ANÁLISE DE DADOS NA PREDIÇÃO DE PROJETOS DE SOFTWARE

Luiz F. P. Teixeira[1]
lluizteixeira@gmail.com
Fábio Pittoli[1]
fabio.pittoli@gmail.com
Abraham L. R. de Sousa[1,2]
rabelo@unilasalle.edu.br
Daltro J. Nunes[2]
daltro@inf.ufrgs.br

[1] Centro Universitário La Salle – UNILASALLE – Canoas, RS
[2] Universidade Federal do Rio Grande do Sul – UFRGS – Porto Alegre, RS

RESUMO

Este artigo tem como objetivo a investigação do impacto que a análise de dados sobre estimativas geradas através de Redes Bayesianas - RB. Fluxos de mineração, técnicas e algoritmos de tratamento de dados são apresentados. Os dados tratados são usados para treinar uma RB que é utilizada em análises What-if. Conclui-se que a qualidade dos dados pode influenciar nas estimativas e aconselha-se o uso do algoritmo K-Means.

Palavras-chave: Estimativas de Software. Mineração de Dados. Redes Bayesianas.

ABSTRACT

This paper aims to investigate the data analysis impact over estimates produced through Bayesian Networks - BN. Data flow mining techniques and algorithms for data processing are presented. We conclude that data quality may influence the estimates generated by BN and advise the use o K-Means algorithm.

Keywords: *Software estimatives, Data mining, Bayesian Networks*

1 NTRODUÇÃO

Há diversas pesquisas voltadas para a engenharia de software relacionada a novas ferramentas e modelos para auxiliar os gerentes de projeto na complexa tarefa de concretizar um projeto de desenvolvimento de software com sucesso. Porém, mesmo com todo o progresso nos estudos e pesquisas, ainda há desafios a serem solucionados. Entre eles é possível citar a elaboração de estimativas para os projetos de desenvolvimento de software como sendo uma das tarefas mais complexas que um gerente de projetos pode enfrentar. As estimativas são tão vitais para os projetos que podem determinar seu sucesso ou fracasso.

Sendo assim, as estimativas podem ser consideradas a fundação para todas as outras atividades de planejamento do projeto.

As estimativas têm como principal objetivo mensurar o projeto que está sendo desenvolvido. No entanto, há um grande índice de projetos com estimativas imprecisas ou incorretas. Há diversas técnicas que visam amenizar esse tipo de problema, sendo uma das mais indicadas as Redes Bayesianas, que são grafos de probabilidade, representando cenários de um projeto, com o objetivo de realizar predições levando em conta situações de causa e efeito. Todavia o uso das Redes Bayesianas requer dados históricos para que seu modelo seja treinado a fim de aperfeiçoar suas predições. Entretanto nem sempre a base de dados históricos utilizada no treinamento tem dados com a qualidade necessária, o que pode comprometer a precisão das estimativas geradas pela Rede Bayesiana.

Este estudo apresenta uma análise de impacto que o uso de técnicas de mineração de dados pode exercer sobre as estimativas geradas através de Redes Bayesianas.

2 TRABALHOS RELACIONADOS

Até o momento não foi encontrado nenhuma publicação que apresentasse um estudo sobre o impacto da análise de dados em estimativas de software geradas através de Redes Bayesianas.

A pesquisa que mais se assemelha é um artigo apresentado por Gernot Liebchende, Bheki Twala, Martin Shepperd e Michelle Cartwright através da publicação "*Assessing the Quality and Cleaning of a Software Project Dataset: An Experience Report*" em 2006, que se trata de um relatório sobre uma experiência usando três técnicas de eliminação de ruídos em bases de dados de projetos de software, com o intuito de verificar qual técnica qualificaria melhor os dados para uso posterior em ferramentas de auxilio a tomada de decisão.

Outra publicação semelhante em alguns aspectos principalmente por testes com Redes Bayesianas é o artigo "*Bayesian Network Models for Web Effort Prediction: A Comparative Study*", de Emilia Mendes e Nile Mosley em 2008 que surge como uma importante fonte de referencia na área de Redes Bayesianas, pois propõe comparar, usando um conjunto de dados cruzados, vários modelos de Redes Bayesianas para estimar esforço. Ao longo do artigo é apresentado os motivos da utilização das redes bayesianas bem como todo o desenvolvimento e comparativo das oito redes utilizadas, sendo quatro delas geradas automaticamente através da utilização das ferramentas Hugin e PowerSoft. Destaca-se que foram utilizados dados de 130 projetos web para realizar o estudo.

Outro estudo possível de ser citado é o artigo de Jhonas Bonfante Guinzani, Priscyla Waleska Targino de Azevedo Simões, Merisandra Côrtes de Mattos, Jane Bettiol através da publicação "*Mineração de Dados em Redes Bayesianas Utilizando a API da Shell Belief Network Power Constructor (BNPC)*". Esta pesquisa tem como objetivo a integração de softwares de apoio a tomada de decisões com base de dados históricos, com a finalidade de auxiliar o gerente de projetos de software na sua tomada de decisões. Para concretizar esse objetivo o modelo proposto utiliza técnicas de mineração de dado em conjunto com redes bayesianas, oferecendo assim recursos de aprendizagem automatizada. Tendo como intuito que essas regras sejam integradas ao uma interface gráfica intuitiva para o gerente de projetos.

3 MÉTODOS, TÉCNICAS E FERRAMENTAS DE ANÁLISE DA INFLUÊNCIA DOS DADOS

Nesta seção é descrita as etapas desta pesquisa, bem como as ferramentas e as técnicas utilizadas para a análise do impacto que a mineração de dados exerce em estimativas geradas através de uma Rede Bayesiana. O resultado desse experimento consiste em uma análise das mudanças comportamentais da Rede Bayesiana quando treinada com uma base normal, que nunca passou por um processo de mineração de dados, e quando treinada com a base resultante de um processo de mineração de dados.

Nesse experimento foram utilizados dois algoritmos para classificação dos dados: J48 (Arvores de decisão) e SimpleKMeans (K-means), afim de atestar que o processo de analise de dados e eliminação de ruídos em bases de dados históricos, tem um impacto significativo nas predições geradas através de Redes Bayesianas. Além desses resultados foram analisados outros aspectos envolvendo as técnicas de mineração de dados, relativo ao desempenho e a taxa de erros corrigidos em cada algoritmo.

Para atingir esse objetivo, devemos seguir uma serie de passos os quais compreendem: a modelagem de um processo de desenvolvimento de software, a construção de uma base de dados sintética para simulação dos dados históricos de projetos, a utilização de ferramentas e técnicas para mineração de dados, além de ferramentas para auxilio à tomada de decisões que fazem predições através de redes bayesianas. O objetivo é investigar o impacto que a análise de dados exerce sobre estimativas geradas através de Redes Bayesianas. Levando em consideração um fluxo de atividades que o gerente de projetos executa, e que exemplifica o modelo proposto de uso de bases históricas de projetos, análise de dados e ferramentas de apoio à tomada de decisão (Redes Bayesianas).

As ferramentas utilizadas neste experimento foram Weka (Mineração de dados), Genie (Redes Bayesianas) e WebApsee (Ferramenta para Gerencia de Projetos), que serão detalhadas ao decorrer do texto.

3.1 O Modelo de Processo e Ambiente de Apoio

Para realização do estudo foi necessária a criação de processo de desenvolvimento de software que pudesse ser adotado como linha base de comparação entre as predições com dados tratados e as sem tratamento.

Assim, para simular um projeto de desenvolvimento foi utilizado o *WebApsee*, um ambiente de apoio a gerencia projetos baseado em processos. Nele foi definido um processo de desenvolvimento simulando um projeto de software real, também foram definidas algumas variáveis contidas em qualquer projeto de software, como definição de atividades, numero de participantes, tempo estimado de cada atividade, entre outras. Esse processo e suas variáveis são importantes para a simulação, pois definem vários aspectos da base dados histórica e simulam o comportamento de um gerente de projetos.

