

K implementaci IT - Zápis na zmeny

Lenka Obergriesová

October 2020

Obsah

1	Úvod	1
2	Dáta	1
3	Výpočty	2
3.1	Plná obsadenosť zmeny	2
3.2	Predošlé zapísanie na zmenu	2
3.3	Odhad pravdepodobnosti práce	2
3.4	Tolerancia	3
3.4.1	Výpočet tolerancie	3
4	Príklad	3
5	Poznámky	4
6	Výkonnosť	5

1 Úvod

Cieľom je implementovať systém, ktorý by doporučoval (obmedzoval) dĺžku zmeny zamestnanca na základe jeho výkonnosti.

Zamestnanec A, ktorý zapisuje na zmeny zadá do programu ID/meno zamestnanca B, ktorý sa chce na zmeny zapísať. Zamestnancovi A sa zobrazí "personalizovaný" kalendár zamestnanca B na dva týždne dopredu. V ňom vidí doporučená, ktoré dni by mohol zamestnanec B pracovať bez obmedzení, s obmedzeniami dĺžky zmeny, na ktoré dni už je zapísaný a na ktoré dni už ho nemožno zapísať pretože zmena je plne obsadená. Vid' obrázok 2.

2 Dáta

Aké dáta potrebujeme ukladať:

1. Kalendár zmien, každá zmena potom obsahuje:

- Koľko hodín práce je na danú zmenu potrebné zapísať (expertný odhad pracovníkov v sklade) (`hours_ideal`)
- Koľko hodín práce je už na danú zmenu zapísaných (`hours_registered`)
- Počet dní do tejto zmeny (`i`)
- Ktorí zamestnanci sú už na zmenu zapísaní a kedy sa zapísali, a ktorí chcú byť zapísaní, ale už im to nebolo umožnené
- Predpokladaný počet zamestnancov, ktorý sa budú chcieť na zmenu zapísať. Vid'. odhad pravdepodobnosti práce.
- Aktuálnu hodnotu tolerancie. Vid'. Tolerancia 3.4.

2. Zamestnanec:

- Koľko hodín práce ešte môže odpracovať (DPP - 300 hodín)
- Aká je jeho priemerná výkonnosť
- Na ktoré zmeny je zapísaný a na ktoré zmeny sa chcel zapísať, ale už mu to nebolo umožnené

3 Výpočty

3.1 Plná obsadenosť zmeny

Najprv sa vyhodnotí či už zmeny na najbližšie dva týždne nie sú obsadené. Porovná sa teda `hours_registered >= hours_ideal` a na základe výsledku sa vyhodnotí či už je zmena obsadená. Vid'. obrázok 2: červené štvorčeky.

3.2 Predošlé zapísanie na zmenu

Po zadaní id alebo mena zamestnanca B do systému, by sa malo skontrolovať na ktoré zmeny je už zapísaný. Vid'. obrázok 2: oranžové štvorčeky.

3.3 Odhad pravdepodobnosti práce

Potrebuje odhadnúť koľko zamestnancov sa bude chcieť na danú zmenu zapísať. Odhadneme to na základe minulých dát, teda sa pozrieme na zmenu v daný deň dva týždne dozadu - sčítame počet ľudí, ktorý pracovali a počet ľudí, ktorým sme to neumožnili a vydelíme počtom všetkých (?aktívnych) zamestnancov. Alternatívne môže zamestnanec A odhadnúť počet ľudí, ktorí budú mať o zmenu záujem. Na základe tohto údaju a za predpokladu, že ľudia sa budú zapisovať na zmenu lineárne dva týždne dopredu, dostaneme odhad počtu záujemcov o zmenu pre každý z desiatich dní, počas ktorých prebieha zápis (`total_of_people_expected_daily`). Teda napr. počítač/zamestnanec A odhadne že o danú zmenu bude mať záujem 120 ľudí - a teda za predpokladu lineárnych zápisov, vieme, že každý deň sa na zmenu príde zapísať približne 12 ľudí.

3.4 Tolerancia

Tolerancia je hraničná hodnota výkonnosti; ak má zamestnanec vyššiu priemernú výkonnosť než je aktuálna hodnota tolerancie tak sa môže zapísať na zmenu bez obmedzení ak nie, tak je mu povolené pracovať len kratšiu dobu (maximálne 4 hodiny).

Na základe simulácii, môžeme povedať, že nemá zmysel uvažovať hodnotu tolerancie viac než cca. 1,065.

