

Aprenda Java e Processo de software

Baseado no Personal Software Process (PSP) para Engenheiros - Parte 1

Programa 5

1. Requisitos do software

Regressão linear múltipla é uma maneira de ajustar de forma ótima uma linha para um conjunto de dados considerando múltiplas variáveis. A linha da regressão linear múltipla é a aquela em que a distância de todos os pontos para a linha é minimizada. Para n variáveis, a equação pode ser escrita como $y = \beta_0 + \beta_1 x + \dots + \beta_{n-1} x + \beta_n x$.

Considerando um conjunto de valores para n variáveis independentes x (em que cada variável independente é geralmente uma medida facilmente obtida e anterior ao que se deseja estimar) e um conjunto de valores para uma variável dependente y (relativa ao resultado desejado em função de variáveis x_n), é definido um sistema de equações lineares. Por exemplo, considere o conjunto de dados apresentado na Tabela 1. Nele temos três variáveis independentes: linhas adicionadas (w), linhas reutilizadas (x) e linhas modificadas (y). A variável dependente é o tempo de desenvolvimento (z). Assim, a forma final de nosso modelo de regressão seria $z = \beta_0 + \beta_1 w + \beta_2 x + \beta_3 y$.

Identificador do programa	Linhas adicionadas (w)	Linhas reutilizadas (x)	Linhas modificadas (y)	Tempo de desenvolvimento (z)
1	1142	1060	325	201
2	863	995	98	98
3	1065	3205	23	162
4	554	120	0	54
5	983	2896	120	138
6	256	485	88	61

Tabela 1. Medidas relativas ao tamanho e tempo de desenvolvimento de um conjunto de programas.

Uma forma de gerar um sistema de equações lineares é considerar cada instância de um conjunto de dados como uma equação. Desta forma, em um conjunto de dados com K instâncias, teríamos K equações lineares. Considere novamente a Tabela 1. Para aquele caso, podemos ter as seguintes equações lineares:

$$201 = \beta_0 + \beta_1 1142 + \beta_2 1060 + \beta_3 325$$

$$98 = \beta_0 + \beta_1 863 + \beta_2 995 + \beta_3 98$$

$$162 = \beta_0 + \beta_1 1065 + \beta_2 3205 + \beta_3 23$$

$$54 = \beta_0 + \beta_1 554 + \beta_2 120 + \beta_3 0$$

$$138 = \beta_0 + \beta_1 983 + \beta_2 2896 + \beta_3 120$$

$$61 = \beta_0 + \beta_1 256 + \beta_2 485 + \beta_3 88$$

Observe que, para β_0 , todos os valores foram definidos para 1. Isso se deve ao fato de que não temos uma variável independente associada a interceptação do eixo y (no caso, seria o tempo mínimo de desenvolvimento). Ao definir para 1, que é um valor neutro para a multiplicação, a resolução do sistema de equações proverá o valor β_0 mais ajustado para a interceptação do eixo y considerando os demais dados fornecidos.

Existem diversos métodos para resolver sistemas de equações lineares. Como nem todo sistema possui uma solução exata (como é o caso do exemplo), é útil considerar uma abordagem que produz uma solução aproximada. O método de resolução Linear Least Square é uma opção simples e viável para resolver esse sistema.

O método Linear Least Square decompõe o sistema de equações lineares em três matrizes A, B e X. A matriz A possui as variáveis dependentes, possuindo n colunas (uma coluna para cada variável independente) e K linhas (uma linha para cada instância). A matriz B possui os valores da variável dependente, armazenada em uma coluna. Os coeficientes β são representados pela matrix X, a qual desejamos calcular os coeficientes, sendo representada por uma coluna e quantidade de linhas igual à quantidade de coeficientes. Finalmente, o sistema é representado pela equação $AX = B$.

1	1142	1060	325
1	863	995	98
1	1065	3205	23
1	554	120	0
1	983	2896	120
1	256	485	88

Tabela 2. Matriz A.

201
98
162
54
138
61

Tabela 3. Matriz B.

