MODELOS DESCRIPTIVOS Y PREDICTIVOS PARA LA ESTIMACIÓN DE COSTES EN PROYECTOS INFORMÁTICOS

Rubén Escribano-García

Francisco Javier Martínez-de-Pisón

Manuel Castejón-Limas

Andrés Sanz-García

Roberto Fernández-Martínez

Grupo EDMANS, Universidad de La Rioja. España

Abstract

Due to their high uncertainty, the precise estimation of real costs in software projects is a very difficult task to achieve. The use of data mining and artificial intelligence techniques could be of great help to develop predictive and descriptive models in order to assist in this task. In this paper, several models obtained from the ISBSG *Benchmarking & Research Suite Release 10* are presented. This database, provided by the *International Software Benchmarking Standards Group*, contains international software projects. The models were developed using the J48 y M5P machine learning algorithms that made possible to condensate the database knowledge into decision and regression trees. The database contains multiple project parameters: developing times and inactivity periods, projects quality, architecture types, costs, etc. From the models obtained, decision and regression trees easy to implement were developed. The results can help in the estimation of the software projects real costs according to their topology.

Keywords*: software projects; ISBSG; estimation of costs; estimation of time; data mining.*

Resumen

Realizar con precisión la estimación de los costes en proyectos de ingeniería del software es difícil debido a la elevada incertidumbre que presentan estos proyectos. La utilización de técnicas de minería de datos e inteligencia artificial puede ser de gran utilidad para desarrollar modelos predictivos y descriptivos que ayuden en estas tareas. En este artículo se muestran una serie de modelos obtenidos con los algoritmos J48 y M5P que han condensado el conocimiento de una base de datos de proyectos software internacionales, denominada *ISBSG Benchmarking & Research Suite Release 10*, en diversos árboles de decisión y regresión. Esta base de datos, suministrada por la *International Software Benchmarking Standards Group*, abarca multitud de parámetros de proyectos: tiempos de desarrollo e inactividad, calidad de los proyectos, tipos de arquitectura, costes, etc. A partir de los modelos obtenidos, se han desarrollado una serie de árboles de decisión y regresión fáciles de aplicar que pueden ser de gran ayuda para la estimación de los costes de los proyectos software atendiendo a su tipología.

Palabras clave: *proyectos informáticos; ISBSG; estimación de costes; estimación de plazos; minería de datos.*

1. Introducción

En proyectos de ingeniería del software es difícil estimar con precisión los costes y plazos de cada proyecto debida a la elevada incertidumbre que presenta este tipo de proyectos. Para evitar esto, desde finales de los años 70, se han desarrollado diferentes métricas para medir el tamaño del software y el esfuerzo necesario para realizarlo. Métricas como: la ISO/IEC 20926:2003 "IFPUG 4.1: *Unadjusted functional size measurement method - Counting practices manua*l", la ISO/IEC 19761:2003 "COSMIC-FFP: *A Functional Size Measurement Method*", la ISO/IEC 20968:2002 "*Mk II: Function Point Analysis - Counting Practices Manual*", la ISO/IEC 24570:2004 "*NESMA Guide to Using Function Point Analysis*" o la norma española equivalente a la ISO 14143, UNE 71045-1:2000. "*Tecnología de la información. Medida del Software. Medida del tamaño funcional*."; permiten medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y explotación del software, y también ser útil en cualquiera de las fases de vida del software, desde el diseño inicial hasta la explotación y mantenimiento.

Una de las posibilidades de disponer de una medida del tamaño funcional del software es la de poder comparar el coste del desarrollo de aplicaciones (y otros parámetros de gestión) entre diferentes proyectos y organizaciones (Molokken-Ostvold et al., 2004; Jørgensen, 2004). Aunque esto permite determinar los costes de los proyectos software de una forma comparable, la utilización de este tipo de métricas suele suponer un esfuerzo en personal y tiempo del que no se dispone (Cicmil et al., 2006). Además, suelen carecer de precisión en proyectos pequeños.

Para poder contrastar las métricas con casos reales, el "*International Software Benchmarking Standards Group (ISBSG)*" mantiene una base de datos de métricas de diversos proyectos dependiendo de la tecnología utilizada, el tamaño del proyecto, los requisitos de calidad exigidos y otros parámetros. Esta base de datos abarca multitud de parámetros de proyectos: tiempos de desarrollo e inactividad, calidad de los proyectos, tipos de arquitectura, costes, etc. Estos datos pueden ser útiles para la estimación de costes y plazos pues corresponden con casos reales (Ko et al., 2007). Por ejemplo, Villanueva (2005) utilizó un método multivariante adaptativo de regresión por splines (MARS) para predecir los plazos y los costes en proyectos software a partir de la base de datos ISBSG.