3.4.1 Výpočet tolerancie

Počiatočná tolerancia bude 1,065; Ak by sme mali v určitý čas zapísaných menej ľudí, než by bolo optimálne hodnotu tolerancie prenásobíme koeficientom (menším než jedna), aby sa mohlo zapísať bez obmedzení hodín viac ľudí. V opačnom prípade by to fungovalo obdobne, ak by sme mali zapísaných viac ľudí než je optimálne, tak by sme prenásobili toleranciu koeficientom (väčším než jedna), avšak tolerancia by nikdy nemala byť väčšia než 1,065. Zároveň by minimálna hodnota tolerancie mala byť nastavená na hodnotu 0.5, keďže aj najmenšia možná výkonnosť zamestnanca je nastavená na hodnotu 0.5.

Výpočet koeficientov

- Do premennej `tan_beta` uložíme koľko hodín by sme chceli v priemere zapísať za jeden zápisný deň, teda ideálny sklon priamky plnenia zmeny.

$$\text{tan_beta} = \text{hours_ideal}/10$$

- Do premennej `tan_alfa` uložíme aktuálny optimálny sklon priamky plnenia zmeny.

$$\text{tan_alfa} = \frac{\text{hours_ideal} - \text{hours_registered}}{i - \text{daily_registration_ratio}}, \quad \text{kde}$$

$$\text{daily_reigistration_ratio} = \min \left(0.99, \frac{\text{registration_i_daily}}{\text{total_of_people_expected_daily}} \right)$$

- Koeficient, ktorým budeme prenásobovať toleranciu potom dostaneme ako:

$$\text{coefficient} = \begin{cases} \text{tan_beta}/\text{tan_alfa}, & \text{ak } \text{tan_alfa} > 0 \\ 1, & \text{inak} \end{cases}$$

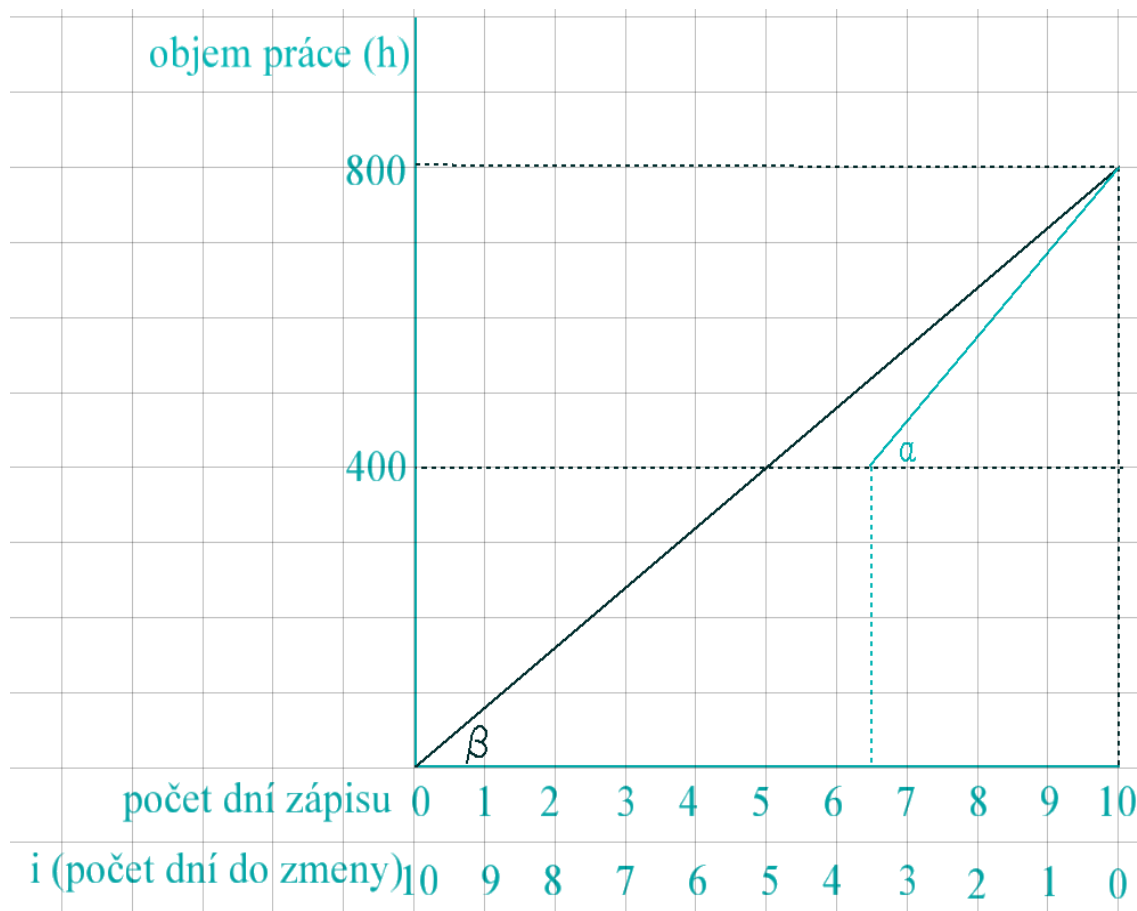
4 Príklad

- Zamestnanec B sa príde v pondelok zapísať na piatočnú zmenu (počet dní do zmeny = $i = 4$). Zamestnanec A zadá do systému id/meno zamestnanca B.