O *WebApsee* foi desenvolvido pelo LABES-UFPA em 2003 a partir de um projeto de pesquisa sobre problemas críticos relacionados a gerencia de projetos. Em um ambiente fácil e intuitivo o gerente de projetos pode facilmente gerenciar qualquer projeto. Dentre as suas funcionalidades destacam-se: a) modelagem de processos de desenvolvimento de Software; b) definição de tarefas para cada etapa do projeto; c) definição de atores e suas experiências e habilidades; d) alocação de recursos; e) atribuição de tarefas aos atores; f) De uma forma geral, o gerente de projetos pode acompanhar e monitorar todo o processo de desenvolvimento de um projeto de software. É desenvolvido em JAVA, tem seus dados guardados em base de dados relacionais e seu código fonte é aberto. Essas características motivaram a escolha deste ambiente.

Neste ambiente foi modelado um processo de desenvolvimento baseado no "*RUP for small projects*", que se é um processo em projetos pequenos. O processo original possui sete etapas bases, no entanto, para este estudo foram utilizadas apenas seis, descritas a seguir e ilustrado na Figura 1.

- Requisitos: Etapa onde são coletados os requisitos principais do projeto e definidos seus limites, ou seja, onde o projeto é mensurado. Nesse ponto, os itens elencados servem principalmente para dar uma margem de custo e para fazer uma linha de entendimento entre o desejo do cliente e desenvolvedores.

- Análise e Projeto: Etapa onde os requisitos serão interpretados e definidos em termos do sistema a ser desenvolvido. Todos os artefatos gerados nessa etapa têm como principal objetivo facilitar a tarefa dos desenvolvedores de abstrair as funcionalidades inseridas nos requisitos e transformá-las em um sistema.

- Implementação: Essa etapa propõe definir, criar e organizar o código fonte do sistema, construindo objetos e todos os arquivos pertinentes ao funcionamento do mesmo. Também prevê a execução de testes orientados aos desenvolvedores como testes de unidade.

- Testes: Nesta etapa é atestada a qualidade do projeto. Com base nos testes é possível se ter certeza do correto funcionamento do software desenvolvido e se atende a todos os requisitos listados pelos *stakeholders.*

- Gerência de Mudança: Nesta etapa são controladas as inúmeras versões e objetos que são criados ou atualizados no projeto atual, contribuindo para que não haja conflitos com os artefatos e códigos gerados. Pode-se ter uma ideia de quem criou ou atualizou cada pequeno pedaço do projeto, além de que a gerência de mudança dispõe de todos os artefatos para a completa execução do projeto.

- Gerência de Projeto: Nessa etapa podemos ter o controle da gerência de risco, planejamento das iterações, execução e monitoria do projeto. Essa etapa para o RUP não visa gerenciar pessoas (contratações e alocações), orçamentos ou contratos e fornecedores.

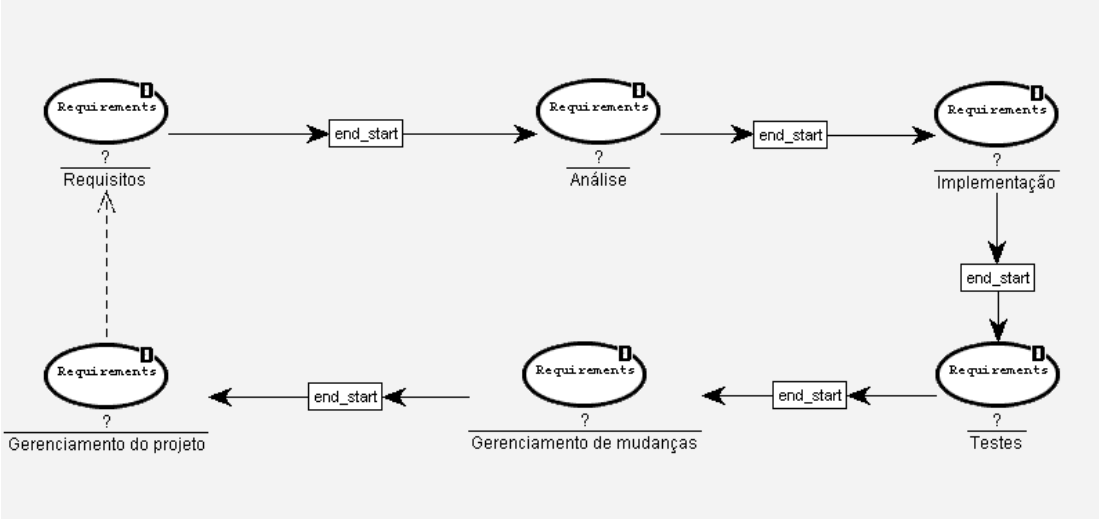

Figura 1- Esquema gráfico do Processo de desenvolvimento

O processo ainda possui algumas características variáveis que foram definidas como padrões e estão em todos os projetos que o utilizam, essas características eventualmente estão espalhadas nas seis etapas contidas no processo e são elas:

- Tarefas: Foram definidas tarefas de suma importância para o processo conforme mostra o quadro abaixo. Essas atividades estão divididas entre as etapas já definidas do processo, algumas atividades podem não ser executadas em alguns projetos de menor porte.

- Tempo Estimado para Execução: Cada tarefa definida no processo e que seja executada em determinado projeto tem pré-definidos intervalos de tempos estimados para sua execução que vão de "Otimista, Médio e Pessimista", conforme o Quadro 1.

- Participantes: Foi definido um número de participantes mínimo e máximo para cada etapa do processo, sendo que esse número pode variar de acordo com os projetos.

- Coeficientes de Experiência: Foram definidos valores de experiência para cada um dos participantes do projeto, também há uma variação dos participantes de cada etapa, nem sempre todos os participantes possíveis tomam parte nas tarefas de uma determinada etapa.

Etapas	Tarefas	Tempo		
		Otimista	Médio	Pessimista
Análise	Definição da Arquitetura	1	2	4
	Design da base de dados	2	4	6
	Design de Interfaces de usuário	3	7	9
	Estruturar Casos de Uso	2	5	8
Gerenciamento de Mudanças	Revisar as requisições de mudanças	1	3	7
	Submeter as Requisições de mudança	2	5	9
Gerenciamento do Projeto	Identificar e avaliar riscos	1	2	5
	Planejar fases e iterações	2	4	7
	Programar e atribuir atividades	1	3	6
Implementação	Estruturar modelo de Implementação	1	3	5
	Implementar design dos elementos	1	3	5
	Planejar integração dos sistemas	2	4	8
	Revisar Código	2	3	5
Requisitos	Descobrir atores e caso de uso	2.5	5	7
	Detalhar Requisitos	4	8	10
	Elencar Stakeholders	0.5	1	2
Testes	Determinar os resultados dos testes	3	6	8
	Estruturar a abordagem dos testes	2	5	7
	Executar o conjunto de testes	3	7	9
	Implementar conjunto de testes	2	4	6

Quadro 1 - Relação das tarefas do processo de desenvolvimento

3.2 A Base de Dados Sintética

Para a pesquisa foi criada uma base de dados sintética simulando dados históricos de projetos de software. A amostragem da base de dados é de cem projetos de software. Cada projeto é formado por um conjunto de quarenta e sete variáveis. Para isso, foi levado em consideração projetos que seguissem os padrões do *"RUP-for small projects"* em seu processo de desenvolvimento. Cada projeto da base de dados é formando pelas seis etapas descritas anteriormente, além de variáveis com valores parciais e finais. Algumas variáveis são dependentes, assim como podem exercer influência sobre outras. Como principal semente

para os valores aleatórios que compõe as variáveis da base de dados foram utilizados intervalos de valores possíveis para as mais diversas características (número mínimo e máximo de participantes em determinadas etapas, experiência mínima e máxima, entre outras características) do projeto modelado.

Os cálculos utilizados para geração dos dados simulados são, em sua maioria, combinações de fórmulas empíricas que resumem as possibilidades de comportamento de um projeto ao longo de seu ciclo de vida. Esses elementos visam a aleatoriedade dos eventos nos projetos da base e também um relacionamento lógico entre os itens da base de acordo com seus valores, como por exemplo, a relação de experiência da equipe e o esforço real em uma determinada etapa (relação inversamente proporcional uma vez que uma equipe experiente tende a ter um esforço menor para a conclusão da etapa), de forma que alguns conceitos e heurísticas foram levados em conta para cálculos de algumas variáveis.