En estos últimos años han surgido técnicas de Minería de Datos orientadas al análisis de Bases de datos que pretenden ayudar a analizar grandes bases de datos con el objetivo de obtener conocimiento útil y oculto que pueda servir de ayuda para la toma de decisiones o la generación de nuevos modelos de predicción. Se puede decir que la minería de datos es un conjunto de metodologías y herramientas que mediante el análisis de grandes cantidades de datos nos ayuda a obtener patrones de comportamiento o tendencias ocultas que pueden ser muy útiles en la toma de decisiones. Para alcanzar buenos resultados es necesario comprender que la minería de datos no se basa en una metodología estándar y genérica que resuelve todo tipo de problemas, sino que consiste en una metodología dinámica e iterativa que va a depender del problema planteado, de la disponibilidad de la fuentes de datos, del conocimiento de las herramientas necesarias, de la metodología desarrollada y de los requerimientos y recursos de la empresa.

En este artículo, se presenta un estudio centrado en la base de datos de proyectos software internacionales denominada *ISBSG Benchmarking & Research Suite Release 10*, con el objetivos de obtener y condensar el conocimiento existente en esta base de datos que pueda ser útil para la toma de decisiones en la Dirección de Proyectos Software. El estudio se ha realizado con dos algoritmos muy utilizados en Minería de Datos: árboles de decisión y árboles de regresión.

1.1 Árboles de Decisión y Regresión

Aunque existen muchas técnicas para la extracción de conocimiento oculto, en muchas ocasiones la mayoría son incapaces de ajustarse correctamente cuando los datos son muy heterogéneos, la densidad de los datos es muy irregular o el número de datos no es elevado. Además, en muchas ocasiones, requieren un esfuerzo considerable en los procesos de entrenamiento.

Unas técnicas de modelado que pueden ser de gran utilidad para estos casos son los árboles de decisión o regresión basados en modelos. Éstos dividen el espacio de instancias y generan modelos exclusivos para cada uno de los grupos de datos que corresponden a esa división.

Los árboles de decisión son muy populares en tareas de clasificación (donde el objetivo es asignar cada ejemplo o instancia a una clase determinada). Un árbol de decisión tiene unas entradas las cuales pueden ser un objeto o una situación descrita por medio de un conjunto de atributos y a partir de esto devuelve una respuesta la cual es una decisión que es tomada a partir de las entradas. Los valores que pueden tomar las entradas y las salidas pueden ser valores discretos o continuos. Cuando se utilizan valores discretos en las funciones de una aplicación se denominan árboles de decisión y cuando se utilizan los continuos se denominan árboles de regresión. Generalmente, los árboles de regresión provienen de la adaptación de árboles de decisión a tareas de regresión.

Existen diversos algoritmos para generar árboles de decisión: ID3 and C4.5 (Quinlan, 1986), CART (Breiman et al., 1984), etc. Sin embargo, los árboles de decisión no pueden aplicarse directamente a problemas de predicción de valores numéricos (regresión). Para poder desarrollar árboles que puedan funcionar como regresores se han desarrollado diversas aproximaciones (Breiman et al., 1984), Friedman (1991) o el algoritmo M5 (Quinlan, 1992; Wang y Witten, 1997), por ejemplo.

Uno de los más utilizados es el algoritmo M5 pues está implementado en herramientas gratuitas como WEKA (Witten y Frank, 2005), es fácil de utilizar y los modelos que generan son eficientes.

El algoritmo M5 desarrolla un modelo lineal multivariante adecuado para cada particular dominio del espacio de entrada. Es decir, cada una de las ramas del árbol clasifica un grupo de casos mostrando en la hoja final de la misma el modelo de regresión lineal que mejor se ajusta a los mismos (ver Figura 1).

Inicialmente, M5 divide el espacio de instancias en grupos muy pequeños. Esto genera un árbol muy complejo que necesita ser podado (reducido) a un árbol más sencillo para poder mejorar la capacidad de generalización del mismo. El proceso de podado se realiza en cada rama hasta que se supera un umbral de error predeterminado. Este error viene determinado por la capacidad de predicción del modelo lineal multivariante de la hoja que queda después del podado.