- Program skontroluje, že zmena nie je už plne obsadená a že zamestnanec nie je na zmenu už zapísaný.
- Následovne sa musí určiť či je vhodné mu dĺžku zmeny obmedziť alebo nie. Na základe minulých dát alebo expertného odhadu zamestnanca A sa domnievame, že o zmenu bude mať záujem cca. 140 ľudí, za predpokladu lineárneho plnenia zmeny počas 10 dní, to znamená, že denne sa zapíše cca. 14 ľudí = `total_of_people_expected_daily` (predpokladáme, že na zmenu sa bude zapisovať dva týždne dopredu - 10 pracovných dní).
- Ďalej na základe odhadu zamestnanca A vieme, že objem práce piatočnej zmeny bude 800 hodín (= `hours_ideal`) a zároveň vieme, že na zmene máme zatiaľ zapísaných 400 hodín (= `hours_registered`). Na zmenu ešte potrebujeme zapísať `hours_ideal - hours_registered` hodín.
- Vieme, že dnes (v pondelok) sa už na piatočnú zmenu zapísalo 7 zamestnancov (= `registration_i_daily`)
- `tan_beta = hours_ideal/10 = 800/10 = 80`
- `daily_reigistration_ratio = min(0.99,  $\frac{\text{registration\_i\_daily}}{\text{total\_of\_people\_expected\_daily}}$ ) = min(0.99,  $\frac{7}{14}$ ) = 0.5`
- `tan_alpha =  $\frac{\text{hours\_ideal} - \text{hours\_registered}}{\text{i} - \text{daily\_registration\_ratio}}$  =  $\frac{800 - 400}{4 - 0.5}$  = 114.286`
- `tan_alpha > 0  $\implies$  coefficient = tan_beta/tan_alpha = 80/114.286 = 0.70`
- Aktuálna hodnota tolerancie (napr. 1) bude pre násobená hodnotou koeficientu (0.7), nová tolerancia bude mať hodnotu 0.7. (TODO Toto je extrémny prípad, za normálnych okolností by nemalo dôjsť k takémuto extrémnemu výkyvu, v simuláciach sa koeficienty menili v ráde stotín)
- Zamestnanec B má priemernú výkonnosť napr. 1,2 a preto mu bude umožnené pracovať bez obmedzenia dĺžky zmeny.
- Zamestnanec chce pracovať plných 8 hodín, do premennej `hours_registered` však pripočítame hodnotu `8*1.2 = 9,6`.

5 Poznámky

- Najväčšou neznámou je pravdepodobnosť práce zamestnanca, teda, že nejaký zamestnanec sa rozhodne pracovať v daný deň.
- Horná medz tolerancie je 2, dolná medz je 0.5.



Obr. 1:

6 Výkonnosť

Máme 4 činnosti: naskladňovanie, rozhadzovanie, vyhľadávanie a balenie, u ktorých chceme namerať pre každého zamestnanca výkonnosti a následne ich vážene spriemerovať.

Po spustení súboru CRON.py (python-analyzy/projects/nabor/performance/CRON.py)

sa do databázy vloží tabuľka vykonnosti so stĺpcami: uzivatel_id, cinnost_id, celkovy_cas a vykonnost. Následne je potrebné vytvoriť novú tabuľku s dvomi stĺpcami:

uzivatel_id a celkova_vykonnost, pomocou SQL dotazu:

```
CREATE VIEW 'knihy-cme'.celkove_vykonnosti AS
SELECT uzivatel_id, SUM(vykonnost * hours) / SUM(hours) AS celkova_vykonnost
FROM 'knihy-cme'.uzivatel_vykonnost
GROUP BY uzivatel_id
```

Minimálna výkonnosť je nastavená na hodnotu 0.5 a maximálna je nastavená na hodnotu 2.

zam. id / meno

↓

Pa	Ut	St	Št	Pi	So	Ne
31.	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27

- * zamestnanca zápis na zmenu bez obmedzenia
- zmena je už plne obsadená
- * zamestnanca zápis na zmenu s obmedzením
- zamestnanec je už na danú zmenu zapísaný

Obr. 2: