy | 0,89 | 0,82 | 0,87 | 0,92 |
| weka.classifiers.trees.M5P | WW3Last2LecturesStrategy | 0,88 | 0,87 | 0,89 | 0,85 |
| weka.classifiers.trees.M5P | WW3Last3LecturesStrategy | 0,88 | 0,88 | 0,88 | 0,87 |
| weka.classifiers.trees.M5P | WW3Last6LecturesStrategy | 0,87 | 0,88 | 0,88 | 0,94 |
| weka.classifiers.trees.M5P | WW3Last8LecturesStrategy | 0,87 | 0,90 | 0,87 | 0,89 |
| weka.classifiers.trees.M5P | WW3SimpleStrategy | 0,88 | 0,87 | 0,88 | 0,86 |
| weka.classifiers.trees.M5P | TwoGridPointStrategy | 0,88 | 0,90 | 0,88 | 0,90 |
| weka.classifiers.trees.M5P | WW3Last2DaysStrategy | 0,89 | 0,87 | 0,89 | 0,85 |
| weka.classifiers.trees.M5P | WW3Last3DaysStrategy | 0,88 | 0,89 | 0,88 | 0,87 |
| weka.classifiers.trees.M5P | WW3Last4DaysStrategy | 0,88 | 0,89 | 0,88 | 0,90 |

Tabla 4.4: Resultados generales de pruebas sobre la ola “North Shore”

Se puede ver en la tabla anterior, la fila marcada con verde mostrando el clasificador y estrategia que mejores resultados obtuvo. El algoritmo es el de SVM y la estrategia usada fue WWW3Last2DaysStrategy (Ver capitulo 3 – Datos para predicción de oleaje – Modelos de instancias). Por otra parte se puede ver que la segunda estrategia que mejores resultados obtuvo fue la fila marcada en azul la cual es la misma que la seleccionada en las tablas anteriores (Tabla 3.2 y 3.3) como mejor opción.

### 4.3.4 Resultados generales en Windward

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Optimizados** | | **No Optimizados** | |
| **Clasificador** | **Estrategia** | **Correl.** | **MAE** | **Correl.** | **MAE** |
| weka.classifiers.functions.LinearRegression | WW3Last2LecturesStrategy | 0,54 | 0,59 | 0,54 | 0,59 |
| weka.classifiers.functions.LinearRegression | WW3Last3LecturesStrategy | 0,54 | 0,60 | 0,54 | 0,60 |
| weka.classifiers.functions.LinearRegression | WW3Last6LecturesStrategy | 0,54 | 0,60 | 0,54 | 0,60 |
| weka.classifiers.functions.LinearRegression | WW3Last8LecturesStrategy | 0,54 | 0,59 | 0,54 | 0,59 |
| weka.classifiers.functions.LinearRegression | WW3SimpleStrategy | 0,54 | 0,59 | 0,55 | 0,59 |
| weka.classifiers.functions.LinearRegression | TwoGridPointStrategy | 0,57 | 0,57 | 0,56 | 0,57 |
| weka.classifiers.functions.LinearRegression | WW3Last2DaysStrategy | 0,56 | 0,59 | 0,56 | 0,58 |
| weka.classifiers.functions.LinearRegression | WW3Last3DaysStrategy | 0,56 | 0,59 | 0,56 | 0,58 |
| weka.classifiers.functions.LinearRegression | WW3Last4DaysStrategy | 0,55 | 0,59 | 0,56 | 0,59 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last2LecturesStrategy | 0,62 | 0,54 | 0,60 | 0,63 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last3LecturesStrategy | 0,61 | 0,54 | 0,59 | 0,57 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last6LecturesStrategy | 0,59 | 0,57 | 0,57 | 0,63 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last8LecturesStrategy | 0,53 | 0,58 | 0,52 | 0,59 |
| weka.classifiers.functions.MultilayerPerceptron | WW3SimpleStrategy | 0,65 | 0,52 | 0,66 | 0,62 |
| weka.classifiers.functions.MultilayerPerceptron | TwoGridPointStrategy | 0,66 | 0,52 | 0,66 | 0,60 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last2DaysStrategy | 0,63 | 0,53 | 0,67 | 0,75 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last3DaysStrategy | 0,63 | 0,53 | 0,68 | 0,71 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last4DaysStrategy | 0,65 | 0,53 | 0,64 | 0,56 |
| weka.classifiers.functions.SMOreg | WW3Last2LecturesStrategy | 0,70 | 0,49 | 0,53 | 0,57 |
| weka.classifiers.functions.SMOreg | WW3Last3LecturesStrategy | 0,74 | 0,47 | 0,53 | 0,58 |
| weka.classifiers.functions.SMOreg | WW3Last6LecturesStrategy | 0,71 | 0,49 | 0,52 | 0,58 |
| weka.classifiers.functions.SMOreg | WW3Last8LecturesStrategy | 0,70 | 0,48 | 0,52 | 0,59 |
| weka.classifiers.functions.SMOreg | WW3SimpleStrategy | 0,68 | 0,51 | 0,54 | 0,58 |
| weka.classifiers.functions.SMOreg | TwoGridPointStrategy | 0,72 | 0,47 | 0,56 | 0,57 |
| weka.classifiers.functions.SMOreg | WW3Last2DaysStrategy | 0,71 | 0,49 | 0,55 | 0,57 |
| weka.classifiers.functions.SMOreg | WW3Last3DaysStrategy | 0,70 | 0,48 | 0,55 | 0,57 |
| weka.classifiers.functions.SMOreg | WW3Last4DaysStrategy | 0,69 | 0,49 | 0,55 | 0,58 |
| weka.classifiers.trees.M5P | WW3Last2LecturesStrategy | 0,72 | 0,47 | 0,72 | 0,48 |
| weka.classifiers.trees.M5P | WW3Last3LecturesStrategy | 0,70 | 0,50 | 0,69 | 0,49 |
| weka.classifiers.trees.M5P | WW3Last6LecturesStrategy | 0,69 | 0,48 | 0,70 | 0,50 |
| weka.classifiers.trees.M5P | WW3Last8LecturesStrategy | 0,70 | 0,51 | 0,70 | 0,49 |
| weka.classifiers.trees.M5P | WW3SimpleStrategy | 0,74 | 0,47 | 0,74 | 0,46 |
| weka.classifiers.trees.M5P | TwoGridPointStrategy | 0,73 | 0,47 | 0,74 | 0,47 |
| weka.classifiers.trees.M5P | WW3Last2DaysStrategy | 0,72 | 0,47 | 0,72 | 0,48 |
| weka.classifiers.trees.M5P | WW3Last3DaysStrategy | 0,72 | 0,49 | 0,71 | 0,51 |
| weka.classifiers.trees.M5P | WW3Last4DaysStrategy | 0,71 | 0,49 | 0,71 | 0,50 |

Tabla 4.5: Resultados generales de pruebas sobre la ola “Windward”

Nuevamente para la ola Windward el mejor clasificador son las Maquinas de soporte Vectorial y mejor modelo de instancias es WW3Last3LecturesStrategy, aunque en este caso también se obtuvieron los mismos resultados con un Árbol de Regresión y la estrategia WW3SimpleStrategy.