A base de dados está dividida em três grupos de variáveis para cada projeto: *Identificação do projeto*, *Etapas do processo* e *Valores finais do projeto*. A seguir um resumo com as principais definições das variáveis utilizadas, separadas por grupo:

- **Identificação do Projeto:** Grupo onde estão variáveis que identificam os projetos listados.
    - **Name (nome do projeto):** serve para facilitar a visualização e comparação de valores. É formado por uma palavra "Projeto-" mais o número do projeto (1-100).
    - **Started Date (data de início):** importante para fixar uma linha temporal dos projetos, mas tem cunho meramente informativo. As datas são geradas de forma aleatória com um algoritmo que seleciona um intervalo de data entre (08/1997 – 08/2011).

$$\boldsymbol{RANDOM}(08/1997\,,08/2011)$$

- **Etapas do Processo:** Grupo de variáveis que quantificam os resultados de cada etapa de um projeto. São parciais de cada projeto. Simulam os valores obtidos pelo grupo de atividades que formam a etapa. As variáveis que constituem esse grupo são:
    - **Staff Number (Número de Participantes):** definido pela função aleatória:

$$\boldsymbol{RANDOM}(\boldsymbol{min\_Participantes}, \boldsymbol{max\_participantes}\,)$$

    - **Staff Experience (Experiência do grupo):** a experiência é definida como um total da experiência de todos participantes e é calculada por uma função aleatória com valores possíveis dentro de um intervalo da experiência do grupo de participantes. Esse valor bastante variável, uma vez que nem sempre todos os possíveis participantes irão atuar em uma determinada etapa.

$$RANDOM(min\_Experiencia, max\_Experiencia)$$

- **Novelty (Novidade):** classifica o nível de inovação de uma etapa do projeto, ou seja, se as tarefas abordam assuntos inovadores e nunca antes explorados pela equipe de desenvolvimento do projeto. Esse nível vai do menor (1) ao maior (10).

$$RANDOM(1, 10)$$

- **Complexity (Complexidade):** classifica o nível de complexidade de uma etapa do projeto. Seus valores variam aleatoriamente do nível menor de complexidade (1) ao nível maior (10).

$$RANDOM(1, 10)$$

- **Difficulty (Dificuldade):** indica o nível de dificuldade de uma etapa, levando em consideração o conjunto de tarefas que formam essa etapa. O seu cálculo é feito de acordo com uma média da complexidade com o nível de novidade de cada etapa.

$$\left(\frac{Novidade + Complexidade}{2}\right)$$

- **Time (days) (Tempo em dias):** Calculado em dias, o tempo total é obtido através de uma fórmula que aleatoriamente retorna um tempo dentro de um intervalo definido pelo gerente de projetos, que vai do mais otimista ao mais pessimista para o grupo de atividades da etapa. Esse tempo também leva em consideração o coeficiente "experiência/dificuldade", calculado por uma média da experiência dos participantes pela dificuldade da etapa em questão, conforme demonstra a seguinte fórmula:

$$SE\left(\left(\frac{(Staff\ Experience + Difficulty)}{2}\right) > Difficulty\right)$$

$$então\ coef. = Random(0.2, 0.5)$$

$$senão\ coef. = Random\ (0.6, 1)$$

- Juntando esse coeficiente ao calculo do tempo é possível criar um cenário de aleatoriedade no tempo total entre os projetos e as diferentes etapas que os envolvem, tendo em vista que dificilmente um projeto será igual à outro em todos os aspectos que o envolve. Dessa forma, a fórmula do tempo de uma etapa é:

$$RANDOM(tempo_{otimista} * coef.\ , tempo_{pessimista} * coef.)$$

- **Real Effort (Esforço Real):** É o esforço necessário para o desenvolvimento do projeto. Está distribuído entre as etapas do projeto, quantificando o esforço real de cada uma. Para este cálculo, conforme mostra a equação abaixo, foi utilizado o cálculo de esforço que é a divisão entre os participantes de uma determinada etapa

pelo total de tempo necessário para execução dessa etapa. No caso da base de dados em questão, o esforço esta calculado em horas por isso a equação é multiplicada por seis, que seriam o número de horas produtivas em um dia de trabalho.

$$\left(\frac{\boldsymbol{Staff\ Number}}{\boldsymbol{Time\ (days)}}\right) * \mathbf{6}$$

- **Estimated Effort (Esforço Estimado):** Esse valor é definido no planejamento do projeto e nem sempre é cumprido à risca. Desta forma este valor varia dependendo da dificuldade da etapa e também do seu conjunto de tarefas definidas. Esse valor é calculado por uma função aleatória que seleciona um valor dentro de um intervalo definido que vai dos possíveis tempos "otimistas" a "normais" para cada etapa do projeto. Também leva em conta o numero de participantes possíveis para essa etapa é calculada em termos de horas conforme a fórmula que segue abaixo.

$$\left(\frac{\boldsymbol{RANDOM(min\_dias\_tarefas, max\_dias\_tarefas))}}{\boldsymbol{num_{participantes}}}\right) * \mathbf{6}$$

- **Valores finais do projeto:** são os valores totais de cada projeto. São as variáveis utilizadas no processo de mineração de dados e posteriormente nas Redes Bayesianas, elas são:
  - **Scenario (Cenários):** são status finais dos projetos que representam o início e fim do projeto. Nesses cenários são informadas as condições que o projeto foi iniciado e como terminou em relação ao tempo estimado para finalização do projeto. A geração de valores desse campo baseia-se em uma função aleatória para definir o início do projeto entre os possíveis valores ("Começou em Tempo", "Começou Atrasado") e para os valores finais do projeto referente ao seu término leva em consideração se o esforço estimado foi maior que o esforço real (Requerido) do projeto, ou seja, se o esforço estimado for maior ou igual ao real então significa que o projeto terminou a tempo, caso contrário, o projeto terá atrasado.
  - **Scenario ID:** Identificador numérico dos cenários.
  - **Size – Nominal / Size (Tamanho por classificação):** projetos de acordo com seu tamanho em termos de número total de pontos por função.
  - **Size – PF (Tamanho em pontos por função):** em termos de funcionalidades produzidas. Para o cálculo foi utilizado uma função randômica para trazer entre os limites mínimos e máximos definidos como tamanhos de projetos possíveis.

$$\boldsymbol{RANDOM}(\mathbf{30, 5000})$$

- **Accuracy (Precisão):** Valor que representa a precisão das informações do projeto como um todo (requisitos, funcionalidades).

$$\boldsymbol{RANDOM(1,10)}$$

- **Total Staff Experience (Total Experiência do Grupo):** calculado em função da soma das experiências dos participantes de cada etapa.

$$\left(\sum \boldsymbol{Staff\ Experience}\right)$$

- **Total Difficulty (Dificuldade do projeto)**: calculado em função da soma das dificuldades de cada etapa e dividido pelo numero de etapas.

$$\left(\sum \boldsymbol{difficulty}\right)\boldsymbol{/6}$$

- **Novelty (Novidade):** calculado pela soma dos valores do nível de novidade de cada etapa e por fim divido por 6.

$$\left(\sum \boldsymbol{novelty}\right)\boldsymbol{/6}$$

- **Complexity (Complexidade):** calculado pela soma dos valores do nível de complexidade de cada etapa e por fim divido por 6.

$$\left(\sum \boldsymbol{complexity}\right)\boldsymbol{/6}$$

- **Total Staff (Numero total de participantes):** calculado pela soma do numero de participantes de cada etapa.

$$\left(\sum \boldsymbol{num\ staff}\right)$$

- **Estimated Effort (Esforço total estimado):** calculado pela soma dos valores de esforço estimado para cada etapa.

$$\left(\sum \boldsymbol{est.effort}\right)$$

- **Real Effort (Esforço real do projeto):** calculado pela soma dos valores de esforço real para cada uma de suas etapas.

$$\left(\sum \boldsymbol{real\ effort}\right)$$

- **Total time (hours) (Total em horas):** calculado pela soma dos valores de dias que cada etapa levou para ser finalizada, vezes o número de horas produtivas definidas para cada projeto por dia (6).

$$\left(\sum \boldsymbol{time}\right) * \boldsymbol{6}$$

- **Total time (days) (Total em dias):** calculado pela soma dos valores de dias que cada etapa levou para ser finalizada.

$$\left(\sum \boldsymbol{time}\right)$$

A base de dados contém uma série de informações sobre os projetos, porém como a coleta de dados nem sempre é bem definida, padronizada e executada, também foram inseridos ruídos. Dessa forma para melhorar a qualidade dos dados foi aplicada técnicas de mineração de dados.

3.3 Ferramentas para Mineração de Dados

Para aumentar a qualidade dos dados históricos de projetos de software foi necessário estudar e selecionar uma ferramenta de mineração de dados. Neste tipo de ferramenta é possível ser definido um conjunto de dados e aplicar uma série de técnicas para eliminação de ruídos e descoberta de padrões. Com esse objetivo foram estudadas as seguintes ferramentas Weka, Rapidminer, PolyAnalyst. A ferramenta Weka se mostrou a mais vantajosa para uso, devido a ser uma ferramenta aberta (*open source*), com API de desenvolvimento totalmente disponível, o que para trabalhos futuros possibilita a construção de um modelo igual ao executado nesse estudo, porém de forma totalmente automatizada, além das facilidades de uso e do manual disponível.

O Weka (*Waikato Environment for Knowledge Analysis*) é um pacote desenvolvido pela Universidade de Waikato, em 1993, com o intuito de agregar algoritmos para mineração de dados na área de inteligência artificial. O software é licenciado pela *General Public License* sendo, assim, possível a alteração do seu código-fonte.

Possui uma série de heurísticas para mineração de dados relacionada à classificação, regressão, clusterização, regras de associação e visualização, entre elas: J48 (Decision Tree), K-means, NaiveBayes, Linear Regression, IB1, Bagging, LogistBoot, Part, Ridor, ID3 e LMT.

O processo de extração do conhecimento do Weka segue praticamente o processo seguido pelos conceitos de mineração de dados e é composto por cinco etapas: a) **Seleção**: etapa de escolha da base de dados a ser analisada. b) **Pré-processamento**: etapa de "limpeza" dos dados, ou seja, reduzir discrepâncias de valores com ruídos e corrigir inconsistências. c) **Transformação**: transformação de dados, estes são modificados ou transformados em formatos apropriados à mineração, que pode por agregação, generalização, normalização, construção de atributos ou redução de dados. d) **Mineração de dados**: etapa de utilização de técnicas e algoritmos. e) **Interpretação**: etapa de análise dos resultados obtidos por meio da mineração dos dados, e a partir dos quais se adquire o conhecimento. O Weka também conta com uma opção chamada "*KnowledgeFlow",* onde é possível montar um fluxo de mineração dados.

O primeiro passo para a construção do fluxo de mineração de dados é o carregamento da base de dados, então após essa definição deve-se ir atrás de quais nodos (técnicas e algoritmos) serão utilizados. Tendo decidido quais técnicas e algoritmos serão utilizados o usuário deve adicionar os nodos selecionados no quadro e ligá-los como um *workflow* comum de dados. Essa conexão dos nodos é sempre realizada informando quais tipos de dados serão passados de um nodo para outro, de forma que os nodos precisam seguir uma sequência de execução para que a mineração de dados seja aplicada corretamente. Depois de terminada a construção do "*KnowledgeFlow"* o usuário precisa iniciar o processo de mineração, acompanhar pelo *log* os status de execução e ao final do processo coletar os dados nos relatórios inseridos no fluxo.

3.4 Aplicando análise de dados em dados históricos de projetos

Com o objetivo de qualificar os dados da base de dados histórica executando uma limpeza dos dados e eliminando ruídos, a fim de atestar o impacto que análise de dados exerce em estimativas através de Redes Bayesianas, foram criados dois fluxos (processos) de mineração de dados utilizando a opção "*KnowledgeFlow*" da ferramenta Weka. Cada um dos fluxos criados utilizam diferentes técnicas de mineração de dados, as técnicas selecionadas foram os algoritmos *j48* e *SimpleKMeans*. Os algoritmos em questão foram escolhidos por serem bastante referenciados na literatura como boas opções para classificação de dados e eliminação de ruídos em grandes conjuntos de dados.

3.4.1 J48 – Árvores de Decisão

O algoritmo J48 configurado para construir uma árvore de nodos binários, cada folha contendo no máximo dois elementos. Foi utilizado um coeficiente de confiança baixo para evitar o alto número de ruídos que possam passar despercebidos pelo algoritmo, com isso também temos um número elevado de podas na árvore o que diminui bastante a amostragem final da base de dados a ser analisada.

Para criação do fluxo de mineração de dados utilizando o algoritmo de J48, conforme mostra a Figura 2, foram utilizadas os seguintes nodos no *KnowledgeFlow:*

- *CSVLoader:* Onde deve ser selecionada o arquivo da base de dados a ser utilizado. Quando executado o fluxo irá carregar e preparar a base.

- *ClassAssigner:* É o classificador geral de qualquer conjunto de dados utilizado no algoritmo, ele obrigatoriamente deve ser nominal. No caso da base de dados utilizada, foi selecionado o atributo "*Scenario*" da base de dados.
- *CrossValidationFoldMaker:* Serve para analisar o conjunto de dados e atestar sua validade, faz isso usando uma equação para calcular o valor do erro médio de seus testes. Neste experimento, esse teste foi configurado para ser repetido dez vezes. A cada repetição a precisão dos dados aumenta.
- *J48:* Após as classificações e testes, os dados são enviados para o algoritmo J48, que irá classificá-los utilizando o algoritmo de árvore de decisão. Esse nodo irá gerar três saídas: uma do tipo "*graph*" (possibilita que sejam analisados de forma gráfica os nodos da árvore de decisão) e duas saídas do tipo "*batchClassifier*" que são "*ClassifierPerformanceEvaluator*" (permite avaliar o desempenho do algorítmo) e "*PredictionAppender*" (permite analisar os dados através de gráficos).

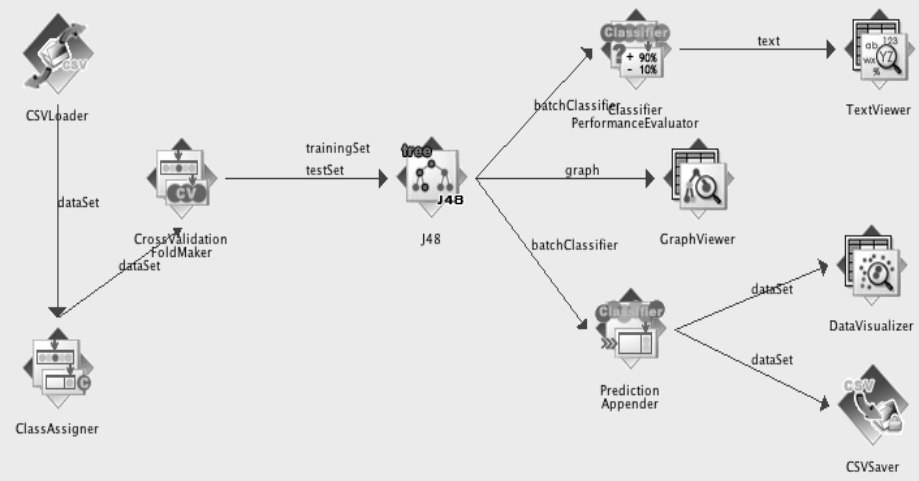

Figura 2 - Fluxo de mineração de dados com J48.

3.4.2 SimpleKMeans – K-means

O algoritmo *SimpleKMeans* utilizado no experimento foi configurado usando o cálculo de distância euclidiana, também foi definido que o algoritmo substitua valores faltantes por valores melhores qualificados pelas médias de suas proximidades em relação ao centroide, o número de iterações para o algoritmo foi definido em 500 e por fim foram limitados três clusters.

Para a criação do fluxo de mineração de dados utilizando o algoritmo *SimpleKMeans*, conforme mostra a figura 9, foram utilizados os seguintes nodos:

- *CSVLoader*: Onde deve ser selecionada o arquivo da base de dados a ser utilizado. Quando executado o fluxo irá carregar e preparar a base.

- *CrossValidationFoldMaker*: Serve para analisar o conjunto de dados e atestar sua validade, faz isso usando uma equação para calcular o valor do erro médio de seus testes. Em nosso experimento foi configurado que esse teste deveria ser repetido dez vezes. A cada repetição a precisão dos dados aumenta.
- *SimpleKMeans*: Após os dados serem validados, são enviados para o algoritmo *SimpleKMeans* que irá classificá-los utilizando o algoritmo de *K-means* com distância euclidiana. Esse nodo irá gerar três saídas, uma "*text*" e duas saídas para o "*batchCluster*" que vão para os nodos "*ClassifierPerformanceEvaluator*" e "*PredictionAppender*".

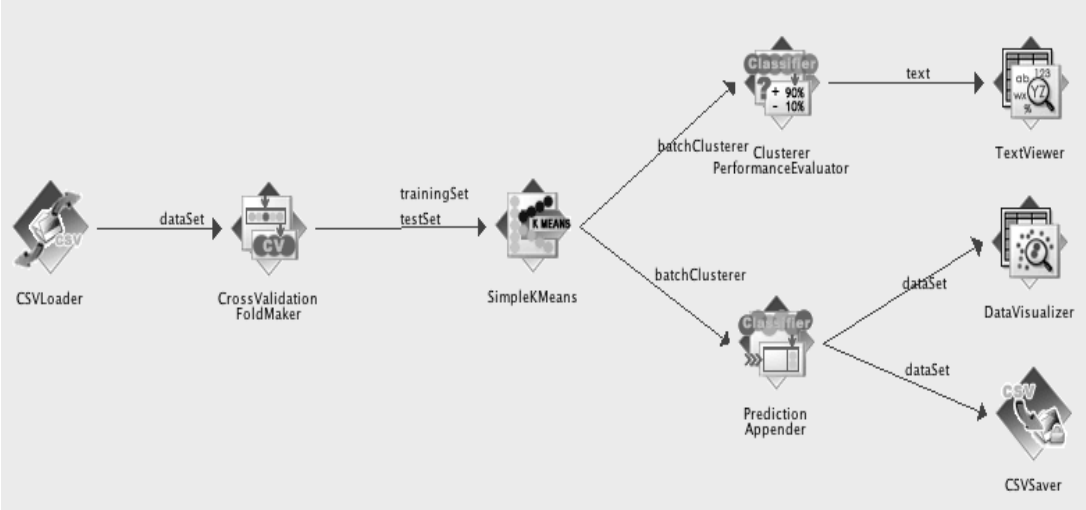

Figura 3 - Fluxo de mineração de dados com algoritmo K-Means.

3.5 Ferramentas para utilização de Redes Bayesianas

Para uso e manipulação de Redes Bayesianas foi utilizada a ferramenta Genie. A escolha dessa ferramenta se deu principalmente por possuir uma versão bastante robusta em termos de funcionalidade, por ser gratuita, por possuir uma biblioteca para desenvolvimento (SMILE). Criado em 1998, pela *Decision Systems Laboratory,* a Genie se mostra uma das melhores opções gratuitas para modelagem de Redes Bayesianas.

3.5.1 Treinando as Redes Bayesianas

Para treinar as Redes Bayesianas com diferentes bases de dados históricas foram necessários alguns passos na ferramenta Genie:

- **Carregar a Rede Bayesiana**: É o primeiro passo, deve ser selecionada qual Rede será utilizada para os testes.

- **Carregar os dados**: Então se deve carregar os dados da base de dados para dentro da ferramenta Genie.
- **Discretização dos dados**: Primeiramente os dados devem ser discretizados, assim facilita o treinamento das Redes. Na própria discritização a rede reconhece os intervalos que cada valor ira fazer parte, podendo ser configurado mais ou menos intervalos de valores para um nodo da rede, nesse estudo foram utilizados três intervalos de valores.
- **Associar valores e treinar**: O último passo após definir os intervalos de cada nodo da Rede Bayesiana é associar as colunas da base de dados aos nodos que existem na Rede Bayesiana e então treinar as Redes.

3.5.2 Definindo a Rede Bayesiana para análise

A Rede Bayesiana utilizada no experimento foi incorporada dos modelos propostos pelo MODIST (*Models of Uncertainty and Risk for Distributed Software Development*), que é um método e ferramenta para ajudar os gerentes de projetos a tomarem decisões mais efetivas em relação a qualidade e risco dos projetos. A Rede Bayesiana utilizada visa agrupar alguns aspectos importantes para o desenvolvimento de projeto de software, principalmente em termos de estimativas como esforço, experiência necessária, tamanho do projeto dentre outros, a fim de auxiliar o gerente de projetos na sua tomada de decisão no início de cada projeto. Conforme se pode ver na figura 10 a Rede Bayesiana é composta por oito nodos.

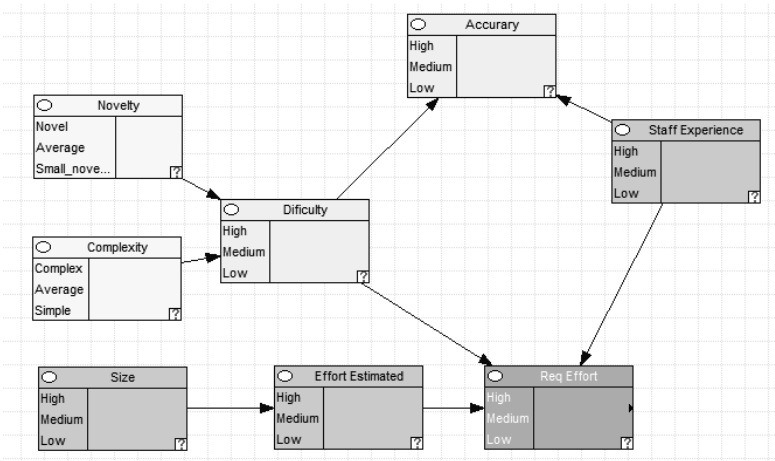

Figura 4 - Representação gráfica da Rede Bayesiana.

4 ANÁLISE DOS RESULTADOS

Para esta análise, a Rede Bayesiana foi testada em diferentes estágios. Primeiramente treinada com uma base de dados pura, que nunca sofreu qualquer tipo de tratamento de dados, então posteriormente foi treinada com uma base de dados que passou pelo processo de mineração de dados com uso do algoritmo J48 e por fim a mesma Rede Bayesiana foi treinada com uma base resultante do processo de mineração com o uso do algoritmo *SimpleKMeans*.

Depois do processo de treinamento da Rede Bayesiana com as diferentes bases de dados, foi selecionado um projeto de uma base de dados histórica, para análise na Rede Bayesiana, a fim de comparar os resultados obtidos pela Rede Bayesiana em seus diferentes treinamentos. Desta forma foi possível atestar o impacto que a análise de dados exerce sobre as Redes Bayesianas para diferentes tipos de predições, como esforço requerido, nível de experiência necessário da equipe e sobre o cenário de dificuldade do projeto.

O projeto foi escolhido de forma aleatória. Levou-se em consideração que um gerente de projetos poderia escolher ou estar vivenciando qualquer tipo de projeto e seu desejo seria obter ajuda a estimar esse projeto, fazendo uso de sua base de dados histórica juntamente com um processo de análise de dados e Redes Bayesianas. O projeto selecionado tem as seguintes características que serão analisadas pela rede:

Propriedade	**Valor**
Nome	Projeto-39
Numero de etapas	6
Cenário	Começou em tempo - terminou atrasado
Accurary (precisão)	1
Total Experiência	36
Dificuldade	Média
Complexidade	Média
Novidade	Média
Esforço Estimado	183
Esforço Real	258
Tamanho (Pontos por Função)	4869
Tamanho	Grande

Quadro 2 - Características do projeto selecionado para testes na Rede Bayesiana.

4.1 Rede Bayesiana – Treinada com base de dados pura

Essa Rede Bayesiana foi treinada com uma base de dados que nunca sofreu qualquer tipo de manipulação, seus dados foram carregados na rede e tiveram os seguintes resultados para os testes propostos:

- **Estimando o esforço Requerido**: A Rede Bayesiana na hora de estimar o esforço segundo os dados informados do projeto de testes, teve uma taxa de precisão baixa em relação ao que realmente aconteceu no projeto. A Rede marcou que provavelmente o projeto teria um esforço médio e não alto como foi o que ocorreu. A taxa de projetos com esforço alto foi de 38% enquanto os projetos de médio esforço foram a maioria com 43% conforme mostra a figura abaixo.

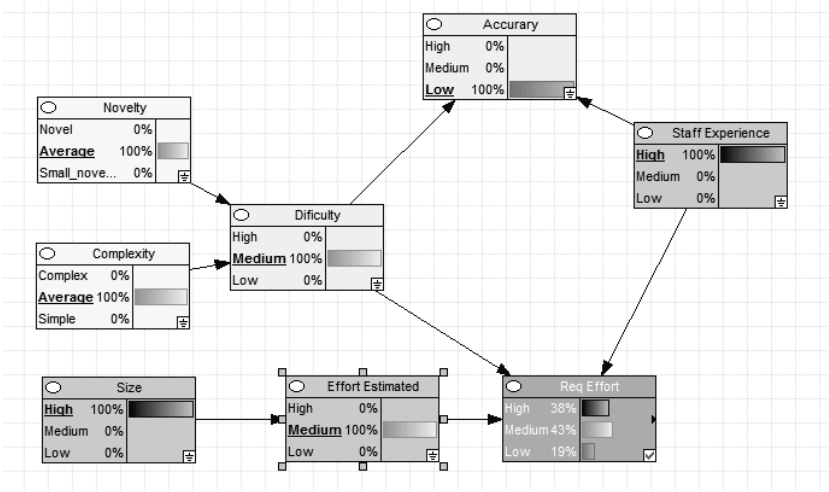

Figura 5 - Predição da Rede Bayesiana para Esforço Requerido.

- Estimando nível de experiência necessário em relação a precisão (accurary) do projeto: A Rede Bayesiana quando estimando o nível de experiência necessário para um projeto com o nível de precisão baixo, estimou que não seria necessário uma equipe experiente, o que o projeto de teste mostra estar incorreto, os resultados podem ser vistos na figura abaixo.

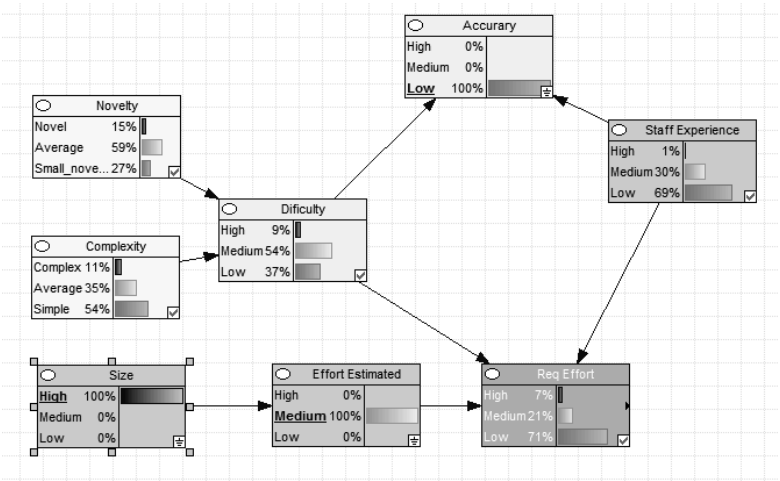

Figura 6 - Predição da Rede Bayesiana para Experiência necessária.

- Estimando nível de dificuldade de um projeto com equipe experiente: Quando definido uma equipe experiente, a rede estimou que praticamente todos os projetos

serão fáceis, ignorando a possibilidade de um projeto ser difícil e ter esforço alto, o que mais uma vez o projeto de teste mostra ser incorreto, conforme pode ser observado na figura 13.

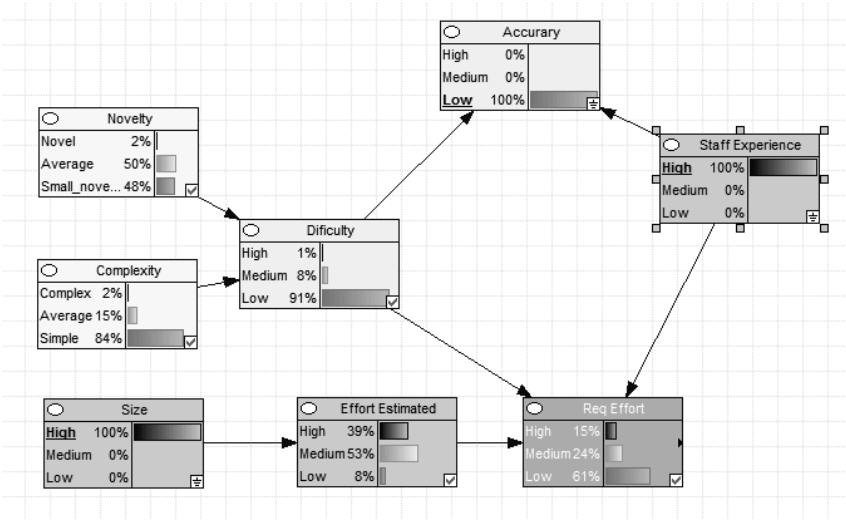

Figura 7 - Estimando a dificuldade estimada de um projeto em uma base pura.

Os testes com esse tipo de base de dados (pura, sem tratamento) demonstraram estimativas pouco confiáveis para o gerente de projetos, sendo a margem de acerto muito pequena e predições bastante imprecisas.

4.2 Rede Bayesiana – Treinada com base de dados tratada com o algoritmo J48.

Esta Rede em questão foi treinada com uma base que passou por um processo de mineração de dados com técnicas de arvores de decisão (J48) e coeficiente de baixa confiança (mais podas) para evitar um numero alto de ruídos. Após o treinamento, foram aplicados os testes com os seguintes resultados:

- Estimando esforço requerido: A aplicação deste teste com esse tipo de base também demonstrou que a Rede Bayesiana estimou de forma incorreta o esforço real necessário para o projeto do tipo selecionado. A precisão das estimativas não foram satisfatórias uma vez que também indicou esforço requerido como médio e não alto.

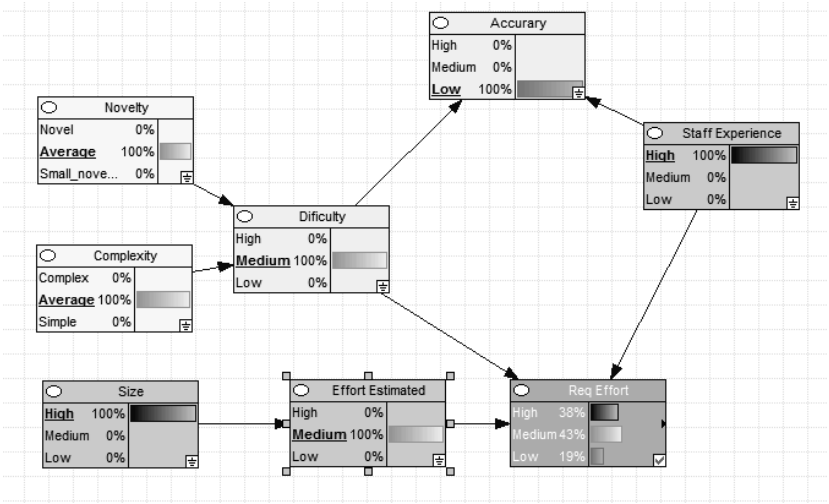

Figura 8 - Estimando esforço com algoritmo J48.

- Estimando o nível de experiência necessário em relação a Precisão (accurary) do projeto: Configurando a rede para estimar a experiência necessária em um projeto com baixa precisão, foi possível ver uma estimativa um pouco mais precisa em relação ao projeto de testes. Porém ainda pouco precisa e relevante para o gerente de projetos, uma vez que a estimativa mais aproximada era a de experiência média e não alta. É possível observar que houve uma melhora na precisão da estimativa com relação a base anterior, se aproximando mais de uma estimativa ideal nesse quesito.

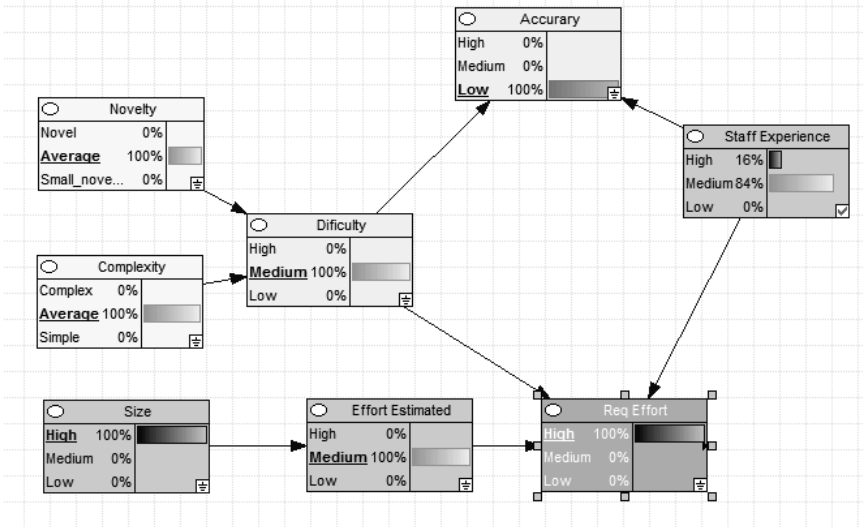

Figura 9 - Estimando a experiência necessária de um projeto.

- Estimando dificuldade de um projeto com equipe experiente: A precisão da estimativa sobre a dificuldade de um projeto de tamanho grande e esforço estimado médio, com uma equipe experiente, foi o teste mais preciso dessa base, acertando com 95% de precisão a estimativa para possível dificuldade enfrentada nesse tipo de projeto. Conforme pode ser observado na figura 16.

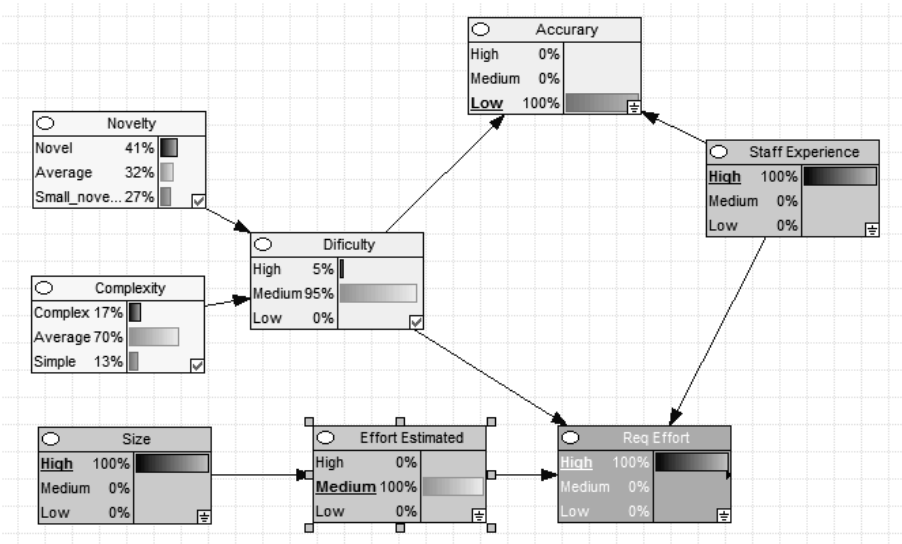

Figura 10 - Estimando a dificuldade em projetos.

Os testes com a base de dados usando tratamento com algoritmos J48 obtiveram uma margem de precisão maior em relação a base de dados pura, mostrando estimativas mais próximas as características do projeto de teste. Porém ainda não se mostraram estimativas confiáveis para um gerente de projetos.

4.3 Rede Bayesiana – Treinada com base de dados tratada com algoritmo K-Means.

Os testes a seguir foram aplicados a uma base de dados que passou por um fluxo de mineração de dados usando algoritmo K-Means, fazendo uso de distância euclidiana, para classificação e eliminação de ruídos. Os resultados dos testes podem ser conferidos a seguir:

- Estimando esforço requerido: Foi verificado que houve um aumento considerável na precisão desse tipo de estimativa quando gerada com essa base. Informando as configurações do projeto de teste a Rede Bayesiana retornou esforço requerido alto como estimativa mais provável, mostrando estar mais aproximada das características do projeto de teste.
- Estimando nível de experiência necessário em relação a precisão (accurary) do projeto: Novamente foi verificado que a rede teve uma precisão maior ao estimar o nível de experiência necessário para este tipo de projeto de baixa precisão em relação as duas bases anteriores com o mesmo teste. O aumento na precisão foi considerável.
- Estimando dificuldade de um projeto com equipe experiente: A precisão da estimativa de dificuldade a ser enfrentada em um projeto com as características do projeto de testes, foi altamente precisa definindo como sendo a dificuldade média a mais provável.

-

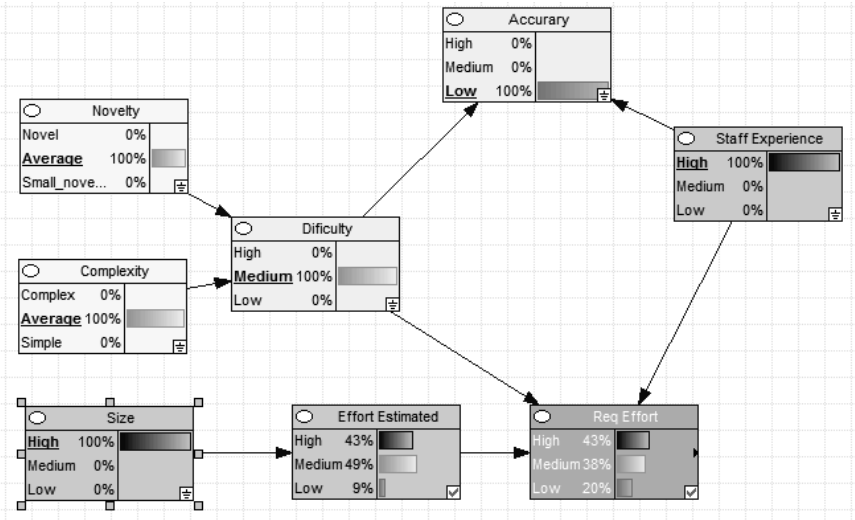

Figura 11 - Estimativa de esforço mais precisa com uso algoritmo de K-Means.

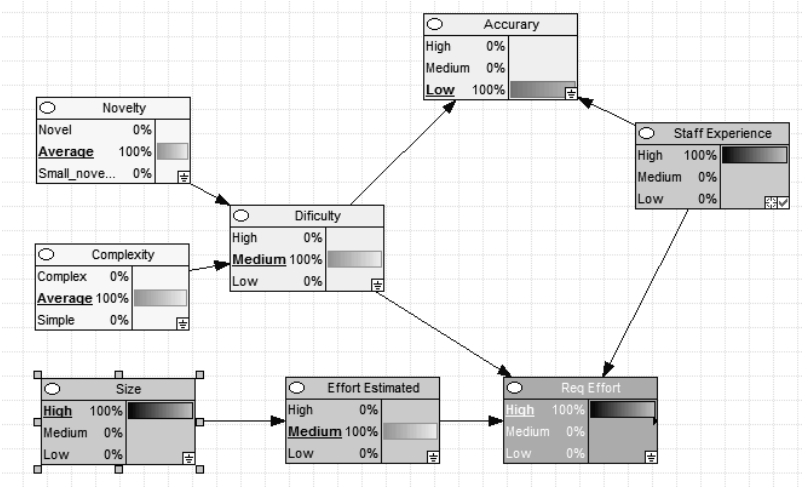

Figura 12 - Estimando nível de experiência para o projeto.

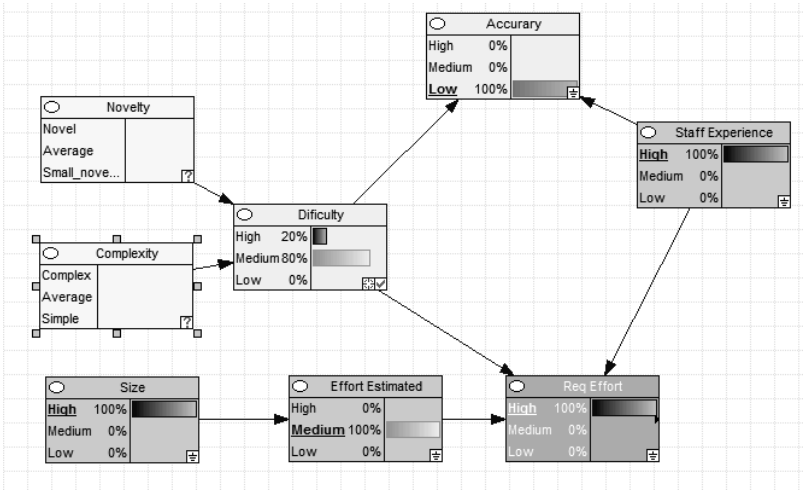

Figura 13 - Estimando a dificuldade para o projeto de teste.

Os testes com a base de dados tratada com o algoritmo de K-Means foram os que obtiveram maior sucesso para geração de estimativas tendo um nível maior de acerto e precisão nas estimativas geradas em relação aos testes anteriores com outras bases de dados, conforme demonstra o quadro comparativo abaixo.

Testes Executados na Rede Bayesiana			
	Correto	Predição Rede B.	Status do projeto
Base de dados pura sem uso de análise de dados			
Estimando Esforço	-	Médio	Alto
Estimando Experiência	-	Baixa	Alta
Estimando Dificuldade	-	Fácil	Médio
Base de dados minerada com algoritmo J48			
Estimando Esforço	-	Médio	Alto
Estimando Experiência	-	Médio	Alta
Estimando Dificuldade	SIM	Médio	Médio
Base de dados minerada com algoritmo K-Means			
Estimando Esforço	SIM	Alto	Alto
Estimando Experiência	SIM	Alta	Alta
Estimando Dificuldade	SIM	Média	Média

Quadro 3 - Demonstrativo do resultado dos testes aplicados.

Além do teste com o projeto demonstrado acima, foi realizado mais nove testes exatamente iguais, porém utilizando outros projetos também selecionados de forma aleatória da base de dados históricos de projetos de software. Os outros projetos utilizados possuíam as mais diversas características conforme pode ser visto na tabela abaixo:

Projetos								
Nome	Size	Esforço estimado	Esforco requerido	Experiencia	Complexidade	Novidade	Dificuldade	Acuracia
projeto-39	grande	médio	alto	alta	médio	médio	médio	baixa
projeto-5	grande	alto	alto	alta	médio	médio	médio	alta
Projeto-59	Pequeno	alto	alto	médio	alta	médio	médio	alta
projeto-44	médio	médio	alto	alta	médio	médio	médio	alta
projeto-68	médio	alto	alto	baixa	baixa	médio	médio	baixa
projeto-101	grande	médio	alto	médio	alta	médio	médio	alta
projeto-10	grande	baixo	alto	médio	alta	médio	alta	alta
projeto-13	grande	médio	alto	alta	alta	alta	alta	média
projeto-37	médio	alto	baixo	alta	baixa	baixa	baixa	baixa
projeto-43	grande	médio	médio	alta	médio	médio	médio	alta

Quadro 4 - Demonstrativo dos projetos de testes utilizados.

Os resultados obtidos em todos os testes seguem representados no quadro abaixo:

Resultados - 10 projetos de testes (N° Acertos das Predições)			
	Estimando Esforço	Estimando Experiência	Estimando Dificuldade
Base de dados - original	4	1	3
Base de dados - J48	7	4	4
Base de dados – K-means	9	7	9

Quadro 5 - Demonstrativo dos resultados gerais dos testes aplicados.

5 CONCLUSÃO

Tendo em vista o estudo realizado atesta-se que é de extrema importância que as empresas incentivem seus gerentes de projetos a sempre guardarem um histórico detalhado de todos os projetos os quais participem. Todavia apenas utilizar essa base em conjunto a um sistema de apoio a tomada de decisão (Redes Bayesianas) não configura estimativas mais precisas para o gerente de projetos, afinal não é possível atestar com 100% de certeza a qualidade dos dados inseridos na base. Provavelmente haverá ruídos inseridos na base que podem gerar estimativas imprecisas.

Conclui-se que é extremamente importante aplicar um processo de análise de dados à base de dados históricos com o objetivo de melhorar a qualidade para os sistemas de apoio a tomada de decisão e que a analise de dados tem um impacto bastante alto em estimativas geradas através de Redes Bayesianas aumentando aproximadamente 70% as chances de estimativas mais precisas. Pode-se ainda atestar que o uso do algoritmo K-means se mostrou mais apto em questões de eliminação de ruídos e em relação à classificação de dados, superando a precisão das estimativas da Rede Bayesiana em relação ao algoritmo J48. Isto se deve provavelmente devido ao comportamento de poda e redução de amostras que o J48 apresenta em sua execução.

Como trabalhos futuros podemos considerar o estudo comparativo de diferentes técnicas de mineração de dados para fins de analisar quais técnicas teriam melhor desempenho no treinamento de Redes Bayesianas, podendo ser desenvolvido um catalogo de técnicas e suas características.

Outra sugestão seria a coleta e utilização de uma base de dados de projetos reais em conjunto a dados reais de um projeto em andamento para realização dos mesmos experimentos.

REFERÊNCIAS

GUINZANI, Jonas Bonfante; SIMÕES, Priscila W. T.; MATTOS, Merisandra Côrtes de; BETTIOL, Jane. **Mineração de Dados em Redes Bayesianas Utilizando a API da *Shell Belief Network Power Constructor* (BNPC)**. Curso de Ciência da Computação – Universidade do Extremo Sul Catarinense
(UNESC). Criciúma, SC, Brasil. 2006.

HOLMES, Geoffrey; ANDREW, Donkin; WITTEN, Ian H. **WEKA: A Machine Learning Workbench**. Department of Computer Science University of Waikato, Hamilton, New Zealand. 2002.

JAIN, Sapna; AALAM, M Afshar; DOJA, M. N. **K-MEANS CLUSTERING USING WEKA INTERFACE.** Proceedings of the 4th National Conference. INDIACom-2010. 2010.

LIEBCHEN, Gernot; TWALA, Bheki; SHEPPERD, Martin. **Assessing the Quality and Cleaning of a Software Project Dataset: An Experience Report.** Brunel University, UK – EASE. 2006.

MODIST Method and Tool User Manual; MODIST Models of Uncertainty and Risk for Distributed Software Development IST-2000-28749. Deliverable 7.1. WP7/AGENA/TECH/DEL/1, Version 3. 2004.

WebAPSEE (2003). **http://www.webapsee.com.br**. 2003. Acessado em Novembro, 2011.

Site oficial do *Rup – for small projects*.
http://www.wthreex.com/rup/smallprojects/index.htm. Acessado em Novembro, 2011.

Site oficial do Modist. **http://www.agenarisk.com/resources/MODIST.shtml**. Acessado em Novembro, 2011.

Site oficial da ferramenta Genie. **http://genie.sis.pitt.edu/about.html**. Acessado em Novembro, 2011.