Para calcular a matriz X, a seguinte fórmula pode ser utilizada: $X = (A^T A)^{(-1)} \cdot (A^T B)$, em que T representa a operação de transposição e $^{-1}$ representa a operação de inversão da matriz. Ao executar essa fórmula com os dados das Tabelas 2 e 3, temos como solução os valores apresentados na Tabela 4. Cada linha da matrix X corresponde a um coeficiente β .

6,7013365364	(β_0)
0,0783660367	(β_1)
0,0150413312	(β_2)
0,2460563326	(β_3)

Tabela 4. Matriz X.

No PSP, atualmente utilizamos apenas a quantidade total de linhas adicionadas e modificadas para estimar o tamanho. Com o emprego de uma regressão linear múltipla, é possível considerar mais de uma variável independente. Por exemplo, poderia ser utilizada a quantidade inicial de código base, as adições e modificações planejadas para o código base e as adições e modificações planejadas referente ao código específico do programa sob desenvolvimento.

2. Processo de software: Visão geral

2.1. Propósito

Guiar o desenvolvimento de programas modulares.

2.2. Critérios de entrada

- Descrição do problema.
- Formulário de Resumo de Planejamento de Projeto.
- Dados históricos de tamanho de programa e tempo de desenvolvimento.
- Modelo de regressão linear simples para estimativa de tamanho a partir de tamanho planejado.
- Modelo de regressão linear simples para estimativa de tempo de desenvolvimento a partir de tamanho planejado.
- Registro de Tempo.
- Registro de Erros.
- Padrão de estilo de codificação.

2.3. Atividades

- **Planejar**
 - Produzir ou obter requisitos.
 - Estimar o tamanho do programa.
 - **Estimar o tempo necessário para desenvolvimento.**
 - Preencher os dados de planejamento no formulário de Resumo de Planejamento de Projeto, incluindo o tamanho previsto para as partes do software, **o tamanho estimado do programa e o tempo estimado para desenvolvimento.**
 - Completar o Registro de Tempo.
- **Desenvolver**
 - Projetar o programa.
 - Implementar o projeto.
 - Compilar o programa, consertar e registrar todos os erros encontrados.
 - Testar o programa, usando testes automatizados e critérios de teste, consertar e registrar todos os erros encontrados.
 - **Refatorar o programa.**
 - Completar o Registro de Tempo.
- **Encerrar**
 - Completar o formulário de Resumo de Planejamento de Projeto com os dados de tempo efetivo, erros e tamanho, incluindo o tamanho efetivo das partes do software.

2.4. Critérios de saída

- Programa rigorosamente testado, considerando a satisfação de critérios de teste referente a comandos e desvios.
- Formulário de Resumo de Planejamento de Projeto preenchido com dados planejados, **estimados** e efetivos.
- Registro de Tempo completo.
- Registro de Erros completo.

3. Processo de software: Planejar

3.1. Propósito

Guiar o processo de planejamento do PSP.

3.2. Critérios de entrada

- Descrição do problema.
- Formulário de Resumo de Planejamento de Projeto.
- Dados históricos de tamanho de programa e tempo de desenvolvimento.
- Modelo de regressão linear simples para estimativa de tamanho a partir de tamanho planejado.
- Modelo de regressão linear simples para estimativa de tempo de desenvolvimento a partir de tamanho planejado.
- Registro de Tempo.

3.3. Atividades

- **Analisar requisitos**
 - Produzir ou obter os requisitos para o programa.
 - Garantir que os requisitos estão claros e sem ambiguidade.
 - Resolver qualquer questão quanto aos requisitos.
- **Definir partes do software**
 - Desenhar o software, organizando-o em partes para fins de planejamento e análise.
 - Preencher os dados de tamanho das partes projetadas.
- **Estimar recursos**
 - **Fazer sua melhor estimativa do tamanho do programa, utilizando o modelo de regressão linear ajustado a partir dos dados históricos.**
 - Fazer sua melhor estimativa do tempo necessário para desenvolver o programa, **utilizando o modelo de regressão linear ajustado a partir dos dados históricos.**
 - Preencher os dados de tempo estimado/planejado no formulário de Resumo de Planejamento de Projeto.
 - Usando como guia os dados totais de tempo consumido por fase do projeto (To Date %), distribuir o tempo total de desenvolvimento para cada fase do projeto do programa atual.