### 4.3.5 Resultados generales en West Shore

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Optimizados** | | **No Optimizados** | |
| **Clasificador** | **Estrategia** | **Correl.** | **MAE** | **Correl.** | **MAE** |
| weka.classifiers.functions.LinearRegression | WW3Last2LecturesStrategy | 0,72 | 0,74 | 0,72 | 0,74 |
| weka.classifiers.functions.LinearRegression | WW3Last3LecturesStrategy | 0,78 | 0,67 | 0,79 | 0,67 |
| weka.classifiers.functions.LinearRegression | WW3Last6LecturesStrategy | 0,78 | 0,67 | 0,78 | 0,67 |
| weka.classifiers.functions.LinearRegression | WW3Last8LecturesStrategy | 0,78 | 0,68 | 0,78 | 0,67 |
| weka.classifiers.functions.LinearRegression | WW3SimpleStrategy | 0,75 | 0,71 | 0,75 | 0,71 |
| weka.classifiers.functions.LinearRegression | TwoGridPointStrategy | 0,76 | 0,71 | 0,75 | 0,72 |
| weka.classifiers.functions.LinearRegression | WW3Last2DaysStrategy | 0,78 | 0,68 | 0,78 | 0,68 |
| weka.classifiers.functions.LinearRegression | WW3Last3DaysStrategy | 0,78 | 0,68 | 0,77 | 0,68 |
| weka.classifiers.functions.LinearRegression | WW3Last4DaysStrategy | 0,77 | 0,68 | 0,77 | 0,68 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last2LecturesStrategy | 0,81 | 0,66 | 0,82 | 0,77 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last3LecturesStrategy | 0,81 | 0,63 | 0,81 | 0,68 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last6LecturesStrategy | 0,81 | 0,65 | 0,77 | 0,74 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last8LecturesStrategy | 0,79 | 0,67 | 0,76 | 0,72 |
| weka.classifiers.functions.MultilayerPerceptron | WW3SimpleStrategy | 0,82 | 0,62 | 0,81 | 0,71 |
| weka.classifiers.functions.MultilayerPerceptron | TwoGridPointStrategy | 0,82 | 0,62 | 0,82 | 0,71 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last2DaysStrategy | 0,82 | 0,62 | 0,82 | 0,75 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last3DaysStrategy | 0,82 | 0,62 | 0,80 | 0,71 |
| weka.classifiers.functions.MultilayerPerceptron | WW3Last4DaysStrategy | 0,81 | 0,63 | 0,77 | 0,80 |
| weka.classifiers.functions.SMOreg | WW3Last2LecturesStrategy | 0,82 | 0,61 | 0,78 | 0,68 |
| weka.classifiers.functions.SMOreg | WW3Last3LecturesStrategy | 0,82 | 0,61 | 0,78 | 0,66 |
| weka.classifiers.functions.SMOreg | WW3Last6LecturesStrategy | 0,81 | 0,65 | 0,78 | 0,67 |
| weka.classifiers.functions.SMOreg | WW3Last8LecturesStrategy | 0,79 | 0,66 | 0,78 | 0,66 |
| weka.classifiers.functions.SMOreg | WW3SimpleStrategy | 0,82 | 0,62 | 0,75 | 0,70 |
| weka.classifiers.functions.SMOreg | TwoGridPointStrategy | 0,82 | 0,62 | 0,75 | 0,71 |
| weka.classifiers.functions.SMOreg | WW3Last2DaysStrategy | 0,83 | 0,60 | 0,77 | 0,67 |
| weka.classifiers.functions.SMOreg | WW3Last3DaysStrategy | 0,82 | 0,61 | 0,77 | 0,68 |
| weka.classifiers.functions.SMOreg | WW3Last4DaysStrategy | 0,82 | 0,63 | 0,77 | 0,67 |
| weka.classifiers.trees.M5P | WW3Last2LecturesStrategy | 0,82 | 0,62 | 0,82 | 0,63 |
| weka.classifiers.trees.M5P | WW3Last3LecturesStrategy | 0,82 | 0,61 | 0,82 | 0,62 |
| weka.classifiers.trees.M5P | WW3Last6LecturesStrategy | 0,81 | 0,63 | 0,81 | 0,64 |
| weka.classifiers.trees.M5P | WW3Last8LecturesStrategy | 0,80 | 0,64 | 0,80 | 0,63 |
| weka.classifiers.trees.M5P | WW3SimpleStrategy | 0,81 | 0,63 | 0,82 | 0,62 |
| weka.classifiers.trees.M5P | TwoGridPointStrategy | 0,82 | 0,64 | 0,81 | 0,63 |
| weka.classifiers.trees.M5P | WW3Last2DaysStrategy | 0,82 | 0,62 | 0,82 | 0,62 |
| weka.classifiers.trees.M5P | WW3Last3DaysStrategy | 0,82 | 0,63 | 0,81 | 0,63 |
| weka.classifiers.trees.M5P | WW3Last4DaysStrategy | 0,81 | 0,63 | 0,81 | 0,63 |

Tabla 4.6: Resultados generales de pruebas sobre la ola “West Shore”

En este último caso de estudio, sobre la ola West Shore en Oahu, vemos que el mejor clasificador/estrategia fue SVM/WW3Last2DaysStrategy, y el segundo mejor resultado se obtuvo nuevamente con SVM/WW3Last3LecturesStrategy, y la diferencia entre ambos indicadores es ínfima.

## 4.4 Algoritmo de aprendizaje y modelo de instancia seleccionado

En base a los resultados obtenidos y analizados en las tablas 4.2, 4.3, 4.4, 4.5 y 4.6 se selecciono un algoritmo de aprendizaje de máquina y un modelo de instancia que maximizan la performance de los clasificadores que se generan a partir de estos. El algoritmo y modelo resultante de este análisis serán los utilizados en la implementación del proyecto para la generación de los pronosticadores entrenados capaces de predecir olas.

A simple vista podemos visualizar en las tablas 4.2, 4.3 y 4.5 que la diferencia entre los indicadores estadísticos retornados por los algoritmos no lineales (Arboles de regresión, SVM y Redes neuronales) y el de regresión lineal es grande respecto de la misma diferencia en las tablas 4.4 y 4.6. Una diferencia pequeña en los indicadores resultantes del entrenamiento de estos dos grupos de clasificadores se suele dar cuando las lecturas en altamar (off-shore) son similares a las observaciones visuales costeras (on-shore), en estos casos no hay mucho por predecir y tanto los algoritmos de regresión lineal y los no lineales brindan una performance similar. Caso contrario, las diferencias entre ambos tipos de clasificadores se hacen muy notorias siendo más efectivos los de aprendizaje de maquina (en el caso de estudio, las olas Windward, Ala Moana y Diamond head).

En las cinco tablas mencionadas antes se observa que el algoritmo “Maquinas de soporte vectorial” obtiene los mejores resultados en las cuatro de las cinco olas evaluadas en el caso de estudio para Oahu – Hawái. En la tabla 4.5 para la ola “Windward” vemos que el algoritmo ganador son los Arboles de Regresión y que la Maquina de soporte vectorial obtiene el segundo puesto, y a su vez que la diferencia entre los indicadores estadísticos de ambos algoritmos es ínfima. Por lo tanto el algoritmo a utilizar en el posterior desarrollo del sistema será las **Maquinas de Soporte Vectorial**, dado que para las cinco olas estudiadas ha dado los mejores resultados.

En cuanto al modelo de instancia a seleccionar, se puede observar que en tres de las cinco olas estudiadas WW3Last3LecturesStrategy es el que mejores resultados genera (ver tablas 4.2, 4.3 y 4.6). En la tabla 4.6 vemos que retorna los mismos indicadores que el modelo WW3SimpleStrategy. Al mismo tiempo, en base a los resultados mostrados en las tablas 4.4 y 4.6 WW3Last2DaysStrategy es el modelo que retorna los mejores resultados, pero en ambas WW3Last3LecturesStrategy es el que lo sigue en segundo lugar, y la diferencia entre los indicadores es verdaderamente pequeña. Por lo tanto el modelo de instancia seleccionado para la implementación será **WW3Last3LecturesStrategy**.

A continuación se detallará las pruebas realizadas para las cinco olas correspondientes al caso de estudio alrededor de la isla Oahu – Hawái, en base al algoritmo de aprendizaje y modelo de instancia seleccionado previamente.

# Capitulo 5 - Desarrollo de la aplicación

## 5.1 Descripción de la aplicación

El sistema desarrollado “**Surf-Forecaster**”, es una aplicación interactiva que ofrece la posibilidad de obtener pronósticos detallados del oleaje, en cualquier lugar del planeta. Basta con crear una ola (Ver sección 5.3.3 - Nueva ola) en cualquier lugar del mundo para comenzar a recibir pronósticos detallados con 180hs (una semana aprox.) de anticipación, en incrementos de tres horas, para la ola creada.

Adicionalmente el sistema cuenta con una sección de comparaciones entre diferentes olas, la cual brinda una herramienta muy práctica para ayudar al usuario a reconocer fácilmente en qué lugares romperán las mejores olas del día con una semana de anticipación, evitándole la necesidad, por ejemplo, de tener que recorrer toda la costa de su ciudad, diariamente, en busca de tales olas.

Surf-Forecaster ofrece además la posibilidad de registrase en el sistema (ver sección 5.2 – Tipos de usuario). Al ser un usuario registrado el sistema le permitirá dar de alta sus olas preferidas, generar comparaciones entre ellas y persistirlas, para no tener que volver a crearlas cada vez que quiere comparar las mismas olas.

### 5.1.1 Que diferencia a Surf-Forecaster de los demás sistemas de pronósticos ya existentes?

En general los sistemas de pronósticos existentes logran predecir olas a partir de la resolución de un modelo matemático global sumamente complejo. Este provee la predicción de la altura de las olas, la dirección de las mismas y el período (tiempo entre una ola y la siguiente). Si bien este pronóstico tiene buen rendimiento en alta mar, no sucede lo mismo en la cercanía de la costa. Esto se debe a que la interacción de las olas con el lecho marino cercano no es contemplada por estos modelos. De acuerdo a la geografía de la costa puede suceder que en un rango de 15 km un lugar presente condiciones apropiadas para la práctica del deporte mientras que en otro las condiciones sean desfavorables. Por esta razón es que la utilización de un pronóstico preciso en las cercanías de la costa es de suma utilidad y no es un servicio que los sistemas existentes logren brindar.

Para lograr esto Surf-Forecaster utiliza modelos de aprendizaje de máquina para aprender de casos pasados como un determinado pronóstico de alta mar se vio reflejado en las cercanías de la costa. De esta manera ante un nuevo pronóstico en alta mar nuestro sistema podrá predecir cuál será el efecto del mismo en las cercanías de la costa. El sistema se centra en la predicción de la altura de la ola exactamente en la ubicación de la ola que le interesa al usuario, mejorando así la predicción del pronóstico de alta mar.

### 5.1.2 Como logra pronosticar Surf-Forecaster?

Surf-Forecaster como muchos otros pronosticadores de olas y clima, utiliza el modelo WAVEWATCH III (ver sección 3.3 - Modelo WAVEWATCH III), como fuente primaria de obtención pronósticos.

WAVEWATCH III es un modelo representado por una grilla que abarca a todo el planeta. Mediante métodos numéricos complejos el modelo genera, a partir de cada predicción de viento, el pronóstico correspondiente en cada punto o intersección de la grilla. Estos puntos o intersecciones serán nombrados varias veces en el sistema y el resto del capítulo como “**GridPoint**” o “**WW3Gridpoint**”.

WAVEWATCH III se corre 4 veces al día, cada 6 horas, donde cada corrida produce pronósticos en incrementos de 3 horas desde la hora actual hasta 180 horas a futuro (7 días y medio), Es decir que genera alrededor de 60 pronósticos para cada punto de la grilla cada 6 horas.

Surf-Forecaster muestra al usuario la siguiente información por cada uno de los 60 pronósticos generados para un GridPoint específico:

* Dirección del viento.
* Velocidad del viento (km/h).
* Dirección de ola.
* Altura de ola (mts.).
* Período de ola (seg.).

Cuando se da de alta una nueva ola en Surf-Forecaster (ver sección 5.3.3 – Nueva ola), se le solicita al usuario que seleccione la ubicación exacta (latitud y longitud) de esa ola. A partir de esa ubicación, el sistema averigua y muestra por pantalla cuales son los GridPoints más cercanos que la rodean (de no existir ninguno cercano a esa ubicación, se le comunica al usuario que esa ola no podrá ser pronosticada). El usuario deberá seleccionar luego cual de los GridPoints ofrecidos desea utilizar como fuente de pronósticos (cuya latitud y longitud posiblemente no sean exactamente las mismas que la de ola creada). Una vez completado estos pasos y llenado el formulario correspondiente para dar de alta una nueva ola, el usuario podrá ver en la sección de pronósticos, cuales son las predicciones para la ola recientemente creada. Debe tener en cuenta que estas predicciones corresponden a pronósticos en el GridPoint y no exactamente donde la ola fue ubicada por el usuario. Hasta aquí lo que obtenemos es el mismo resultado que cualquier otro sistema de pronósticos existente.

Surf-Forecaster, a diferencia de otros pronosticadores de olas, soluciona este inconveniente. A partir de información adicional que puede suministrar opcionalmente el creador de la ola, el sistema logra predecir la altura de las olas, con 180hs de anticipación, exactamente en las coordenadas donde la ola fue creada. En este caso el sistema utiliza esta información para entrenar un **clasificador de olas especializado** (o pronosticador especializado). Para esto es necesario proveer al sistema de un archivo con información histórica de observaciones visuales de la altura de las olas en esa ubicación indexadas por fecha. Cada observación debe indicar la fecha de la observación (aaaa/mm/dd) y la altura de la ola más grande (en metros) que rompió ese día (ver sección 5.3.3 – Nueva ola, para conocer el formato del archivo de observaciones visuales requerido). Como mínimo se necesita un conjunto de **50 observaciones** de acuerdo a lo estimado en nuestra experimentación. Esta información adicional puede ser actualizada constantemente por el usuario a medida que pasa el tiempo para ir mejorando el rendimiento del clasificador. A medida que recibe más información, dispone de más datos para entrenarse. Al mismo tiempo, si las condiciones geográficas de la ola varían (como puede ser el suelo), datos actualizados mantendrán más actualizado al clasificador.

Si este archivo de información histórica es suministrado al sistema para la ola creada Surf-Forecaster generará y mostrará por pantalla dos valores diferentes con la altura de las olas por cada pronóstico (ver sección 5.3.1 - Pronóstico). La primera corresponde a la lectura en el GridPoint, ubicación cercana a la ola pronosticada. La segunda lectura corresponde a la altura de la ola **exactamente** en el punto donde la ola fue localizada por el usuario a la hora de crearla, por lo que este será seguramente el pronóstico que al usuario más le interese. Además de estos dos pronósticos de altura de ola, también se mostrarán los datos listados anteriormente que corresponderán a la localización del GridPoint seleccionado, pero dado que el GridPoint se encuentra muy cerca de la ola, y que la geografía del lugar no afecta a los datos restantes (solo a la altura de las olas), estos valores serán muy similares tanto en el GridPoint como en la localización de la ola.

De esta manera Surf-Forecaster puede pronosticar, si se le suministra la información necesaria, la altura de las olas en las diferentes costas que presenta la geografía de cualquier localidad del mundo, evitándole al interesado recorrer varios kilómetros buscando la playa con las mejores condiciones del oleaje para realizar su actividad.

## 5.2 Tipos de usuario

Surf-Forecaster distingue entre tres tipos de usuarios con roles y permisos diferenciados. De acuerdo con el tipo de usuario con el que se ingresa al sistema, se puede acceder a diferentes secciones, así como también dar de alta, baja o modificar información del mismo.

La siguiente lista muestra los tres tipos de usuario que el sistema reconoce actualmente:

* Usuario no registrado u anónimo.
* Usuario registrado.
* Administrador.

### 5.2.1 Usuario no registrado u anónimo

No es necesario registrarse en el sistema para poder operar con él. Surf-Forecaster permite acceder a pronósticos detallados de olas o a la sección de comparaciones entre diferentes olas siendo un usuario anónimo. Dado que este usuario no está registrado en el sistema, y no dispone de un código de identificación único, el mismo no podrá persistir ningún dato dentro de la aplicación, es decir, no tiene acceso a las secciones para dar de alta/editar/borrar olas, como tampoco alta/baja/modificación de comparaciones entre olas. Por otra parte dado que una ola en el sistema se puede declarar privada o pública, este usuario solo tendrá acceso a los pronósticos y comparaciones de olas públicas registradas por otros usuarios del sistema que tienen la intención de compartir sus olas preferidas.

### 5.2.2 Usuario registrado

Un usuario registrado, puede principalmente comenzar a interactuar con la aplicación, pudiendo realizar operaciones de alta/baja/modificación de olas y comparaciones. Este usuario no solo tiene acceso a pronósticos de olas creadas por él, sino también olas creadas y declaradas como publicas por otros usuarios. Puede además entrenar un clasificador especializado para una ola específica y/o declarar sus olas como privadas o públicas con el fin de compartir o no los pronósticos en esa ubicación con el resto de la comunidad interesada.

### 5.2.3 Administrador

Este usuario tiene acceso total a todas las funcionalidades del sistema, podría operar de la misma manera que cualquier “Usuario registrado”, pero a su vez tiene privilegios sobre toda la información persistida en el mismo, pudiendo dar de alta, baja o modificar cualquier dato persistido ya sea información de usuarios, olas, comparaciones, países, áreas, zonas, etc. Es el único tipo de usuario con acceso a un panel de backend el cual permite realizar estas operaciones privilegiadas. Un usuario con estos permisos privilegiados deberá ser otorgado a una persona de confianza ya que dispone de las herramientas necesarias para controlar todo el sistema. Surf-Forecaster no permite crear un usuario con estos privilegios desde el frontend, solo otro usuario administrador podría crearlo desde el panel de backend (herramienta privada y únicamente administrativa).

## 5.3 Secciones de la aplicación

### 5.3.1 Pronóstico

En esta sección del sistema, la cual es la pantalla principal del mismo, está dedicada para que el usuario pueda visualizar el pronóstico de cualquier ola a la que tenga acceso, ya sean las suyas y las públicas si es que es un usuario registrado, o solo las olas públicas si es un usuario anónimo.

Las olas en esta sección están organizadas en cuatro listas con diferente granularidad.

Fig. 5.1: Panel de localización de olas

* **Área**: Divide al planeta en grandes bloques, como pueden ser continentes o subdivisiones de continentes (América del Norte, Oceanía,…).
* **País**: Muestra una lista de países que pertenecen al área seleccionada anteriormente
* **Zona**: Divide al país seleccionado en zonas más pequeñas, estas zonas son creadas por los usuarios cuando dan de alta sus olas (Ver sección 5.3.3 - Nueva Ola) y pueden contener una o varias olas.
* **Ola**: Finalmente aquí selecciona la ola que quiere pronosticar perteneciente a la zona seleccionada anteriormente.

Pulsando el botón “Ver pronóstico”, el sistema recupera las próximas 180hs de pronósticos para la ola seleccionada y las mostrará en pantalla de la siguiente manera:

En la parte superior vemos la siguiente leyenda que muestra algunos datos de la ola seleccionada

Fig. 5.2: Información de ola seleccionada

Como título se detalla la zona seguida de la ola pronosticada con el formato Zona > 0la.

Debajo se muestran los datos específicos de la ola elegida:

* **Ola Lat.:** Latitud de la ola.
* **Ola Lon.:** Longitud de la ola.
* **WW3GridPoint Lat.:** Latitud del GridPoint (punto de grilla del modelo WAVEWATCH III) seleccionado como fuente de pronósticos para esa ola.
* **WW3GridPoint Lon.:** Longitud del GridPoint (punto de grilla del modelo WAVEWATCH III) seleccionado como fuente de pronósticos para esa ola.
* **Zona horaria:** La zona horaria seleccionada para esta ola. Los pronósticos generados detallan la fecha y hora del mismo, que se corresponden con la zona horaria de la ola.

Todos los datos listados previamente son solicitados al usuario cuando da de alta una nueva ola en el sistema (Ver sección 5.3.3 – Nueva ola).

Debajo de la sección de información de la ola se muestran las siguientes tablas:

**Ahora**

Muestra las condiciones del oleaje en este preciso instante para la ola seleccionada.



Fig. 5.3: Condiciones actuales de la ola seleccionada.

La figura anterior muestra las condiciones actuales de la ola seleccionada. La tabla muestra todos los datos que se listaron anteriormente pertenecientes a un pronóstico específico. Todos los datos corresponden a la ubicación del GridPoint seleccionado, excepto la altura de la ola, la cual mostrará la del GridPoint si esta ola nunca fue entrenada con observaciones visuales o caso contrario mostrará la altura de la ola exactamente en donde esta rompe.

**+3 horas**

Muestra las condiciones del oleaje a partir de las próximas tres horas. Permitiendo al usuario saber cómo va a fluctuar las olas en las siguientes horas.



Fig. 5.4: Condiciones a partir de las próximas 3 horas de la ola seleccionada.

Observando las figuras 5.3 y 5.4 anteriores el usuario puede notar que la ola pronosticada actualmente mide 6 metros y en las próximas 3 horas su tamaño disminuirá 70 cm aproximadamente, pero en el caso del ejemplo las condiciones del viento, dirección de ola y período se mantienen constantes.

**Tabla de pronósticos detallada**

Finalmente, debajo de las tablas vistas anteriormente (figura 5.3 y 5.4), vemos un pronóstico completo (180hs) en una tabla detallada. Si a la ola seleccionada se la entrenó con clasificador de olas especializado (usando un archivo histórico de observaciones visuales en ese lugar), se verá en la tabla dos filas indicando la altura de la ola. La primera fila indica la altura en el GridPoint seleccionado (generalmente a kilómetros del punto donde la ola rompe), la segunda fila que indica altura de ola, es una predicción localizada exactamente en donde se encuentra la ola seleccionada.

A continuación se muestra un ejemplo de un fragmento de la tabla de pronósticos detallados para la ola Pipeline perteneciente a la isla Oahu en Hawái. Dicha ola fue entrenada con un clasificador especializado por lo que presenta como se dijo anteriormente una fila que detalla la altura de las olas en la posición del GridPoint y otra detallando la altura de las olas en la ubicación exacta de la ola.

Fig. 5.5: Fragmento de tabla de pronósticos detallados para la ola Pipeline en Oahu – Hawái.

Como se puede ver en la figura 5.5 (fragmento de tabla) cada una de las columnas de la tabla corresponde a cada uno de los 60 pronósticos que WAVEWATCH III genera cada 6 horas. Si se miran las horas de cada pronóstico, se ve que avanzan en incrementos de 3 horas como se explico anteriormente (ver sección 5.1.2 – Descripción de la aplicación - Como logra pronosticar Surf-Forecaster?).

Por otra parte, las filas de la tabla están organizadas en dos grupos. Cada grupo (separados por una fila en gris claro) corresponde a los pronosticadores que dicha ola tiene asignados. El primero, llamado “WW3 NOAA Forecaster” corresponde a los pronósticos generados por WAVEWATCH III en la ubicación del GridPoint seleccionado para esa ola. El segundo “Machine Learning based Forecaster” corresponde al generado luego de entrenar un clasificador gracias al suministro de observaciones visuales costeras en la ubicación exacta de esa ola. Este clasificador especializado solo es capaz de calcular la altura de las olas, siendo el resto de los datos muy similares en ambas ubicaciones.

Adicionalmente, y solo con el fin de enriquecer la interfaz de usuario y facilitar la lectura de los datos, utilizamos iconos de diferentes colores y tamaños que representan el tamaño de las olas, y de la intensidad del viento. Además clickeando sobre el símbolo (-) al lado del nombre del pronosticador de WW3 NOAA Forecaster, podemos encoger dicha tabla, lo cual facilita la comparación entre las predicciones de cada pronosticador. Posándose sobre el icono de la ola del pronosticador WW3 aparece una ventana popup indicando el resto de los datos ocultos. Esta vista se puede apreciar en la siguiente figura.

 Fig. 5.6: Fragmento de tabla de pronósticos detallados minimizada.

### 5.3.2 Comparación de olas

Surf-Forecaster presenta una sección de comparación de olas, mediante gráficos de barras y tablas comparativas lo cual supone una herramienta interesante para el usuario a la hora de evaluar cual de sus olas favoritas presentara las mejores condiciones en este preciso instante o dentro de algunas horas, días, etc.

Esta sección es accesible por todos los tipos de usuario reconocidos por el sistema, presentando más o menos herramientas de acuerdo con los permisos del usuario que está operando. A continuación se mostrarán las pantallas tal cual las ve cada tipo de usuario del sistema según sus permisos.

**5.3.2.1 Usuario no registrado**

Fig. 5.7: Vista de pantalla de generación de comparaciones – Usuario no registrado.

La figura 5.7 muestra la pantalla de generación de comparaciones que podrá visualizar un usuario no registrado en el sistema. Solo dispone de herramientas para generar una comparación, buscar las olas a comparar, ordenar las olas que desea comparar y finalmente visualizar la comparación presionando el botón “Comparar”.

**Generar una comparación:**

1. Utilizar los links de localización de olas que aparecen en la parte superior (Área > País > Zona) para encontrar las olas que desea comparar.



Fig. 5.8: Menú de localización de olas en pantalla de generación de comparaciones

Nota: Las olas a comparar no tienen que ser necesariamente de la misma localización, es decir, se pueden ir agregando olas de diferentes zonas, países y áreas al listado de olas a comparar.

1. Utilizar los listados debajo de los links de localización de olas para seleccionar que olas se desean comparar, para agregar olas a la comparación, seleccionar una o más olas del listado de la izquierda y presionar .Para quitar olas de la comparación, seleccionar una o más olas del listado derecho y presionar .
2. Utilizar los botones a la derecha (Primero, Subir, Bajar, Último) para organizar el orden de las olas en que se desea visualizar la comparación.
3. Finalmente presionar el botón “Comparar".

**5.3.2.2 Usuario registrado u Administrador**

Fig. 5.9: de pantalla de generación de comparaciones – Usuario registrado u Administrador.

A diferencia de la pantalla de generación de comparaciones para un usuario anónimo (Fig. 5.7), en esta pantalla el usuario registrado dispone de las herramientas necesarias para crear, editar, guardar y borrar sus comparaciones preferidas. De esta manera si frecuentemente tiene que realizar la misma comparación, basta con persistirla, para luego solo seleccionarla de la lista “Mis comparaciones” y presionar el botón “Comparar”.

**Generar una comparación** (Idéntico a generar una comparación explicado previamente para usuarios no registrados).

**Guardar una comparación**

Una vez generada la comparación deseada, presionar “Guardar…”. Automáticamente, debajo de dicho botón aparecerá un formulario para completar el nombre (obligatorio) y descripción de la comparación.

Fig. 5.10: Guardar comparación.

Finalmente presionar el botón “Grabar” para persistir la comparación definitivamente o “Cancelar” para deshacer la operación.

**Editar una comparación**

Esta operación es similar a la de Guardar.

1. Seleccionar una comparación del listado “Mis comparaciones”.
2. Utilizar las herramientas para generación de comparaciones para agregar, quitar o cambiar el orden de aparición de las olas.
3. Presionar “Guardar…” para visualizar el panel “Guardar comparación”. Si lo que queremos es editar, dejar el nombre como está, y editar la descripción si lo desea.
4. Presionar “Grabar”. Aparecerá un mensaje para confirmar la operación:



Fig. 5.11: Mensaje de confirmación para sobrescribir una comparación.

1. Presionar “Aceptar” para asentar la edición, o “Cancelar” para deshacer.

**Borrar una comparación**

1. Seleccionar la comparación deseada de la lista “Mis comparaciones”.
2. Presionar el botón “Borrar” a la derecha de dicha lista. Aparecerá un mensaje para confirmar la operación:



Fig. 5.12: Mensaje de confirmación para eliminar una comparación.

1. Finalmente presionar “Aceptar” para concretar la operación, o “Cancelar” para deshacerla.

**Ver comparación generada**

Una vez generada o seleccionada de la lista la comparación, y luego de presionar el botón “Comparar” aparecerá en pantalla la vista de resultados de comparación de las olas seleccionadas. Esta vista es idéntica para todos los tipos de usuarios del sistema. Se compone de cuatro partes:

1. Selección de pronosticadores de cada ola que integra la comparación generada.

Fig. 5.13: Selección del pronosticador deseado para cada ola.

Dado que las olas pueden tener más de un pronosticador (Ver sección 5.3.1 - Pronóstico) si se suministró información de observaciones visuales de la ola, uno en la ubicación del GridPoint (WW3 Noaa Forecaster) y otro en la ubicación de la ola misma (Machine Learning based Forecaster), a la hora de querer visualizar una comparación entre la altura de distintas olas, Surf-Forecaster permite seleccionar que pronosticador queremos utilizar para cada una en la comparación. Para esto basta con elegir de cada lista el pronosticador deseado y presionar “Actualizar” para regenerar la comparación.

La lista de pronosticadores de cada ola solo mostrará más de uno si la ola en cuestión ha sido previamente entrenada con un pronosticador especializado, caso contrario solo aparecerá “WW3 Noaa Forecaster”.

1. Gráfico de barras que muestra la altura de las distintas olas comparadas durante las próximas 12 horas. Los datos de cada ola son pronosticados con el pronosticador seleccionado previamente.



Fig. 5.14: Altura de las olas en las próximas horas.

1. Gráfico animado que permite visualizar como irán variando las alturas de las olas comparadas desde este momento hasta la semana próxima (180 hs). Los datos de cada ola son pronosticados con el pronosticador seleccionado previamente.

Fig. 5.15: Captura de animación con la altura de cada ola a lo largo de las 180hs de pronóstico. Vista de gráfico de barras.

El mismo gráfico puede ser visualizado en un formato de grafico de líneas, como se muestra a continuación.

 Fig. 5.16: Gráfico con la altura de cada ola a lo largo de las 180hs de pronóstico. Vista de gráfico de líneas.

Estas últimas dos imágenes (Fig. 5.15 y Fig. 5.16), muestran una herramienta muy práctica para poder visualizar como irán oscilando las olas a lo largo de toda la semana, dándole al usuario la posibilidad de organizar su calendario de acuerdo al comportamiento del océano.

1. Finalmente puede visualizar la Tabla de pronósticos detallado, una tabla similar a la vista en la sección 5.3.1 – Pronostico, pero en su formato encogido, donde cada fila corresponde a cada una de las olas seleccionadas para la comparación y solo se muestra la altura de la ola. Si se desea ver el resto de los datos, basta con situarse sobre el icono de la ola. En la siguiente figura vemos un fragmento de dicha tabla.

Fig. 5.17: Fragmento de tabla de pronósticos detallado en sección de comparaciones.

Vemos adicionalmente en la figura 5.17, que cada fila tiene como titulo el nombre de la ola que representa seguido del nombre del pronosticador utilizado.

### 5.3.3 Nueva ola

Surf-Forecaster permite a cualquier **usuario registrado** (no es accesible para usuarios anónimos) en el sistema obtener pronósticos de olas en cualquier punto del planeta. El medio para permitirle al usuario obtener esta información es la sección “Nueva ola”.

Esta sección permite generar olas donde sea, opcionalmente entrenarla con un pronosticador especializado si es que el usuario lo desea y dispone de suficientes observaciones visuales para hacerlo (50 días como mínimo, 100 mínimo recomendado) y adicionalmente permite declarar las olas creadas como públicas o privadas si es que desea compartirla con cualquier otro usuario de Surf-Forecaster (registrado u anónimo) o no respectivamente.

Los pasos y datos necesarios para dar de alta una nueva ola en el sistema son sencillos y no son demasiados. A continuación se presentan estos acompañados por algunas capturas de pantalla. Dar de alta una ola se podría dividir en tres etapas sencillas:

1. Cargar información básica de la ola (área, país, zona, nombre de ola, …). Este paso es **obligatorio**.

Fig. 5.18: Nueva ola, información básica

La figura anterior muestra los datos básicos a cargar obligatoriamente al dar de alta una nueva ola.

* **Área:** Seleccionar un área de las disponibles en el sistema.
* **País**: Seleccionar un país de los disponibles en el sistema para el área seleccionada.
* **Zona**: Seleccionar una zona de las disponibles en el sistema para el usuario registrado actualmente, del país seleccionado. Si ninguna de las zonas disponibles son válidas para ubicar la nueva ola, se puede crear una nueva pulsando el botón “Crear una zona”, lo cual mostrará un campo de texto para ingresar un nuevo nombre de zona. La zona creada pertenecerá al país seleccionado previamente.
* **Ola**: En este campo se debe ingresar el nombre de ola deseado para la ola que se va a dar de alta.
* **Zona horaria**: La zona horaria seleccionada será utilizada para mostrar los pronósticos en la sección de pronósticos. Cada vez que se solicite visualizar los pronósticos de esta ola, se le aplicará la conversión de horas necesarias para mostrar los datos en el Timezone seleccionado.
* **Visibilidad de ola**: Definir si la ola será visible para todos los usuarios del sistema registrados u anónimos (PUBLICA) o solo para el usuario que la creo (PRIVADA).

1. Ubicar la ola en el mapa que se muestra en la pantalla, y seleccionar el GridPoint del cual se desean obtener pronósticos. Este paso es **obligatorio**.

Fig. 5.19: Nueva ola, ubicarla en el mapa.

La pantalla anterior muestra el mapa en el que el usuario debe localizar la ola que desea crear. Con un simple click en la pantalla, aparecerá en pantalla un icono de una ola, se completaran los datos de ola (latitud y longitud) en el formulario de la derecha, y aparecerán en pantalla los GridPoints disponibles alrededor de dicha ubicación (estos se presentan como pelotitas rojas). El siguiente paso es elegir alguno de ellos.

Una vez seleccionado el GridPoint dicho icono (cliqueando sobre alguno de los disponibles) este se pondrá de color verde y se llenarán automáticamente los datos correspondientes al Pronosticador (WW3Gridpoint) del formulario de la derecha. En la siguiente figura 5.20 se ve un ejemplo en la que el usuario selecciono una ola en la costa norte de Oahu – Hawái y a su vez seleccionó el GridPoint superior izquierdo dado que es el que esta inmediatamente enfrente de la localización de la ola.

 Fig. 5.20: Nueva ola, ya ubicada en el mapa.

Es obligatorio para poder completar el alta de una ola que los formularios a la derecha del mapa (Ola y Pronosticador) estén completados, y la única forma de hacerlo es cliqueando sobre el mapa, dado que dichos campos están bloqueados para llenarlos manualmente.

1. Suministrar un archivo de observaciones visuales para dicha ola con el fin de entrenar un pronosticador especializado. Este paso es **opcional**.



Fig. 5.21: Nueva ola, entrenar un pronosticador

La figura 5.21 muestra el formulario para completar si es que el usuario desea entrenar un pronosticador especializado para dicha ola. Para realizar esta operación el usuario debe suministrar a Surf-Forecaster un archivo con observaciones visuales de dicha ola.

* **Formato del archivo**: Este enlace muestra una guía de cómo debe formatearse el archivo de observaciones visuales, para que pueda ser correctamente analizado por el sistema.
* **Horas de luz solar**: Es necesario que el usuario agregue adicionalmente (solo en el caso que desee entrenar un pronosticador especializado) los horarios aproximados de luz solar en la ubicación de la ola.

El archivo de observaciones debe estar indexado por fecha y contener en cada línea la altura de la ola (METROS) más grande observada en dicha ubicación en el día especificado, como se puede ver en la imagen debajo, donde cada línea del archivo contiene cuatro columnas separadas por un ESPACIO que representan año, mes, día y altura de ola (METROS) respectivamente. Si la altura de ola contiene valores decimales como se ve en la imagen, utilizar el punto (.) como separador decimal. Se recomienda usar archivos de texto con formato TXT.



Fig. 5.22: Nueva ola, formato del archive de observaciones visuales.

Los descriptos anteriormente son los pasos y la información necesaria para dar de alta una nueva ola en el sistema. Finalmente pulsando sobre el botón “Grabar” se procede a dar de alta la nueva ola.

### 5.3.4 Mis olas

En Surf-Forecaster “Mis olas”, solo accesible por usuarios registrados, es la sección correspondiente para que cada usuario pueda administrar cada una de las olas que ha creado. Esta vista permite editar o borrar cualquier ola creada por el usuario logueado en el sistema.

En la parte superior de la pantalla se muestra una tabla detallando todas las olas creadas por el usuario, y cada una de estas presenta en las últimas dos columnas iconos cliqueables para editar o eliminar la ola seleccionada como se ve en la siguiente figura:

Fig. 5.23: Mis olas, tabla de olas pertenecientes al usuario actualmente logueado.

Si el usuario pulsa sobre el icono (borrar) aparece el cuadro de confirmación de eliminación de una ola (ver Fig. 5.24). Presionando “Aceptar” dicha ola será eliminada completamente del sistema junto con toda la información asociada a esta persistida como pronosticadores, observaciones visuales, etc.



Fig. 5.24: Confirmación de eliminación de la ola seleccionada.

Si el usuario pulsa sobre el icono (editar) aparece debajo de la tabla un formulario idéntico al de la sección “Nueva ola” (ver sección 5.3.3 – Nueva ola), con toda la información ya cargada con los datos de la ola seleccionada para su edición.

La única diferencia entre la ventana de edición de ola y “Nueva ola” se puede ver en el último paso del formulario (entrenamiento del pronosticador especializado) como muestra la Fig. 5.25:



Fig. 5.25: Mis olas – Edición de ola, pronosticador especializado.

Como se ve en la figura anterior, en esta sección, el formulario muestra una lista de los pronosticadores especializados con los que la ola ha sido entrenada previamente, si esta fue entrenada alguna vez, en la que cada ítem muestra:

* **Nombre del pronosticador**.
* **Valor de correlación** resultante del entrenamiento entre los datos suministrados y los pronósticos generador por WAVEWATCH III en el GridPoint de la ola.
* **Valor del error absoluto medio** resultante del entrenamiento, el cual indica cuanto se puede llegar a equivocar el pronosticador entre la altura predicha de la ola y la altura que realmente alcanzo la ola al momento de romper.

Adicionalmente se puede observar, debajo del campo para seleccionar el archivo de observaciones visuales, un control para decidir (en el caso que se intente actualizar la información de observaciones) si deseamos reemplazar todas las observaciones existentes por nuevas observaciones o si deseamos agregar más observaciones.

Mantener actualizadas las observaciones es recomendable para mantener actualizado al pronosticador especializado respecto de los posibles cambios de las condiciones geográficas de la ola, como puede ser por ejemplo cambios de la superficie del suelo en olas donde el fondo es de arena.

Finalmente presionando “Grabar”, los cambios realizados sobre la ola seleccionada son persistidos. En el caso que se hayan agregaron o reemplazado observaciones visuales, o si se cambio la ubicación de la ola o del GridPoint utilizado, se forzará a un reentrenamiento del pronosticador especializado que esta ola poseía, si es que poseía alguno.

# Capítulo 6 - Conclusiones

## 6.1 Resumen

El aprendizaje de maquina es una disciplina de la inteligencia artificial dedicada al desarrollo de algoritmos que permiten a la computadora aprender a partir de la experiencia. Dentro del área de aprendizaje de máquina el aprendizaje supervisado es una técnica que consiste en aprender una función a partir de un conjunto de datos de entrenamiento. El conjunto de entrenamiento consiste en pares de objetos de entrada y salida esperada, es decir cada par es una instancia y una clase a la que esa instancia pertenece. Una instancia representa un conjunto de observaciones sobre un hecho determinado. Una clase representa la clasificación que se le da al hecho. A partir de un conjunto de ejemplos y un algoritmo de aprendizaje supervisado podemos entrenar un clasificador para que el mismo pueda predecir cuál será la clase de una instancia que desconocemos.

El resultado del proceso de aprendizaje es un clasificador, el cual representa un concepto aprendido a partir de los ejemplos. El clasificador obtenido es resultado de un proceso inductivo y como tal su efectividad no está garantizada. Además de evaluar el clasificador con el sentido crítico del usuario es necesario establecer métricas claras del desempeño del mismo. Para esto se prueba el clasificador con instancias cuya clase se conoce previamente y se comparan la clase verdadera y la predicha.

Dentro del aprendizaje supervisado existen diferentes métodos de predicción, los cuales son más o menos eficientes de acuerdo a la naturaleza de los datos del problema en cuestión. Ejemplos de ellos son las maquinas de soporte vectorial, redes neuronales, árboles de decisión, regresión lineal, etc.

La herramienta WEKA contiene una colección de algoritmos para el análisis de datos y modelado predictivo para hacer experimentos de aprendizaje automático. Provee, entre otros, los métodos de predicción mencionados en el párrafo anterior, para experimentación e investigación. Dicha herramienta toma como entrada un conjunto de instancias (cada instancia se compone de un conjunto de variables predictoras y uno o más atributos de clase que representan las variables a predecir) y ejecuta el algoritmo correspondiente. El resultado de este proceso es un clasificador entrenado el cual es capaz de predecir el atributo de clase (variable a predecir) suministrándole únicamente las variables predictoras que componen una instancia. Esta herramienta fue utilizada en este proyecto para el entrenar los clasificadores que utiliza el sistema desarrollado para realizar predicciones.

## 6.2 Conclusiones del trabajo

En el proyecto presentado utilizamos el técnicas de aprendizaje supervisado para desarrollar un sistema capaz de pronosticar olas en las cercanías de la costa de cualquier lugar del mundo basándonos en lecturas de altamar provistas por un sistema de pronósticos global denominado WAVEWATCH III desarrollado por el NOAA (National Oceanic and Atmospheric Administration) y en observaciones visuales suministradas por un observador experto en la ubicación exacta de la que se espera obtener pronósticos. El resultado del estudio fue un sistema denominado “Surf-Forecaster” en el cual los usuarios pueden dar de alta sus olas preferidas en cualquier lugar y obtener pronósticos en dichas ubicaciones con hasta una semana de anticipación en incrementos de tres horas, siempre y cuando se le provea al sistema de un histórico de observaciones visuales tomadas en dicha ubicación.

El sistema cuenta con un clasificador entrenado generado con WEKA para poder realizar las predicciones en la ubicación deseada por el usuario. El algoritmo utilizado fueron las Maquinas de Soporte Vectorial debido a que fue el que mejor rendimiento demostró luego de varias etapas de evaluaciones y experimentaciones (Ver capitulo 4 - Evaluaciones).

Finalmente para evaluar el sistema desarrollado utilizamos información de diferentes playas ubicadas alrededor de la isla de Oahu en Hawái, más específicamente en las playas Sunset, Diamond Head, Makapu, Makaha y Ala Moana, de las cuales disponíamos de una base de datos histórica de observaciones visuales con más de seis años de registros con periodicidad diaria. Los resultados obtenidos fueron comparados, con muy buenos resultados, con los publicados en diferentes sitios web locales de la isla, orientados a deportes acuáticos como puede ser el surf, windsurf, kite-surf, etc. Estos sitios dan información al lector de las condiciones del mar para la práctica de estos deportes en distintas playas de Oahu. Con este enfoque logramos demostrar que el uso de algoritmos de aprendizaje de maquina son una herramienta efectiva y viable para realizar pronósticos de oleaje y del comportamiento del mar, mejorando también los resultados e información ofrecidos por modelos matemáticos desarrollados para el mismo fin.

## 6.3 Ventajas del enfoque

El sistema desarrollado, Surf-Forecaster, presenta un gran número de ventajas respecto de los sistemas de pronósticos de oleaje existentes actualmente:

* Se logro predecir la altura de las olas exactamente en la ubicación del observador, lo que le da al usuario mayor certeza de las condiciones del mar donde a este realmente le interesa, dado que otros pronosticadores dan esta información pero en una zona alejada de la costa ya sea a partir de mediciones de boyas en altamar o de mediciones brindadas por algún sistema como puede ser WAVEWATCH III en un GridPoint especifico.
* Con Surf-Forecaster llevamos la predicción a una granularidad de “Ola” a diferencia de los pronosticadores existentes más usados los cuales tienen una granularidad de “Localidad” o “Zona”, es decir que logramos diferenciar la predicción de la altura de dos olas que pueden romper por ejemplo a una distancia de no más 100 metros entre ambas. Aunque la distancia sea mínima, estas dos olas pueden ser totalmente diferentes dependiendo de la geografía de la zona, de la arquitectura de la playa o del fondo del mar entre otros parámetros. Tal vez una escollera puede separar una ola que rompe en mar abierto de una ola que rompe a tan solo 50 metros de esta última, pero esta se encuentra rodeada entre escolleras, estas condiciones hacen que la predicción de esta ola sea distinta a la que rompe en mar abierto. Con este nivel de granularidad trabaja Surf-Forecaster, permitiendo definir múltiples olas a lo largo de la costa de una localidad específica. Los pronosticadores comunes, generalmente obtienen la información de la boya más cercana en altamar o del GridPoint más cercano si estos utilizan un sistema como WAVEWATCH III y dan el mismo pronóstico para toda la costa de una localidad, lo cual es irreal.
* Surf-Forecaster es un sistema web muy sencillo de entender y utilizar. Si es un usuario registrado puede crear las olas que desee y si dispone de un histórico de observaciones visuales para esas olas podría entrenar varios clasificadores para comenzar a recibir pronósticos exactamente en esos lugares, si no dispone de esta historia, entonces solo recibirá los pronósticos de los GridPoint elegidos como fuente de medición en altamar, en este último caso el sistema se comportara como cualquier otro pronosticador existente basado en WAVEWATCH III.
* Si el usuario interesado depende de encontrar un lugar a lo largo de toda la costa de una localidad que presente las mejores condiciones de oleaje y vientos para desarrollar una actividad determinada, Surf-Forecaster le ahorra recorrer largas distancias en busca de tales condiciones, permitiéndole ir directamente al lugar de su interés.
* En lugares donde la población se encuentra ubicada cerca de la costa el sistema puede predecir posibles catástrofes con olas de gran tamaño, con hasta una semana de anticipación. Esta es una posible ventaja de utilización de un sistema de este tipo, aunque para detectar este tipo de catástrofes, el clasificador debe ser entrenado con datos excepcionales, ya que el aprendizaje es en base a experiencia de casos pasados, y estos hechos son poco frecuentes.
* Este enfoque hace que no necesitemos recolectar datos costosos de conseguir como la batimetría marina. La batimetría es la ciencia que mide las profundidades marinas para determinar la topografía del fondo del mar, actualmente las mediciones son realizadas por GPS diferencial para una posición exacta, y con sonares hidrográficos mono o multihaz para determinar la profundidad exacta. Recolectar esta información es sumamente costoso, requiere de equipamiento especial y de expertos en la materia para analizar los resultados y poder cargarlos una vez procesados en un sistema que haga uso de ellos para poder pronosticar el oleaje. Gracias al uso de algoritmos de aprendizaje de máquina para entrenar un clasificador, los datos suministrados por las boyas en altamar, mas las observaciones visuales costeras son suficientes para poder pronosticar olas en aguas poco profundas.
* Surf-Forecaster puede predecir olas sin importar el tipo de fondo que la geografía de esa costa presenta. Es indistinto para el sistema si este es de arena, piedra, arcilla, arrecife de corales, etc. Aunque el tipo de fondo no afecta a las predicciones, si el fondo no es de un material solido como la piedra o un arrecife, este puede requerir una actualización de las observaciones visuales cada determinado tiempo (dependiente de cuán rápido la actividad marina pueda remover el fondo cambiando su topografía).
* Es posible pronosticar olas en cualquier lugar del mundo. Si se dispone de un histórico suficiente de observaciones visuales en cualquier lugar se podrán obtener pronósticos exactamente en esa ubicación, caso contrario el sistema nos mostrará los pronósticos capturados por WAVEWATCH III de un GridPoint cercano a ese punto elegido por el usuario al momento de dar de alta la ola. Cualquiera sea el caso, se recibirán pronósticos con una semana de anticipación en incrementos de tres horas.
* El sistema provee una sección de comparaciones, lo cual puede resultarle útil al usuario a la hora de decidir en qué lugar va a practicar su actividad. En dicha sección puede ver gráficos y tablas comparativas con las condiciones del mar actuales y durante las siguientes 180hs que el sistema alcanza a pronosticar. Así mismo puede resultar útil a usuarios con intenciones de realizar algún viaje y todavía no han elegido el destino, dado que esta elección depende de las condiciones que el mar presente en cada lugar.

## 6.4 Limitaciones

La siguiente lista muestra algunas de las limitaciones, o posibles implementaciones que podrían ser agregadas a Surf-Forecaster en el futuro a fin de tener un sistema pronosticador aun más robusto:

* Solo predice la altura de la ola en la costa y nos da información del periodo y dirección de la misma y a su vez velocidad y dirección del viento en esa zona, pero el sistema no indica si estas condiciones son óptimas, buenas o malas para una actividad. Supongamos que al usuario le interesa practicar surf, el sistema puede indicar que la altura de las olas en un determinado lugar va a ser de 3 metros (lo cual le puede resultar una altura ideal), pero esto a veces no es suficiente, dado que si, por ejemplo, el periodo de ola es pequeño (3 o 4 segundos) y los vientos son fuertes y tienen una dirección determinada, esto puede ser un indicador de que el mar estará revuelto, generando olas grandes de altura como predijo pero desparejas y sin forma, que no son buenas para la práctica del surf, pero tal vez lo sean para la práctica de windsurf o kite-surf. En definitiva el sistema puede predecir la altura de las olas en la costa con seguridad, pero aun requiere de un mínimo conocimiento del usuario en cuanto al análisis del resto de las variables pronosticadas para saber cual ola será la más apropiada para practicar su actividad favorita.
* Surf-Forecaster requiere de una historia de al menos 60 observaciones visuales en el punto exacto en el que se quiere obtener pronósticos. Solo requiere de la altura de la ola más grande de ese día junto con la fecha de la observación lo cual no es difícil de recolectar, pero si el usuario desea obtener pronósticos de todas las olas que rompen en la costa de su ciudad tendrá que dedicar al menos dos meses (aunque es recomendable más tiempo para mejorar los pronósticos) para recorrer la costa y recolectar la información necesaria, para luego entrenar cada clasificador en el sistema.
* Si el fondo del mar varia (por ejemplo en playas donde el fondo es de arena), hay que recolectar datos actualizados cada N cantidad de meses u años (según la frecuencia con que el fondo cambia lo suficiente como para que la ola que rompe en esa zona cambie su comportamiento) para mantener entrenado con información actualizada al clasificador que pronostica dicha ola.
* El sistema requiere de hardware con grandes capacidades de almacenamiento ya que está continuamente descargando información de las olas para todos los GridPoint definidos por WAVEWATCH III del planeta los cuales reportan lecturas actualizadas cada 6 horas. Toda esta información es persistida en la base de datos que utiliza luego Surf-Forecaster como historia para futuras predicciones.

# Referencias

1. Witten, I. H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, Second Edition. Morgan Kaufmann, 2005.
2. Specht, D. F.: A General Regression Neural Network. IEEE Transactions on Neural Networks. Vol. 2(6), pp. 568-576. 1991.
3. Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone.. Classification and regression trees. Monterey, California, U.S.A.: Wadsworth, Inc. 1984.
4. Browne, M. Strauss, D. Castelle, B. Blumenstein, M. Tomlinson, R. Lane, C.: Empirical Estimation of Nearshore Waves From a Global Deep-Water Wave Model. IEEE Geoscience and Remote Sensing Letters. Vol. 3(4), pp. 462-466. 2006.
5. Caldwell, P. C. and J. P. Aucan. 2007. An empirical method for estimating surf heights from deepwater significant wave height and peak periods in coastal zones with narrow shelves, steep bottom slopes, and high refraction. Journal of Coastal Research 23/5:1237–1244.[Bioone](http://www.bioone.org/servlet/linkout?suffix=i1551-5036-25-3-539-b37&dbid=4&doi=10.2112%2F07-0958.1&key=10.2112%2F04-0397R.1)
6. Deo M.C., Sridhar Naidu C. Real time wave forecasting using neural networks. [Ocean Engineering](http://www.ingentaconnect.com/content/els/00298018;jsessionid=3ldc1bvjg94j9.alice), Volume 26, Number 3, August 1998, pp. 191-203(13). [Elsevier](http://www.ingentaconnect.com/content/els;jsessionid=3ldc1bvjg94j9.alice).
7. Makarynskyy O. Improving wave predictions with artificial neural networks. Ocean Engineering 2004, Volume 31, no5-6, pp. 709-724. Elsevier.
8. A.Baranauskas, C.Monard. Reviewing Some Machine Learning Concepts and Methods. Univeristy of Sao Paulo.2000.
9. Bishop, C. Neural Networks for Pattern Recognition. Oxford: University Press. 1995.
10. Booij, N., R. C. Ris, and L. H. Holthuijsen, A third-generation wave model for coastal regions 1. Model description and validation, J. Geophys. Res., 104(C4), 7649–7666. 1999.
11. B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor 5th Annual ACM Workshop on COLT, pp 144-152, Pittsburgh, PA, 1992. ACM Press.
12. H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. En M. Mozer, M Jordan, y T. Petsche, editores, Advances in Neural Information Processing Systems 9, paginas 155-161, Cambridge, MA, 1997, MIT Press.
13. Fausett, L. (1994). Fundamentals of Neural Networks. New York: Prentice Hall. Guyon, B. Boser and V. Vapnik. Automatic capacity tuning of very large VCdimension classiers. En Stephen Jose Hanson, Jack D. Cowan, y C. Lee Giles, editors, Advances in Neural Information Processing Systems, volumen 5, paginas 147-155, Morgan Kaufmann, San Mateo, CA, 1993.
14. J. Han y M. Kamber. Data mining: concepts and techniques, Morgan Kaufmann; 1st edition 2000.
15. Haykin, S. (1994). Neural Networks: A Comprehensive Foundation. New York: Macmillan Publishing.
16. Kubat, M., Bratki, I., y Michalski, R.S (1998). A review of Machine learning Methods.
17. D. Mattera y S. Haykin. Support vector machines for dynamic reconstruction of a chaotic system. In B. Scholkopf, C.J.C. Burges, y A.J. Smola, editores, Advances in Kernel Methods - Support Vector Learning, paginas 211-242, Cambridge, MA, 1999. MIT Press.
18. T. M. Mitchell (2006). The Discipline of Machine Learning
19. Huang, W., Foo, S., 2002. Neural network modelling of salinity variation in Apalachicola River. Water Research36, 356–362.
20. MIT resource, Artificial Neural Networks – Lecture 6, <http://ocw.mit.edu/NR/rdonlyres/Sloan-School-of-Management/15-062Data-MiningSpring2003/650A194A-828C-4990-98CE-7EB966628437/0/NeuralNet2002.pdf> (accessed August 29, 2009).
21. K.R. Muller, A. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. En W. Gerstner, A. Germond, M. Hasler, y J.D. Nicoud, editores, Artificial Neural Networks - ICANN'97, paginas 999-1004, Berlin, 1997. Springer Lecture Notes in Computer Science, Vol. 1327.
22. Patterson, D. (1996). Artificial Neural Networks. Singapore: Prentice Hall.
23. Quinlan, J. R. Learning with continuous classes. In N. Adams and L. Sterling, editors, Proceedings of the Fifth Australian Joint Conference on Artificial Intelligence,Hobart, Tasmania. Singapore: World Scientific, pp. 343–348.1992.
24. B. Scholkopf, C. Burges, Y V. Vapnik. Extracting support data for a given task. En U. M. Fayyad y R. Uthurusamy, editores, Proceedings, First International Conference on Knowledge Discovery and Data Mining, Menlo Park 1995. AAAI Press.
25. Smola y B. Scholkopf (1998). A tutorial on support vector regression.
26. <Shalizi> Cosma, Regression Trees– Lecture 10, <http://www.stat.cmu.edu/~cshalizi/350-2006/lecture-10.pdf> (accessed August 29, 2009).
27. M. Stitson, A. Gammerman, V. Vapnik, V. Vovk, C. Watkins, y J. Weston. Support vector regression with ANOVA decomposition kernels. En B. Scholkopf, C.J.C. Burges, y A.J. Smola, editores, Advances in Kernel Methods - Support Vector Learning, paginas 285-292, Cambridge, MA, 1999.MIT Press.
28. V. Vapnik, S. Golowich, y A. Smola. Support vector method for function approximation regression estimation, and signal processing. En M. Mozer, M. Jordan, y T. Petsche, editores, Advances in Neural Information Processing Systems 9, paginas 281-287, Cambridge, MA, 1997. MIT Press.
29. Caldwell, P., 2005: Validity of North Shore, Oahu, Hawaiian Islands surf observations. Journal of Coastal Research, 21(1), 2005.
30. Tolman, H. L., 1989: The numerical model WAVEWATCH: a third generation model for the hindcasting of wind waves on tides in shelf seas. Communications on Hydraulic and Geotechnical Engineering, Delft Univ. of Techn., ISSN 0169-6548, Rep. no. 89-2, 72 pp.
31. Tolman, H. L., 1991a: A third-generation model for wind waves on slowly varying, unsteady and inhomogeneous depths and currents. J. Phys. Oceanogr. , 21, 782-797
32. Tolman, H. L., 1992: Effects of numerics on the physics in a third-generation wind-wave model. J. Phys. Oceanogr., 22, 1095-1111.
33. [Tolman, H. L., 1997](http://polar.ncep.noaa.gov/mmab/papers/tn151/OMB_151.pdf): User manual and system documentation of WAVEWATCH-III version 1.15. NOAA / NWS / NCEP / OMB Technical Note 151, 97 pp. (0.74MB PDF file).
34. [Tolman, H. L., 1999a](http://polar.ncep.noaa.gov/mmab/papers/tn166/OMB_166.pdf): User manual and system documentation of WAVEWATCH-III version 1.18. NOAA / NWS / NCEP / OMB Technical Note 166, 110 pp. (0.76Mb pdf file).
35. [Tolman, H. L., 2009](http://polar.ncep.noaa.gov/mmab/papers/tn276/MMAB_276.pdf)a: User manual and system documentation of WAVEWATCH III version 3.14. NOAA / NWS / NCEP / MMAB Technical Note 276, 194 pp.+ Appendices (0.83Mb pdf file).
36. [Tolman, H. L., 2009](http://polar.ncep.noaa.gov/mmab/papers/tn276/MMAB_276.pdf)b: Operational Use of WAVEWATCH III. Webcast produced by COMET program. Accessed 02/11/2009 11:00 am. Url: <http://www.meted.ucar.edu/oceans/wavewatch3/index.htm>.
37. WMO (World Meteorological Organization), Guide to wave analysis and forecasting, 1998 Second Edition.

1. Base de datos Goddard-Caldwell: <http://www.nodc.noaa.gov/archive/arc0012/0001754/> [↑](#footnote-ref-2)
2. Acceso a datos NOAA WAVEWATCH III: <http://polar.ncep.noaa.gov/waves/download.shtml?> [↑](#footnote-ref-3)