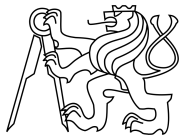
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



# ZADÁNÍ DIPLOMOVÉ PRÁCE

**Název:** Meta-learning na relačních datech
**Student:** Bc. Adéla Blažková
**Vedoucí:** Ing. Jan Motl
**Studijní program:** Informatika
**Studijní obor:** Znalostní inženýrství
**Katedra:** Katedra teoretické informatiky
**Platnost zadání:** Do konce zimního semestru 2018/19

## Pokyny pro vypracování

Data v databázi jsou v podobě mnoha tabulek, ale klasifikační algoritmy vyžadují na vstupu data v podobě jediné tabulky. Propositionalizace řeší tento rozpor převedením dat z podoby mnoha tabulek do podoby jedné tabulky.

Problémem propositionalizace ale je, že produkuje velké množství příznaků. A velké množství příznaků klade vysoké výpočetní požadavky jak při nápočtu příznaků, tak při klasifikaci.

Úkolem diplomové práce je navrhnout a implementovat tzv. meta-learning model, který odhadne, které příznaky jsou užitečné ještě před jejich kompletním nápočtem.

1) Vytvořte rešerši k meta-learningu pro účely klasifikace.
2) Identifikujte a popište meta-příznaky pro meta-learning nad relačními daty.
3) Natrénujte meta-learning model na poskytnutých relačních datech.
4) Vyhodnoťte přesnost meta-learning modelu.

## Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Jan Janoušek, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 25. června 2017

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra teoretické informatiky

Diplomová práce

# Meta-learning na relačních datech

***Bc. Adéla Blažková***

Vedoucí práce: Ing. Jan Motl

9. ledna 2018

# Poděkování

Ráda bych poděkovala svému vedoucímu práce Ing. Janu Motlovi za pravidelné konzultace, poskytnuté cenné rady a věcné přípomínky při tvorbě diplomové práce. Dále bych také ráda poděkovala svému příteli a rodině za podporu v průběhu celého studia.

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 9. ledna 2018 .....

### Odkaz na tuto práci

Blažková, Adéla. *Meta-learning na relačních datech.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018. Dostupný také z WWW: ⟨`http://site.example/thesis`⟩.

# Abstrakt

Předmětem této práce je návrh a implementace meta-learningového modelu, který predikuje optimální pořadí výpočtu příznaků při transformaci relačních dat do jedné tabulky. V návrhu řešení je stanovena metrika pro vyhodnocení optimálního pořadí příznaků, na základě které jsou vytvořeny 4 predikční modely. Zvolené algoritmy pro klasifikační a regresní modely jsou logistická regrese, algoritmus ElasticNet a algoritmus XGBoost. Použitými technologiemi jsou Jupyter Notebook (Python), databáze MySQL a nástroj Predictor Factory pro transformaci relačních dat. Výstupem práce jsou vytvořená trénovací meta-data a vyhodnocení přínosu jednotlivých predikčních modelů.

**Klíčová slova** Meta-learning, propositionalizace, meta-příznaky, meta-data, predikce, logistická regrese, ElasticNet, XGBoost, KI.

# Abstract

The aim of this thesis is to design and implement a meta-learning model that predicts the optimal order of calculation features when transforming relational data into a single table. The design part specifies a metric for evaluation of the optimal order of the features, based upon which four prediction models are created. Logistic regression, ElasticNet algorithm and XGBoost algorithm were

chosen to create classification and regression models. The technologies used for implementation of these algorithms were Jupyter Notebook (Python), MySQL database, and Predictor Factory tool for relational data transformation. The output of the thesis is represented by the created training meta-data and the evaluation of the contribution of each individual prediction model.

**Keywords** Meta-learning, propositionalization, meta-features, meta-data, prediction, logistic regresion, ElasticNet, XGBoost, KI.

# Obsah

# Seznam obrázků

# Seznam tabulek

# Úvod

## Motivace

S příchodem strojového učení vzrostl i význam dat. Data jsou dnes sbírána téměř ve všech oblastech. Na základě dat jsou vytvářeny různé predikce a analýzy, které jsou velmi přínosné v oblasti řízení a plánování.

Klasifikační a regresní algoritmy zpracovávající data jsou obvykle přizpůsobené pro práci s jednou tabulkou, kde jeden sloupeček reprezentuje cíl a ostatní sloupečky představují příznaky. Ve většině systémů jsou data ovšem uložena v relační podobě — několika tabulek, které jsou navzájem propojeny relacemi. Aby mohla být relační data zpracována algoritmy ze strojového učení, je nutné je převést do podoby jedné tabulky. Tento transformační proces se nazývá propositionalizace.

Během procesu propositionalizace se nad relačními tabulkami a jejich sloupečky napočítávají různé příznaky. Každý napočítaný příznak vytvoří jeden sloupec do výsledné tabulky. Některé příznaky mohou být výpočetně velice náročné a celý proces tak může běžet několik hodin až dní. Proto je žádoucí, aby se příznaky napočítávaly v pořadí podle užitečnosti pro následné zpracování klasifikačním nebo regresním algoritmem.

V tom může být nápomocný meta-learning, který sbírá zkušenosti z předchozích učení. Náplní této práce je vyvinout meta-learningový model, který bude predikovat užitečnost jednotlivých příznaků ještě před jejich výpočtem. Výstupem tohoto modelu je odhadované optimální pořadí příznaků pro nápočet.