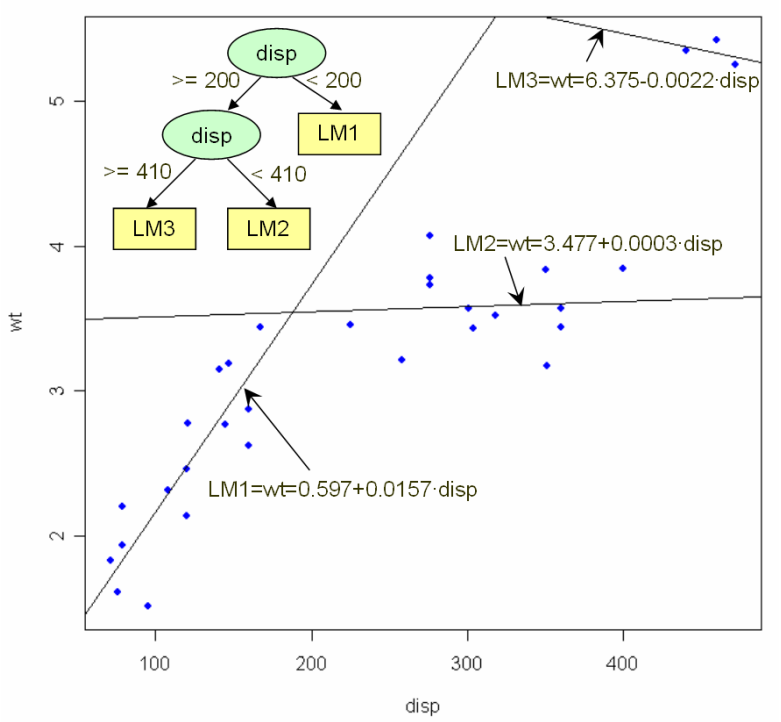
La gran ventaja de este tipo de técnicas es que M5 desarrolla una colección de modelos lineales localmente precisos mediante la selección de las variables y sus valores, que mejor agrupan cada familia de casos. De esta forma:

- M5 puede ajustarse automáticamente y eficientemente a datos altamente no lineales y desestructurados.
- Es muy robusto frente a espurios pues éstos son clasificados en una rama distinta.
- Puede trabajar con un elevado número de atributos (cientos de atributos), incluso con atributos no significativos o vacíos, pues el algoritmo selecciona aquellos que mejor clasifican los grupos para cada una de las ramas del árbol y desestima las menos útiles.

- El modelo resultante, si no es excesivamente grande, es transparente y fácil de interpretar.

Estas técnicas han sido ampliamente usadas en Minería de Datos para el desarrollo de modelo predictivos en campos como el empresarial, medioambiental, medicina, etc.

Figura 1: Ejemplo de un árbol de regresión.



2. Objetivos

El objetivo de este artículo es la presentación de los resultados obtenidos a partir de la aplicación de técnicas de Minería de Datos a una base de datos de proyectos software internacionales denominada *ISBSG Benchmarking & Research Suite Release 10*.

La técnica utilizada corresponde con los conocidos algoritmos de aprendizaje autómatico J48 y M5P que generan árboles de decisión y regresión muy eficientes en la extracción de conocimiento oculto de bases de datos heterogéneas, con gran cantidad de datos vacíos, ruido, etc.

A partir de los modelos obtenidos, se han desarrollado una serie de modelos fáciles de aplicar que pueden ser de gran ayuda para la estimación de los costes de los proyectos software atendiendo a su tipología.

3. Metodología

La metodología propuesta es la siguiente:

1. Primero se realiza una preselección de variables, eliminando aquellas que sean redundantes o que no aporten nada útil.
2. Después es necesario filtrar, recomponer, eliminar espurios, transformar la información, etc.; tareas todas éstas típicas de las primeras fases de la MD.

3. Una vez obtenida la base de datos final, se desarrollan diversos árboles de decisión y regresión variando las variables y el tipo de configuración de los algoritmos hasta que el error de testeo de validación cruzada es adecuado y los modelos son fácilmente interpretables. La validación cruzada permite estimar el error que cometerá un clasificador al aplicarlo a nuevos ejemplos sin tener que perder ningún caso de la base de datos para el proceso de validación. La idea básica detrás de la validación es construir un clasificador con un conjunto de ejemplos y estimar lo bien que clasifica con otro conjunto diferente. No se trata de que el modelo memorice un conjunto sino de que generalice el problema lo mejor que pueda. Concretamente, la validación cruzada rompe el conjunto original en *C* subconjuntos y calcula *C* clasificadores, dejando un subconjunto fuera, cada vez, en la construcción de cada clasificador. El subconjunto que se aparta en la construcción de cada clasificador es el que se utiliza para la estimación del error del clasificador. La media de estos errores es el error estimado por validación cruzada. El modelo final, en cambio, se realiza con toda la base de datos.