3.4. Critérios de saída

- Requisitos documentados.
- Formulário de Resumo de Planejamento de Projeto preenchido com o tamanho previsto para as partes do software, **as estimativas de tempo de desenvolvimento e de tamanho do programa.**

- Registro de Tempo completo.
- Registro de Erros completo.

4. Processo de software: Desenvolver

4.1. Propósito

Guiar o desenvolvimento de programas pequenos.

4.2. Critérios de entrada

- Requisitos.
- Registro de Tempo.
- Registro de Erros.
- Padrão de estilo de codificação.

4.3. Atividades

- **Desenhar / Projetar**
 - Revisar os requisitos e produzir um projeto para atendê-los.
 - Registrar no registro de erros qualquer erro encontrado nos requisitos.
 - Registrar o tempo no Registro de Tempo.
- **Programar**
 - Implementar o projeto, obedecendo o padrão de estilo de codificação.
 - Registrar no Registro de Erros qualquer erro encontrado nos requisitos e no projeto.
 - Registrar o tempo no Registro de Tempo.
- **Compilar**
 - Compilar o programa até que não existam mais erros de compilação.
 - Corrigir todos os erros encontrados.
 - Registrar erros no Registro de Erros.
 - Registrar o tempo no Registro de Tempo.
- **Testar**
 - Criar casos de teste, considerando critérios de teste de software.
 - Testar o programa até que os casos de teste executem sem erros e satisfaça os critérios de teste.
 - Corrigir todos os erros encontrados.
 - Registrar erros no Registro de Erros.
 - Registrar o tempo no Registro de Tempo.
- **Refatorar**
 - **Identificar indícios de problemas no programa (software smells).**
 - **Tratar os indícios de problema, aplicando operações de refatoração.**

4.4. Critérios de saída

- Programa rigorosamente testado, considerando a satisfação de critérios de teste referente a com comandos e desvios, e implementado de acordo com o padrão de estilo de codificação.
- Registro de Tempo completo.

- Registro de Erros completo.

5. Processo de software: Encerrar

5.1. Propósito

Guiar o processo de encerramento do PSP.

5.2. Critérios de entrada

- Descrição do problema e requisitos.
- Formulário de Resumo de Planejamento de Projeto com dados de tempo de desenvolvimento do programa.
- Registro de Tempo.
- Registro de Erros.
- Programa testado e executável que está em conformidade com o padrão de estilo de codificação.

5.3. Atividades

- **Garantir a consistência de dados de erros**
 - Verificar que todos os erros encontrados em cada fase foram realmente registrados.
 - Verificar se os dados de cada erro no Registro de Erros estão corretos e completos.
 - Verificar se a quantidade de erros inseridos e removidos em cada fase é razoável e correta.
 - Usando o melhor possível sua memória, corrigir qualquer dado omissos ou incorreto sobre os erros.
- **Garantir a consistência de dados de tempo**
 - Revisar o Registro de Tempo, procurando por erros e omissões.
 - Usando o melhor possível sua memória, corrigir qualquer dado omissos ou incorreto sobre os registros de tempo.
- **Medir tamanho do programa**
 - Medir o tamanho do programa completo e de suas partes.
 - Determinar o tamanho do código base, apagado, adicionado, reutilizado (que não seja o código base) e total.
 - Preencher os dados de tamanho no Formulário de Resumo de Planejamento de Projeto.

5.4. Critérios de saída

- Programa rigorosamente testado, considerando a satisfação de critérios de teste referente a comandos e desvios, e implementado de acordo com o padrão de estilo de codificação.
- Formulário de Resumo de Planejamento de Projeto preenchido com o tamanho previsto para as partes do software, **as estimativas de tempo de desenvolvimento e de tamanho do programa.**
- Registro de Tempo completo.
- Registro de Erros completo.

6. Entrega

Quando você completar a fase de encerramento, envie os dados do pacote da tarefa, código fonte e resultados de teste para o professor.

O pacote de tarefa deve conter os seguintes a seguir, na ordem apresentada:

- Formulário de Resumo de Planejamento de Projeto,
- Registro de Tempo,
- Registro de Erros,
- Listagem do código fonte do programa,
- Resultados dos testes.