El propósito es determinar si el error cuadrático medio (*Root Mean Squared Error (RMSE)*) o el error medio absoluto (*Mean Absolute Error (MAE)*) son suficientemente bajos para considerar los modelos predictivos como aceptables.

Estos errores se definen como:

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(y(k) - \hat{y}(k))^2} \quad (1)$$

y

$$MAE = \frac{1}{n}\sum_{k=1}^{n}\left|y(k) - \hat{y}(k)\right| \quad (2)$$

donde y y $\hat{y}$ son, respectivamente, los valores reales y los que predice el modelo y n el número de puntos usados para validar el modelo.

4. Desarrollo de los Modelos y Resultados

4.1 Descripción de la Base de Datos

La base de datos denominada *ISBSG Benchmarking & Research Suite Release 10*, es suministrada por la *International Software Benchmarking Standards Group*, y corresponde con 4.106 proyectos software internacionales con 106 variables que abarcan información sobre: calidad de la información, tamaño según diversas métricas, esfuerzo dedicado, productividad, planificación, calidad, tipo de proyecto, arquitectura usada, documentos y técnicas, producto generado, etc.

4.2 Preprocesado de los Datos

Para conseguir resultados más realistas se eliminaron aquellos casos que tenía una calidad de información nivel C o D. Es decir, solamente se eligieron proyectos con una calidad de información buena (B) o muy buena (A). Además, solamente se eligieron aquellos proyectos que habían sido desarrollados en los últimos 6 años. El siguiente paso consistió en reducir la dimensión de la tabla para conseguir un número de variables adecuadas. En la Tabla 1 se muestran las variables finales seleccionadas.

Una vez seleccionadas las variables más importantes, se filtraron los espurios en las variables numéricas y, en algunas variables categóricas, se realizaron agrupamientos para reducir el número de casos posibles.

Finalmente, el número de instancias de la base de datos se redujo a 1.245 proyectos.

Tabla 1: Variables Utilizadas

Variable	Descripción
ProjectID	Identificación del Proyecto
AdjustedFunctionPoints	Puntos-Función Ajustados según IFPUG-FPA
SummaryWorkEffort	Esfuerzo Total de Trabajo en Horas Hombre
NormalisedPDR	Ratio de Horas/Hombre por Punto-Función
ProjectElapsedTime	Tiempo de Ejecución del Proyecto en Meses
ProjectInactiveTime	Tiempo de Inactividad del Proyecto en Meses
TotalDefectsDelivered	Número de Defectos Totales Encontrados
DevelopmentType	Nuevo desarrollo, repetición de un desarrollo, mejora
OrganisationType	Tipo de Organización
ApplicationType	Tipo de Aplicación
Architecture	Arquitectura Usada
Development Platform	Plataforma de Desarrollo
Hardware	Hardware usado
OperatingSystem	Sistema Operativo Usado
Language	Lenguaje de Programación Usado Principalmente
DataBaseSystem	Motor de Bases de Datos Usado
InputCount	Porcentaje de Puntos-Función en Funciones de Entrada
OutputCount	Porcentaje de Puntos-Función en Funciones de Salida
EnquiryCount	Porcentaje de Puntos-Función en Funciones de Consulta
FileCount	Porcentaje de Puntos-Función en Funciones de Archivos
InterfaceCount	Porcentaje de Puntos-Función en Funciones de Interfaces Externos
AddedCount	Porcentaje de Puntos-Función en Adición de Funciones
ChangedCount	Porcentaje de Puntos-Función en Funciones Cambios de Funciones
DeletedCount	Porcentaje de Puntos-Función en Borrado de Funciones
LinesOfCode	Líneas de Código
Year	Año de Desarrollo del Proyecto

4.3 Modelo de Predicción del "Ratio de Horas/Hombre por Punto-Función"

Uno de los modelos que se consideró que podía ser más interesante consistió en aquél capaz de predecir el ratio de horas/hombre por punto-función a partir de la tecnología utilizada, el tipo de proyecto y organización; u otros parámetros que definieran al proyecto.

El mejor modelo consistió en un árbol de regresión de 10 hojas tal y como se muestran en la Figura 2. Este modelo obtuvo un error de predicción MAE del 3,52% y RMSE del 6,83%. Éste consideraba como variables de entrada: el tiempo esperado de ejecución, el tipo de organización y aplicación; y el sistema operativo, lenguaje y motor de bases de datos.

En la Figura 2 se muestra el árbol obtenido y la función de regresión lineal de la primera hoja del mismo (LM1). Tal y como se puede observar, el modelo seleccionó fundamentalmente las variables "tipo de aplicación", "duración estimada en meses", "tipo de organización", "tipo de base de datos" y "lenguaje de programación usado".

Aunque el algoritmo "se alimentó" con muchas más variables de entrada, el algoritmo M5P seleccionó aquellas que más importancia tenían en la estimación del ratio buscado. El error medio absoluto (MAE) es realmente bajo, considerando el grado de incertidumbre de este tipo de datos y, aunque el error RMSE es el prácticamente el doble, sigue siendo bastante aceptable para una primera estimación de los ratios. Hay que destacar, que un RMSE más alto es lógico pues existen muchos casos aislados que el modelo no será capaz de predecir con suficiente precisión y, como el RMSE, es un error mucho más restrictivo ante espurios el valor de éste aumenta en estos casos.

Figura 2: Ejemplo de Modelo Obtenido del "Ratio de Horas/Hombre por Punto-Función"

```
ApplicationType=Protocols,Softwaredevelopmenttool,EmbeddedSystems,Imagevideosoundprocessing,DSP,Mathematicalmodelling,Financial,ProjectManagement,MSBus:
|   ApplicationType=Humanressourcesmanagement,ElectronicDataInterchange,OLAP,LicencesPermits,RelativelyComplexApplication,Customerbilling,Catalogue,Ope:
|   |   ProjectElapsedTime <= 2.6 : LM1 (20/4.739%)
|   |   ProjectElapsedTime >  2.6 : LM2 (63/10.767%)
|   ApplicationType=Humanressourcesmanagement,ElectronicDataInterchange,OLAP,LicencesPermits,RelativelyComplexApplication,Customerbilling,Catalogue,Ope:
|   |   ProjectElapsedTime <= 6.736 :
|   |   |   ProjectElapsedTime <= 6.236 : LM3 (49/24.843%)
|   |   |   ProjectElapsedTime >  6.236 : LM4 (156/29.072%)
|   |   ProjectElapsedTime >  6.736 : LM5 (58/40.098%)
ApplicationType=Protocols,Softwaredevelopmenttool,EmbeddedSystems,Imagevideosoundprocessing,DSP,Mathematicalmodelling,Financial,ProjectManagement,MSBus:
|   ProjectElapsedTime <= 4.85 :
|   |   Language=Taps,Datastage,MS_Navision,BEA Weblogic,SLOGAN,DELPHI,Vbasic,C,ORACLE,JAVA,TNSDL,COBOL <= 0.5 : LM6 (172/24.256%)
|   |   Language=Taps,Datastage,MS_Navision,BEA Weblogic,SLOGAN,DELPHI,Vbasic,C,ORACLE,JAVA,TNSDL,COBOL >  0.5 :
|   |   |   ProjectElapsedTime <= 1.72 : LM7 (19/108.143%)
|   |   |   ProjectElapsedTime >  1.72 : LM8 (64/72.353%]
|   ProjectElapsedTime >  4.85 :
|   |   Language=COBOL <= 0.5 : LM9 (184/104.198%)
|   |   Language=COBOL >  0.5 : LM10 (49/194.554%)

LM num: 1
NormalisedPDR =
        0.083 * ProjectElapsedTime
        + 1.257 * OrganisationType=Financial,Communications,Consumer Goods,HumanResource,Publishing,Logistic,Retail,Government,Real Estate & Property S
        + 0.684 * OrganisationType=Consumer Goods,HumanResource,Publishing,Logistic,Retail,Government,Real Estate & Property Services;Community Service:
        + 1.2446 * ApplicationType=Web,Personalproductivity,Softwareformachinecontrol,other,Marketing,Trading,Stockcontrolorderprocessing,Reporting,Te.
        + 1.1829 * ApplicationType=Humanressourcesmanagement,ElectronicDataInterchange,OLAP,LicencesPermits,RelativelyComplexApplication,Customerbillin
        - 1.1306 * ApplicationType=LicencesPermits,RelativelyComplexApplication,Customerbilling,Catalogue,Operatingsystemorsoftwareutility,Workflowsupp
        + 1.6192 * ApplicationType=RelativelyComplexApplication,Customerbilling,Catalogue,Operatingsystemorsoftwareutility,Workflowsupport&management,P:
        + 0.6469 * ApplicationType=Geographicspatialinformation,Paymentsocialpensions,Documentmanagement,Taxsystem,Logistic,NetworkManagement,MSBilling
        + 0.3814 * OperatingSystem=HP_UX,Exec,ClientServer,Solaris,Windows,Netware,Proprietary,P2K_Platform,DMS_OS,MVS,OS_390,Z_OS,DOS
        + 0.5387 * OperatingSystem=ClientServer,Solaris,Windows,Netware,Proprietary,P2K_Platform,DMS_OS,MVS,OS_390,Z_OS,DOS
        - 2.0061 * OperatingSystem=Netware,Proprietary,P2K_Platform,DMS_OS,MVS,OS_390,Z_OS,DOS
        + 1.2698 * OperatingSystem=MVS,OS_390,Z_OS,DOS
        + 1.1426 * Language=XML,ASSEMBLER,SAS,SAP,4GL,ASP,RPG3,ABAP;,CoolGen,MSAccess,Others,Java,SQL,5GL,PLI,3GL,Taps,Datastage,MS_Navision,BEA Weblog:
        + 0.5404 * Language=ASP,RPG3,ABAP;,CoolGen,MSAccess,Others,Java,SQL,5GL,PLI,3GL,Taps,Datastage,MS_Navision,BEA Weblogic,SLOGAN,DELPHI,Vbasic,C,
        - 0.2613 * Language=Java,SQL,5GL,PLI,3GL,Taps,Datastage,MS_Navision,BEA Weblogic,SLOGAN,DELPHI,Vbasic,C,ORACLE,JAVA,TNSDL,COBOL
        - 0.4595 * Language=PLI,3GL,Taps,Datastage,MS_Navision,BEA Weblogic,SLOGAN,DELPHI,Vbasic,C,ORACLE,JAVA,TNSDL,COBOL
        + 0.5212 * Language=3GL,Taps,Datastage,MS_Navision,BEA Weblogic,SLOGAN,DELPHI,Vbasic,C,ORACLE,JAVA,TNSDL,COBOL
        + 0.5858 * Language=Taps,Datastage,MS_Navision,BEA Weblogic,SLOGAN,DELPHI,Vbasic,C,ORACLE,JAVA,TNSDL,COBOL
        - 0.572 * Language=Datastage,MS_Navision,BEA Weblogic,SLOGAN,DELPHI,Vbasic,C,ORACLE,JAVA,TNSDL,COBOL
        - 0.2994 * Language=MS_Navision,BEA Weblogic,SLOGAN,DELPHI,Vbasic,C,ORACLE,JAVA,TNSDL,COBOL
        + 1.0806 * Language=C,ORACLE,JAVA,TNSDL,COBOL
        - 1.1563 * Language=ORACLE,JAVA,TNSDL,COBOL
        + 2.0086 * Language=JAVA,TNSDL,COBOL
        - 0.3996 * Language=TNSDL,COBOL
        + 1.3151 * Language=COBOL
        + 3.1017 * DataBaseSystem=Custom,MySQL,RDMS-1100,Batch,AS_400DBMS,Interactive,DB2,Sybase,Informix,IDMS_DB,SAP
        - 0.5005 * DataBaseSystem=AS_400DBMS,Interactive,DB2,Sybase,Informix,IDMS_DB,SAP
        + 0.4987
```

4.4 Modelos de Predicción del "Porcentaje de Puntos-Función" en el Desarrollo de Funciones de Entrada, Salida, Consulta, Archivos, etc.

Otros modelos que se consideraron interesantes para los Directores de Proyectos Software, consistieron en aquellos que permitieran determinar el "porcentaje de esfuerzo" necesario para cada parte del proyecto según su tamaño esperado, tipología, organización y duración estimada.

En la Tabla 2 se muestran los resultados de los mejores modelos obtenidos. En este caso, solamente se disponía de este tipo de información del 20% de los proyectos de la base de datos.

Como se puede apreciar, los errores medios son mucho mayores que el modelo anterior, estando la mayoría de ellos entorno al 15% de error MAE y el 20% de error RMSE. Lógicamente, no son modelos muy precisos aunque es lógico porque este tipo de información es mucho más aleatoria y depende de muchos otros factores que no aparecen en la tabla. Aún así, pueden ser de ayuda para estimar "aunque con reservas" qué porcentajes de esfuerzo van a ser necesarios en cada apartado para un tipo de proyecto determinado.

Figura 3: Ejemplo de Modelo Obtenido para "EnquiryCount"

```
M5 pruned model tree:
(using smoothed linear models)

OrganisationType=AgricultureForestryFishingHunting,Retail,ContentManag
|   ProjectElapsedTime <= 0.025 : LM1 (44/45.492%)
|   ProjectElapsedTime >  0.025 : LM2 (24/2.245%)
OrganisationType=AgricultureForestryFishingHunting,Retail,ContentManag
|   Language=C,SAP,PLI,ABAP;,NET,Centura2000 <= 0.5 :
|   |   ApplicationType=SystemSoftware,MiddleWare,Protocols,Sales,DSP,
|   |   ApplicationType=SystemSoftware,MiddleWare,Protocols,Sales,DSP,
|   |   |   ProjectElapsedTime <= 0.029 : LM4 (30/61.915%)
|   |   |   ProjectElapsedTime >  0.029 : LM5 (18/77.592%)
|   Language=C,SAP,PLI,ABAP;,NET,Centura2000 >  0.5 : LM6 (59/76.13%)

LM num: 1
EnquiryCount =
        -0.1546 * ProjectElapsedTime
        + 0.0115 * OrganisationType=Communications,AerospaceAutomotive
        + 0.0371 * OrganisationType=AgricultureForestryFishingHunting,
        + 0.1297 * ApplicationType=Softwareformachinecontrol,Financial
        - 0.0097 * ApplicationType=Documentmanagement,Telecommunicatio
        + 0.0302 * ApplicationType=ProcessControl,other,Workflowsuppor
        - 0.0153 * OperatingSystem=Solaris,HP_UX,VxWorks,Windows,Clien
        + 0.0783 * OperatingSystem=VxWorks,Windows,ClientServer,Web,Pr
        - 0.082 * OperatingSystem=ClientServer,Web,Proprietary,Netware
        + 0.0103 * Language=Vbasic,Java,ACCESS,ASP,Perl,Others,Uniface
        + 0.014 * Language=ACCESS,ASP,Perl,Others,Uniface8,XML,MS_Navi
        + 0.0194 * Language=C,SAP,PLI,ABAP;,NET,Centura2000
        - 0.0658
```

Tabla 2: Errores obtenidos de los mejores modelos de predicción para los "Porcentajes de Puntos-Función" de cada Tipo de Funciones

Variable	Número de Hojas del Árbol	MAE	RMSE
InputCount	2	16,63%	20,46%
OutputCount	1	14,30%	18,92%
EnquiryCount	6	15,29%	19,70%
FileCount	2	14,70%	19,12%
InterfaceCount	4	11,55%	18,21%
AddedCount	9	20,17%	27,10%
ChangedCount	5	15,77%	21,14%
DeletedCount	1	02,78%	08,42%

4.5 Modelos Descriptivos Mediante el Algoritmo J48

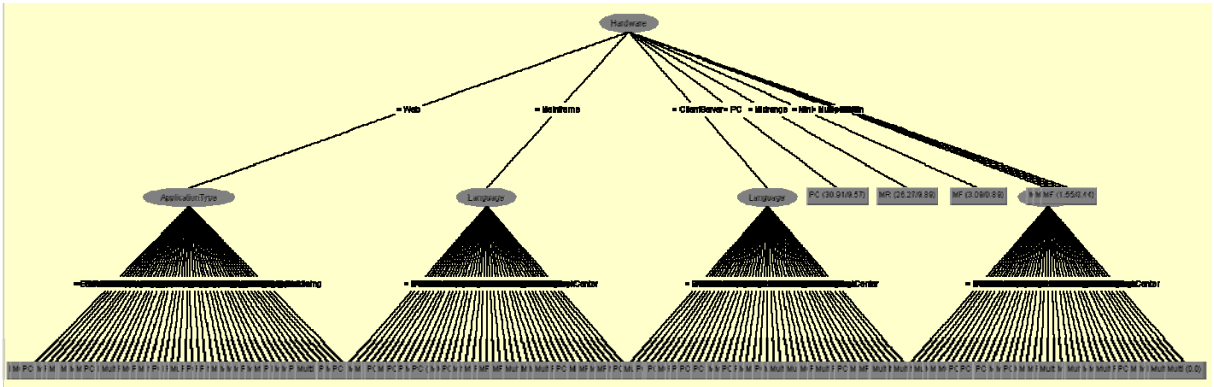
La gran ventaja del uso de este tipo de técnicas, es que es posible desarrollar árboles que nos describan conocimiento extraído de la propia base de datos que pueda ser útil a otro tipo de personas como: vendedores de software, hardware, sistemas operativos, etc.

En la Figura 4 se muestra un ejemplo de árbol clasificador que clasifica la plataforma primaria (PC, Rango Medio (MR), Main Frame (MF) o Multiplataforma (Multi) según el tipo de organización, el tipo de hardware y lenguaje de programación usado. La precisión del clasificador es del 83,69%.

Igualmente, se pueden desarrollar árboles que "expliquen" variables como el tipo de lenguaje de programación usado mayoritariamente o el tipo de bases de datos según el tipo de organización y el tamaño de los proyectos.

Todos estos modelos pueden ayudar a descubrir cual es el comportamiento de las organizaciones a la hora de desarrollar software y puede ser muy útil para consultorías o empresas de ventas de equipos informáticos o software.

Figura 4: Ejemplo de Modelo Obtenido para "EnquiryCount"



5. Conclusiones

En este artículo se ha pretendido mostrar las enormes posibilidades que muestran los árboles de decisión y regresión para el apoyo a la toma de decisiones en el desarrollo de Proyectos Software. La utilización de estas técnicas con una base de datos internacional de Proyectos Software permite extraer conocimiento, no conocido previamente, de la realidad existente en este tipo de proyectos.

Obviamente, los modelos obtenidos no pueden ser 100% precisos aunque el objetivo, en este caso, es proveer de herramientas de predicción que puedan ser útiles en su conjunto.

De entre los múltiples modelos desarrollados, es de destacar el modelo que permite predecir el ratio. Este modelo, ha sido desarrollado para determinar el ratio de horas/hombre por punto-función con solamente tres variables en el árbol y cinco en los modelos lineales finales de cada hoja: tipo de aplicación, duración estimada en meses, tipo de organización, tipo de base de datos y lenguaje de programación a utilizar. Con sólo estos cinco parámetros, el modelo es capaz de predecir con un error medio absoluto (MAE) del 3,5% el ratio buscado. Debido a las enormes incertidumbres de los Proyectos Software, se considera que el modelo es bastante preciso y puede ser de gran ayuda a la hora de estimar el coste real que pueda tener un proyecto determinado.

6. Referencias

Cicmil, S., Williams, T., Thomas, J. & Hodgson, D. (2006). Rethinking project management: Researching the actuality of projects. *International Journal of Project Management, 24*, (8), 675–686.

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and regression trees*. California: Wadsworth & Brooks/Cole Advanced Books & Software.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *Annals of Statistics, 19*, 1-141.

Jørgensen, M. (2004). A review of studies on expert estimation of software development effort. *Journal of Systems and Software, 70*, (1-2), 37–60.

Ko, Y., Park, S., Seo, J. & Choi, S. (2007). Using classification techniques for informal requirements in the requirements analysis supporting system. *Information and Software Technology, 49*, (11–12), 1128–1140.

Molokken-Ostvold, K., Jorgensen, M., Tanilkan, S. S., Gallis, H., Lien, A. C., & Hove, S. E. (2004). A survey on software estimation in the norwegian industry. METRICS '04: Proceedings of the Software Metrics, 10th International Symposium (págs. 209-219). Washington DC.: IEEE Computer Society. doi:http://dx.doi.org/10.1109/METRICS.2004.5.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning 1*, (1), 81-106.

Quinlan, J. R. (1992). Learning with Continuous Classes. 5th Australian Joint Conference on Artificial Intelligence, Singapore, 343-348.

Villanueva, J. M. (2005). *Estimación de costes y plazos en proyectos de sistemas de información*. Tesis doctoral, Universidad de Oviedo, Oviedo, España.

Wang, Y. & Witten, I. H. (1997). Induction of model trees for predicting continuous classes. *9th European Conference on Machine Learning*.

Witten, I. H. & Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. San Francisco: Morgan Kaufmann.

Correspondencia (Para más información contacte con):

Dr. Francisco Javier Martínez de Pisón Ascacíbar
Grupo EDMANS. URL: http://www.mineriadatos.com
Área de Proyectos de Ingeniería. Departamento de Ingeniería Mecánica
Edificio Departamental. ETSII de Logroño. C/ Luís de Ulloa, 20, 26004 Logroño (España).
Phone: +34 941 299 232
Fax: + 34 941 299 794
E-mail: fjmartin@unririoja.es.