Obsahem práce je teoretická část věnující se meta-learningu, možnostem měření výsledků predikce a vybraným algoritmům pro účely regrese a klasifikace. Rešeršní část je zaměřená na využití meta-learningu v oblasti klasifikace. Následuje návrh meta-learningového modelu pro predikci užitečnosti příznaků z procesu propositionalizace. Poslední kapitola obsahuje experimentální část, která zachycuje ladění a testování navrženého modelu.

## Cíl práce

Cílem této práce je navrhnout, implementovat a vyhodnotit meta-learningový model, který predikuje užitečnost příznaků v procesu propositionalizace a na základě predikce je vytvořeno odhadované optimální pořadí příznaků k výpočtu.

Teoretická část práce obsahuje přehled z oblasti meta-learningu, vybrané predikční algoritmy včetně možností jejich vyhodnocení a rešerši zaměřenou na využití meta-learningu v oblasti klasifikace.

Praktická část práce se zabývá návrhem meta-learningového modelu na relačních datech, optimalizací predikčních modelů a testováním.

Kapitola 1

# Meta-learning

Jedna z prvních zmínek o meta-learningu pochází od J. Rice z roku 1976 a ve strojovém učení se tento termín používá od 90.let minulého století. Jedním z důvodu je nárůst různých algoritmů v oblasti zpracování dat, časová náročnost optimalizace jejich parametrů, kvalita předzpracování dat a podobně.

Meta-learning se využívá různými způsoby, a proto pro něj existuje několik mírně odlišných definic. Společnou vlastností těchto definic je, že meta-learningový systém se přizpůsobuje a zlepšuje díky zkušenostem z předchozích učení. [1]

## 1.1 Oblasti meta-learningu

### 1.1.1 Ensemble metody

Jedná se o kombinování několika algoritmů místo zvolení pouze jednoho vhodného algoritmu pro daný problém. Redukuje se tak chyba při výběru algoritmu. Existuje několik metod spadajících do této oblasti, nejznámějšími jsou metody Bagging a Boosting. Výstupem ensemble modelu může být převažující třída v případě klasifikace, nebo průměr hodnot v případě regrese. [1]

1. Bootstrap Aggregation (Bagging) — tato metoda se používá pro snížení rozptylu predikčních algoritmů (takový algoritmus je např. rozhodovací strom, který má tendenci se přeučit na trénovacích datech). Ze vstupního datasetu se náhodně vygeneruje několik menších podmnožin dat. Pro každou podmnožinu dat je naučen predikční model. Výsledný model vznikne zprůměrováním výstupů z těchto dílčích modelů.
2. Boosting — skupina algoritmů, které z několika slabých modelů vytvoří jeden silný model. Jedná se o iterativní proces, během kterého se postupně trénují slabé modely na vstupním datasetu. V průběhu učení se dává větší důraz na data, která byla v předchozích učeních špatně klasifikována. Výstup slabých modelů může být do výsledného modelu

zkombinován pomocí (váženého) průměru nebo hlasováním. Do skupiny boosting algoritmů patří AdaBoost (Adaptive Boosting), Gradient Tree Boosting a XGBoost.

3. Stacked generalisation (Stacking) — několik predikčních modelů je sekvenčně trénováno na vstupním datasetu. Z výstupu těchto modelů je vytvořen nový dataset pro trénování výsledného modelu, typicky se používá lineární nebo logistická regrese.
4. Cascade generalisation — funguje sekvenčně. Po naučení predikčního modelu je jeho výstup přidán k příznakům vstupního datasetu, který je použit pro další učení.

Omezení meta-learningu u ensemble metod je, že se používají pouze pro jednu doménu problému.

### 1.1.2 Doporučování algoritmů

Velká část meta-learnigového výzkumu je soustředěna právě na doporučování vhodných algoritmů pro konkrétní datasety. Takový meta-learnigový systém nejčastěji zkoumá vliv charakteru vstupních dat na klasifikační přesnost algoritmu. Tento výzkum zahrnuje i pár specializujících se odvětví:

1. Optimalizace parametrů jednotlivých algoritmů, jelikož správné nastavení parametrů má na kvalitu výstupu z algoritmu velký vliv.
2. Predikce délky běhu, která slouží jako další kritérium při výběru vhodného algoritmu.

Výstup meta-learningu může vypadat následujícími způsoby:

1. Predikce nejlepšího algoritmu.
2. Ranking — ohodnocení kvality jednotlivých algoritmů.
3. Regrese — predikce relevance pro každý algoritmus.

### 1.1.3 Inductive transfer (Learning to learn)

Oblast meta-learningu sbírající zkušenosti během učení za účelem zvýšení efektivity při budoucích učeních na podobných problémech. Tento přístup k meta-learningu je využívaný pro optimalizaci hyperparametrů a neuronových sítí. Příkladem je paralelní učení neuronových sítí, během kterého sítě sdílí svou interní strukturu.

## 1.2 Meta-data

Meta-data představují nasbírané zkušenosti. Typ sbíraných meta-dat se liší v závilosti na druhu problému. Meta-příznaky by měly být vybrané tak, aby jejich získání nebylo výpočetně náročné.

Meta-příznaky využívané pro klasifikaci lze rozdělit do 5 kategorií:

1. Jednoduché — zahrnují obecné informace o datasetu, reflektují velikost základního problému.
   a) Počet řádků v datasetu
   b) Počet atributů
   c) Dimenze datasetu (poměr mezi počtem atributů a řádků)
   d) ...
2. Statistické — reprezentují numerické vlastnosti o rozložení dat v datasetu. Pomáhají rozlišovat stupeň korelace číselných atributů.
   a) Směrodatná odchylka
   b) Korelační koeficient
   c) Koeficient šikmosti
   d) Koeficient špičatosti
   e) ...
3. Information-theoretic — určují míru informace v diskrétních i spojitých atributech pomocí entropie nebo vzájemné informace (mutual information).
4. Landmarking — představují příznaky, které jsou napočítány pomocí jednoduchého algoritmu (obvykle ze strojového učení).
5. Model-based — pro popis dat jsou použity numerické parametry modelu některého jednoduchého klasifikátoru (např. rozhodovací strom).

## 1.3 Míry

### 1.3.1 Klasifikace

V definicích několika klasifikačních měr se používají tyto 4 jednotky na základě binární klasifikace na negativní a pozitivní třídy:

1. TN (True Negatives)
2. TP (True Positives)
3. FN (False Negatives)
4. FP (False Positives)

Výsledek binární predikce je znázorněn v tabulce 1.1. TN, resp. TP, odpovídá správně klasifikovaným položkám z negativní, resp. pozitivní, třídy. FN, resp. FP, určuje počet špatně klasifikovaných položek z pozitivní, resp. negativní, třídy.


Obrázek 1.1: Kontingenční tabulka reprezentující predikci do pozitivní a negativní třídy

Míry používané pro určení přesnosti klasifikátoru:

1. Přesnost (Accuracy) — vyjadřuje poměr správně klasifikovaných položek vůči součtu všech. Výsledek této míry může být zavádějící. Pokud jsou data nevyvážená a všechny se klasifikují do převažující třídy, výsledek této míry bude působit optimisticky, ačkoliv model je naprosto neprediktivní.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.1}$$

2. Přesnost (Precision) — tato hodnota určuje, kolik položek klasifikovaných do třídy jako pozitivní bylo klasifikováno správně. [2]

$$Precision = \frac{TP}{TP + FP} \tag{1.2}$$

3. Senzitivita (Sensitivity) — tato hodnota určuje, kolik položek patřících do pozitivní třídy bylo správně klasifikováno. [2]

$$Sensitivity = \frac{TP}{TP + FN} \tag{1.3}$$

4. F-Measure — je definována jako harmonický průměr senzitivity a přesnosti. [2]

$$F\text{-}Measure = 2 * \frac{Precision * Sensitivity}{Precision + Sensitivity}. \tag{1.4}$$

5. Specifita (Specificity) — vyjadřuje správnost predikce do TN. [2]

$$Specificity = \frac{TN}{TN + FP} \tag{1.5}$$

6. AUC (Area Under an ROC Curve)

   - ROC (Receiver Operating Characteristic) křivka — vyjadřuje vztah mezi senzitivitou a specifitou. Při vykreslení křivky na grafu svislá osa reprezentuje relativní četnost TP (senzitivita) a vodorovná osa relativní četnost FP (1 - specifita). Ideální ROC křivka stoupá zpočátku prudce vzhůru — to znamená, že většina instancí je správně klasifikována.

   K vyhodnocení klasifikačního modelu podle ROC křivky se používá míra AUC — plocha pod ROC křivkou. Pokud se plocha rovná hodnotě 1, jedná se o silný model se 100% senzitivitou i selektivitou. Pokud je obsah plochy pod hodnotou 0.5, pak model funguje hůře než náhodný model. [3]

Míry vyjádřené pouze pomocí dvojic TP a FP (přesnost), TP a FN (senzitivita), TN a FP (specifita), mohou mít zkreslený výsledek pro data s nevyváženým poměrem predikovatelných tříd.

### 1.3.2 Regrese

U všech měr v tomto výčtu platí, že čím blíže je jejich hodnota nule, tím vykazuje klasifikátor vyšší kvalitu predikce.

1. Mean absolute error (MAE) — tato míra se počítá jako průměr absolutních hodnot z rozdílu predikované a skutečné hodnoty. Hodnota této míry nám říká, jak velkou průměrnou chybu lze od predikce očekávat. [4]

$$MAE = \frac{\sum(|\hat{y}_t - y_t|)}{T} \tag{1.6}$$

2. Mean squared error (MSE) — míra vyjádřena jako průměr druhé mocniny rozdílu predikované a skutečné hodnoty. Díky tomu, že rozdíl hodnot je umocněn ještě před dělením, je tato míra více citlivá na veliké chyby. Použití této míry je tedy vhodné v případech, kdy jsou veliké chyby obzvlášť nežádoucí. [5]

$$MSE = \frac{\sum(\hat{y}_t - y_t)^2}{T} \tag{1.7}$$

3. Root mean squared error (RMSE) — rozšíření MSE o druhou odmocninu. Hodnota této míry je díky tomu blíže skutečné hodnotě chyby, a lépe se tedy interpretuje. [4]

$$RMSE = \sqrt{MSE} \tag{1.8}$$

4. Mean absolute percentage error (MAPE) — vyjadřuje průměrnou chybovost v procentech. Tato míra je vhodná, když se v datech nevyskytují

extrémy. Jelikož ve jmenovateli se používá skutečná hodnota $y_t$, platí zde omezení, že musí být tato hodnota různá od nuly. MAPE vrací extrémní chybovost i případě, že hodnoty $y_t$ jsou příliš blízko nule. Výhoda této míry je, že je nezávislá na měřítku dat. [6]

$$MAPE = \frac{1}{N}\sum(\frac{|\hat{y}_t - y_t|}{|y_t|}) * 100 \tag{1.9}$$

5. Pearsonův korelační koeficient — míra vyjadřující lineární vztah mezi dvěmi proměnnými. Hodnoty koeficientu se pohybují v rozmezí $[-1, +1]$. Hodnota -1, resp. +1, vyjadřuje negativně, resp. pozitivně, lineární vztah mezi proměnnými. Naopak nulová hodnota znamená, že dané proměnné jsou na sobě lineárně nezávislé. [7]
6. Spearmanův korelační koeficient — určuje sílu a směr monotónního vztahu mezi dvěmi proměnnými. Porovnává, zda sobě odpovídající hodnoty ze dvou proměnných rostou či klesají. Monotónní vztah je méně restriktivní než lineární. [8]

### 1.3.3 Míry důležitosti příznaků

1. Chi2 — tato míra se používá při testu nezávislosti dvou kategorických proměnných. Výsledek tohoto testu říká, jestli nezávislá proměnná ovlivňuje hodnotu závislé proměnné.

   V této práci je míra Chi2 využita pro získání závislosti mezi napočítanými příznaky a cílovým atributem z procesu propositionalizace. Čím vyšší závilost bude mezi příznakem a cílovým atributem existovat, tím je vyšší pravděpodobnost, že daný příznak bude užitečný pro klasifikační či regresní algoritmus.

   Hodnota Chi2 se počítá z kontigenční tabulky obsahující četnosti dvou kategorických proměnných. Detailní rozbor Chi2 včetně podmínek aplikovatelnosti je popsán v [9].

### 1.3.4 Anytime

1. KI — Míra se počítá na základě obsahu 2 ploch. V čitateli je plocha mezi optimální a náhodnou křivkou. Jmenovatel obsahuje rozdíl plochy pod predikovanou a náhodnou křivkou (v případě špatně predikujícího modelu tedy může jmenovatel vyjít i záporný). [10]

   Při vyhodnocení přesnosti implementovaného meta-learningového modelu v této práci může míra KI nabývat hodnot [-$\infty$;1], přičemž hodnota 1 značí perfektní model. Dolní mez je ovlivněna výskytem duplicitních příznaků, který způsobuje, že náhodná křivka nemusí být diagonála a tudíž dolní mez může být nižší než hodnota -1.

## 1.4 Regresní modely

1. Lineární regrese vyjadřuje vztah závislé proměnné na nezávislých proměnných vzorcem $y = b_0+b_1x_1+b_2x_2+...+b_px_p$ a snaží se minimalizovat kvadratickou sumu $\sum(y - \hat{y})^2$.
2. Lasso regrese — vychází z lineární regrese, kterou rozšiřuje o penalizaci příznaků, je tedy vhodná pro datasety s velkým počtem atributů. Tuto penalizaci řeší podmínkou, že $\sum_{i=1}^{p} |b_i| \leq s$ - tzv. L1 regularizací. Čím menší je hodnota s, tím více penalizujeme ostatní koeficienty z lineárního modelu — několik koeficientů pak musí být rovno 0. [11]
3. Ridge regrese — od Lasso regrese se liší v podmínce rozšiřující lineární regresi: $\sum_{i=1}^{p} b_i^2 \leq s$ - tzv. L2 regularizací. [11]
4. ElasticNet — jedná se o algoritmus pro lineární regresi, který kombinuje výhody z Lasso regrese a Ridge regrese. [12] Míra využití L1 penalizace a L2 penalizace je dána parametrem

$$\alpha : \alpha * L2 + (1 - \alpha) * L1 \tag{1.10}$$

5. XGBoost (Extreme Gradient Boosting) patří do skupiny boosting algoritmů, které jsou založené na učení s učitelem a iterativně optimalizují klasifikační/regresní model na trénovacích datech. XGBoost je implementací algoritmu Gradient boosting, který je založen na kombinaci několika slabých modelů, v případě XGBoost rozhodovacích/regresních stromů. V každé iteraci je přidán nový strom za účelem snížení predikční chyby. Výsledná predikce se získá sečtením skóre ze všech stromů. Výhoda algoritmu XGBoost spočívá v tom, že při vytváření rozhodovacích stromů používá paralelizaci, a je tedy výrazně rychlejší než algortimus Gradient boosting. [13]

## 1.5 Klasifikační modely

1. Rozhodovací strom — jedná se o algoritmus založený na učení s učitelem, parametry algoritmu jsou tedy optimalizovány na trénovacích datech. Struktura tohoto algoritmu se skládá z uzlů, hran a listů. Každý uzel obsahuje příznak, na základě jehož hodnoty se rozhodne, po jaké hraně pokračovat do dalšího uzlu. Listy obsahují konečnou klasifikaci.
2. Logistická regrese — je určena pro binární klasifikaci. Metoda je založena na logistické funkci (sigmoidě).

$$sigmoida = \frac{1}{1 + e^{-x}} \tag{1.11}$$

Vzorec pro logistickou regresi, jehož koeficienty $b_i$ jsou optimalizovány:

$$y = \frac{1}{1 + e^{-(b_0+b_1x_1+...+b_nx_n)}} \tag{1.12}$$

Výstupem je pravděpodobnost, se kterou instance patří do zvolené třídy, tedy hodnota od 0 do 1. Tato hodnota je finálně transformána na třídu 0 nebo 1.

Kapitola 2

# Využití meta-learningu pro účely klasifikace

## 2.1 Doporučovací nástroj pro výběr klasifikačního algoritmu

Nalezení klasifikačního algoritmu, který vrací nejlepší výsledky pro daný specifický dataset, bývá pro vývojáře bez dostatečných zkušeností časově náročné. Výsledek klasifikátoru ovlivňuje charakter vstupních dat a jak dobře splňuje dataset předpoklady pro použití daným klasifikátorem. Proto častým využitím meta-learningu bývá doporučení vhodného klasifikačního algoritmu pro vstupní dataset.

Touto problematikou se zabývá i práce Automatic classifier selection for non-experts [14]. V práci je představen open source meta-learning nástroj, který je integrován do systému RapidMiner, a slouží pro výběr vhodného klasifikátoru. Nástroj byl vytvořen za účelem ulehčení práce při vývoji pattern recognition systémů.

Učení meta-learningu je založeno na doporučování 9 klasifikačních algoritmů (SVM, MLP, Decision Tree, Random Forest, Naive Bayes atd.). Algoritmy byly vybrány tak, aby se lišily ve způsobu učení. Pro každý klasifikátor je vytvořen regresní model, který je naučen na trénovacích datasetech a jeho výstupem je odhadovaná přesnost klasifikátoru na vstupním datasetu. U každého regresního modelu byly optimalizovány parametry metodou grid search a 10-násobnou křížovou validací. Alternativou výstupu predikce je například tzv. ranking, který ohodnotí výkon klasifikátoru vůči ostatním klasifikátorům. Tato možnost ale nebyla zvolena, jelikož z pohledu uživatele je vhodnější jako výstup přímo odhadovaná přesnost.

Meta-příznaky napočítávané pro vstupní dataset byly rozdělené do pěti kategorií:

1. Základní popis datasetu (počet vzorků, počet atributů apod.)

2. Statistické příznaky
3. Příznaky z teorie informace (entropie)
4. Model-based příznaky — jako model byl zvolen Decision Tree a jeho vlastnosti jako počet listů, počet uzlů apod.
5. Landmarking příznaky — obsahující přesnosti rychle napočítaných klasifikačních algoritmů (Naive Bayes, Decision Node, Linear Discriminant Analysis ...).

Pro zlepšení výsledku predikce bylo využito ještě dvou fázového výběru nejlepších meta-příznaků. Nejprve byla aplikována metoda forward selection, která postupně přidává meta-příznaky do datasetu, dokud po daném počtu iterací neklesne predikční přesnost modelu. Ve druhé fázi byla aplikována metoda backward elimination za účelem odstranění málo užitečných příznaků z první fáze. Ani jeden ze zmíněných postupů ovšem nezaručuje optimální výběr meta-příznaků.

Přesnost predikce byla měřena pomocí RMSE (Root Mean Squared Error) a PMCC (Pearson product-moment correlation coefficient). Podle naměřených výsledků regresní predikce byly nejvíce užitečné landmarkovací příznaky, a poté výběr nejlepších meta-příznaků.

## 2.2 Predikce délky běhu klasifikačních algoritmů

Práce [15] se zabývá využitím meta-learning modelu pro predikci délky běhu klasifikačních algoritmů během trénování modelu. Prvotně je při výběru klasifikátoru pro konkrétní dataset důležitá jeho predikční přesnost. Pokud ovšem máme více klasifikátorů s podobnou predikční přesností, délka běhu je dalším důležitým faktorem pro výběr algoritmu.

Pro predikci hodnoty délky běhu se dají využít dva zdroje informací — hardwarová konfigurace počítače a vstupní dataset. Zde bylo využito pouze meta-dat ze vstupních datasetů. Hardware počítače je ovšem silným faktorem, který ovlivní výslednou hodnotu délky běhu — záleží na rychlosti CPU, počtu jader CPU, kapacitě RAM apod. Cílem proto není přesná predikce délky běhu, ale očekávaná hodnota délky běhu algoritmu na počítači s pevnou konfigurací hardware. Predikce má sloužit především pro porovnání časové náročnosti mezi klasifikačními algoritmy.

Do predikce délky běhu není zahrnuta optimalizace parametrů, protože nalezení optimálních hodnot pomocí metody grid search trvá déle než 24 hodin.

Cílem této práce bylo objevit vlastnosti datasetů, které ovlivňují délku běhu, a vytvořit predikční model. Pro naučení modelu bylo použito 50 datasetů z UCI Machine Learning repository [16] a 28 klasifikačních algoritmů. Pro predikci bylo testováno 8 regresních algoritmů.

Vstupní datasety, ze kterých byly vytvořeny meta-instance pro predikční model, bylo nutné nejprve předzpracovat:

1. Doplnění chybějících hodnot
2. Tranformace všech atributů na numerické hodnoty (binární příznaky)
3. Normalizace atributů
4. Redukce atributů
5. Nápočet meta-příznaků nad datasetem
6. Měření délky běhu klasifikačních algoritmů nad datasetem
7. Vytvoření meta-instance <meta-příznaky, klasifikační alg, délka běhu>

Dále byly napočteny statistiky nad jednotlivými numerickými atributy, to ovšem zapříčiní jiný počet meta-příznaků pro každý dataset. Datasety byly proto transformovány do standardního formátu s počtem $k$ numerických atributů, kde $k$ je hodnota od 2 do 7. K tomu byla využita dimenzionální redukce příznaků. Vedlejším efektem zmenšení dimenze je také snížení výpočetního času pro učení modelu.

Pro redukci příznaků bylo vyzkoušeno 7 (lineárních i nelineárních) technik — PCA (Principal Components Analysis), Factors Analysis, RBF PCA atd. Metody byly testovány na 10 náhodně vybraných datasetech a 28 klasifikačních algortimech. Pro každý dataset, redukční metodu a hodnotu $k$ byla změřena průměrná chyba při klasifikaci. Z experimentů se nejvíce osvědčila metoda Probabilistic PCA a hodnota $k$ rovno 4.

Při učení regresního modelu byla hodnota délky běhu zaznamenávána mírou CPU time, která měří dobu běhu algoritmu nezávisle na běhu ostatních programů, které též využívají CPU. Pokud například klasifikace běžela 2 hodiny, ale využila procesor jen na 75%, hodnota CPU time je 1.5 hodiny.

Důležitým aspektem pro úspěšnou predikci je nalezení správných meta-příznaků, které mají vysokou prediktivní sílu. Mezi často používaný příznak patří průměrná hodnota koeficientu šikmosti spočítaná nad celým datasetem. Průměrná hodnota ovšem nemá velkou diskriminační sílu (z průměru se ztrácí informace o jednotlivých atributech, přitom atributy s vysokou hodnotou koeficientu šikmosti mohou být při klasifikaci užitečné). Proto jsou koeficienty šikmosti a špičatosti spočítány pro každý ze 4 příznaků datasetu zvlášť.

Výčet napočítávaných meta-příznaků:

1. Počet instancí v datasetu
2. Počet tříd v cílovém atributu
3. Velikost datasetu po redukci atributů
4. Pearsonův lineární korelační koeficient pro všechny dvojice příznaků
5. Koeficient špičatosti– pro všechny 4 příznaky

6. Koeficient šikmosti – pro všechny 4 příznaky
7. Entropie
8. Korelace mezi atributy

Během ladění lineární regrese bylo zjištěno, že většina meta-příznaků má příliš vysokou špičatost oproti normálnímu rozdělení. To se dá někdy vyřešit například použitím logaritmu, odmocninou nebo druhou mocninou aplikovanou na data. Zde byla použita BoxCox transformace, která by měla přiblížit data normálnímu rozdělení. BoxCox formule obsahuje parametr $\lambda$, jehož optimální hodnota byla nalezena iterativním testováním přesností predikce.

Poté již byly na datech testovány jednotlivé regresní algoritmy:

1. Principal Components Regression
2. Partial Least Squares
3. Ridge Regression
4. Least Angle Regression
5. ElasticNet
6. K-nearest Neighbor Regression
7. Multi-Variate Adaptive Regression Splines (MARS)
8. Support Vector Regressions (SVR)

Mezi míry pro určení přesnosti predikce byly zvoleny RMSE (Root Mean Squared Error), MAE (Mean Absolute Error) a MAD (Mean Absolute Deviation). Při testování regresních modelů byla použita 10-násobná křížová validace. Chybovost regresních modelů byla porovnávána vůči predikční přesnosti algoritmu SVR, který obecně funguje dobře u většiny problémů.

Z experimentů vyšel nelépe regresní algoritmus MARS. Tento algoritmus měl při predikci nad 50 datasety průměrnou hodnotu RMSE 0.397, přičemž model SVR měl hodnoty RMSE 1.736.

Při testování algoritmu MARS nad 5 datasety, které nebyly součástí učení modelu, byly hodnoty RMSE 21.8, 4.7, 3.4, 17.9 a 19.0. Jelikož délka běhu testovaných datasetů se pohybuje mezi 0.01-3000s, lze považovat nejvyšší hodnotu RMSE 21.8 za stále rozumnou chybu.

## 2.3 Doporučení množiny klasifikátorů pro Moodle datasety

Další práce[17] zabývající se využitím meta-learningu pro predikci vhodných klasifikátorů je zaměřena pouze na datasety ze vzdělávacího systému Moodle. V systému Moodle se mohou studenti připravovat na jednotlivé kurzy, absolvovat testy, diskutovat o předmětech na skupinových fórech apod. Cílem je

predikovat výsledky studentů. Výstupem meta-learningového frameworku je množina klasifikačních algoritmů, u kterých se předpokládá nejvyšší predikční přesnost pro daný dataset.

Navržený meta-learningový framework se skládá z trénovací a predikční části. Pro naučení meta-learningového modelu bylo použito 32 datasetů, které vznikly sběrem dat o studentech používajících systém Moodle mezi rokem 2007 až 2012.

Ze vstupních datasetů byly spočteny meta-příznaky, které lze rozdělit do tří skupin:

1. Statistické příznaky — např. počet instancí, počet numerických atributů, počet kategorických atributů
2. Příznaky popisující složitost datasetu. Jedná se o pokročilejší matematické funkce, které lze použít pro charakteristiku dat. Tyto příznaky byly získány použitím The data complexity library (DCoL) [18]
3. Příznak reprezentující doménu, protože datasety ze systému Moodle se dělí do několika subdomén. Může se jednat o doménu report (obsahuje obecná data o chování studenta v systému Moodle), kvíz (obsahuje data o absolvovaných testech studenta) nebo o doménu fórum (obsahuje data o aktivitě studenta na fórech a o jeho odeslaných zprávách)

Dále bylo nad každým datasetem spuštěno 19 klasifikačních algoritmů. Klasifikátory byly vybrány tak, aby byl jejich výstup snadno srozumitelný i učitelům, kteří nemají odborné znalosti v oblasti vytěžování dat. Mezi algoritmy byly tedy vybrány tzv. rule-based klasifikátory (ConjunctiveRule, DecisionTable, DTN...) a rozhodovací stromy (BFTree, DecisionStump, J48...). Klasifikátory byly použity s výchozím nastavením parametrů a testovány pomocí 10-násobné křížové validace. Přesnost jednotlivých klasifikátorů byla měřena pomocí pěti měr:

1. Citlivost
2. Přesnost
3. F-Measure
4. Kappa
5. Plocha pod křivkou (AUC)

Pro vytvoření množiny klasifikátorů, které nad konkrétním datasetem vykazují nejvyšší přesnost, byl použit následující postup:

1. Nad datasetem byl spušten neparametrický statistický test Iman&Davenport (Friedmanův test s přesnějším odhadem kritické hodnoty [19]), jehož výstupem byly seřazené algoritmy dle hodnocení (průměrná úspěšnost z 5 měr).

2. Spočtení kritické hodnoty pomocí The Bonferroni-Dunn post-hoc testu a vytvoření intervalu [hodnocení nejlepšího klasifikátoru; hodnocení nejlepšího klasifikátoru + kritická hodnota] pro konkrétní dataset.
3. Pokud ohodnocení klasifikátoru spadá do intervalu z bodu 2, je přidán do množiny doporučených algoritmů pro daný dataset.

Predikční část meta-learningového frameworku pro neznámý dataset funguje tak, že se napočítají jeho meta-příznaky, podle nich se dohledá nejpodobnější dataset z meta-dat a doporučí se jeho množina nejlepších klasifikátorů. Pro dohledání datasetu je použit algoritmus KNN, s K=1.

Dále byly provedeny experimenty, které meta-příznaky jsou nejvhodnější pro nalezení nejpodobnějšího datasetu. Příznaky byly rozděleny do 4 skupin a přesnost doporučení při použití konkrétní skupiny byla testována na neznámém datasetu. Pro každou skupinu příznaků byl dohledán jiný nejpodobnější dataset z meta-dat. Pro vyhodnocení kvality doporučení byla pro neznámý dataset ještě spočtena množina klasifikátorů podle stejného postupu jako v trénovací části meta-learningu. Výsledek experimentu byl změřen pomocí míry F-Measure.

Z experimentu vyšlo, že nejlepších výsledků při hledání nejpodobnějšího datasetu dosáhlo použití všech napočítaných meta-příznaků (medián hodnoty F-Measure byl 0.65).

## 2.4 MUDOF

MUDOF (Meta-learning Using Document Feature characteristics) [20] je nástroj pro automatické přiřazování kategorií k textovým dokumentům. Specifikem tohoto nástroje je, že při klasifikaci není použit pouze jeden klasifikátor pro všechny kategorie. Nástroj využívá meta-learningový model pro zvolení vhodného klasifikačního algoritmu, který rozhodne, zda konkrétní dokument patří do dané kategorie.

Většina existujících meta-modelů pro kategorizaci textů je založena na lineární kombinaci několika základních klasifikačních algoritmů, což ovšem omezuje využití některých informací v trénovacích datasetech. Toto omezení řeší MUDOF. Vztah mezi kategorickými příznaky z dokumentů a predikovanou klasifikační chybou algoritmu je vyjádřen pomocí vícerozměrné regresní analýzy.

Cílem automatické kategorizace dokumentů je vyhodnocení klasifikačního schématu pro každou kategorii na základě trénovacích dat (dokumenty s již přidělenými kategoriemi). Během učení meta-modelu je na každém dokumentu trénován klasifikátor pro každou kategorii a následně je vytvořena regresní funkce, jejíž výstupem je chybovost klasifikátoru. Klasifikátory jsou poté doporučovány pro zpracování neznámých dokumentů tak, že pro každou kategorii je použit klasifikační algoritmus s nejmenší chybovostí. Z výsledku klasifikátoru

nad neznámým dokumentem je poté spočtena příslušnost do dané kategorie tak, že pokud je hodnota vyšší než aktuální práh, je kategorie k dokumentu přiřazena.

Jádrem nástroje MUDOF je naučení meta-modelu pro doporučení nejlepšího klasifikačního algoritmu pro konkrétní kategorii. Predikce je založená na regresním modelu, kde napočítané kategorizační příznaky nad datasetem dokumentů reprezentují nezávislé proměnné a klasifikační chyba je závislou proměnnou.

Při učení regresního modelu jsou použity 2 datasety s trénovacími dokumenty:

1. Trénovací set, který slouží pro nastavení optimálních hodnot koeficientů v regresní funkci.
2. Ladící set (tuning set), jehož příznaky jsou použity v prediktivní části algoritmu pro odhad klasifikační chyby a doporučení nejlepšího klasifikátoru.

MUDOF algoritmus je trénován na množině vybraných klasifikačních algoritmů (s optimalizovanými parametry pro jednotlivé kategorie). Každý algoritmus je spuštěn na trénovacím datasetu pro vytvoření klasifikačního modelu. Poté je klasifikační model testován na ladícím setu pro změření klasifikační chyby. Po získání klasifikačních chyb pro jednotlivé kategorie jsou iterativně optimalizovány parametry $\beta$ v regresní funkci a pro každý algoritmus je získán jeden regresní model.

Predikční část algoritmu funguje tak, že pro každý klasifikační algoritmus je na základě regresní funkce (s optimalizovanými koeficienty $\beta$ z meta-modelu a proměnnými pro kategorizační příznaky z ladícího datasetu) odhadnuta jeho chybovost pro konkrétní kategorii. Poté je doporučen algoritmus s nejmenší chybovostí.

Použité příznaky:

1. PosTr — počet dokumentů spadajících do dané kategorie v trénovacím setu
2. PosTu — počet dokumentů spadajících do dané kategorie v ladícím setu
3. AvgDocLen — průměrný počet indexovaných výrazů v dokumentech pro danou kategorii
4. AvgTermVal — průměrná váha výrazů v dokumentech pro danou kategorii
5. AvgMaxTermVal — maximální váha výrazů v dokumentech pro danou kategorii
6. AvgMinTermVal — minimální váha výrazů v dokumentech pro danou kategorii
7. AvgTermThre — průměrný počet výrazů s váhou nad daným prahem

8. AvgTopInfoGain — průměrný informační zisk top m výrazů (m=15)
9. NumInfoGainThres — počet výrazů mající informační zisk nad daným prahem

Během testování byla použita kolekce dat Reuters, odkud bylo využito téměř 10 tisíc dokumentů. Každý dokument je přiřazen k několika kategoriím, kterých je v kolekci celkově 90. 6 tisíc dokumentů bylo použito pro trénovací set, téměř 4 tisíce pro ladící set a 3 tisíce pro testování.

Pro naučení meta-modelu bylo použito 6 klasifikačních algoritmů:

1. Rocchio
2. WH (Widrow-Hoff)
3. KNN
4. SVM
5. GISR (Generalized instance set algorithm with Rocchio generalization)
6. GISW (Generalized instance set algorithm with WH generalization)

Použité měřící míry:

1. MBE (Micro-averaged recall and precision Break-Even point measure) — pro jednotlivé kategorie jsou spočítány dílčí hodnoty pro výpočet přesnosti a citlivosti (pravá pozitivní, falešně pozitivní...). Z těchto dílčích hodnot je následně spočtena výsledná přesnost a citlivost dohromady ze všech kategorií a pomocí interpolace je nalezen bod mezi těmito 2 mírami.
2. ABE (mAcro-averaged recall and precision Break-Even point measure) — pro jednotlivé kategorie jsou spočítány míry citlivost a přesnost, pomocí interpolace je nalezen bod mezi těmito 2 mírami, a výsledná hodnota je průměr z těchto bodů od všech kategorií.

Při testování byla změřena klasifikační přesnost jednotlivých klasifikačních algoritmů (kdy pouze jeden klasifikátor je použit pro všechny kategorie) oproti algoritmu MUDOF. Z naměřených výsledků vyplývá, že MUDOF ve všech případech vykázal lepší přesnost (zlepšení o 1-14% v případě míry ABE).

Při testování byly také zjištěny nejlepší klasifikační algoritmy pro jednotlivé kategorie, aby se spočítala úspěšnost odhadu algoritmu MUDOF. Z 90 kategorií dokázal MUDOF doporučit 59x správně nejlepší klasifikátor.

Kapitola 3

# Návrh meta-learningového modelu

Meta-learningový model má sloužit pro predikci užitečnosti jednotlivých příznaků napočítaných v procesu propositionalizace. Na základě predikované užitečnosti se určí pořadí příznaků k nápočtu, aby relevantní prediktory byly upřednostněny před méně přínosnými příznaky.

Meta-learningový model je učen a testován na datech ze serveru Relational Dataset Repository [22]. K převedení relačních dat do podoby jedné tabulky pomocí propositionalizace je použit nástroj Predictor Factory. Tento nástroj obsahuje seznam šablon s různými typy transformací, které lze na relační data aplikovat.

Úspěšně napočtené a neduplicitní prediktory, které jsou určeny pro další zpracování klasifikačními nebo regresními algoritmy, jsou vložené do tzv. *mainsample* tabulky. Informace o průběhu nápočtu jednotlivých prediktorů se ukládají do tzv. *journal* tabulky. Na základě *mainsample* a *journal* tabulky jsou částečně vytvořeny meta-data pro učení predikčního meta-modelu.

Jako faktory ovlivňující užitečnost příznaků byly zvoleny:

1. Míra Chi2 (kterou již používá nástroj Predictor Factory) — vyjadřuje závislost mezi příznaky a cílovým atributem.
2. Délka běhu — kvůli upřednostnění rychleji napočtených příznaků v případě podobné hodnoty Chi2.
3. Duplicita příznaků — je žádoucí rozpoznat a upozadit duplicitní příznaky v pořadí, aby mohl být využit výpočetní čas na ostatní neduplicitní příznaky.

Z těchto tří veličin je sestavena tzv. *fitness* funkce, jenž určuje pořadí příznaků pro nápočet.

$$fitness = f(\frac{Chi2}{doba\_nápočtu}) \tag{3.1}$$

Funkce $f$ snižuje hodnotu fitness podle odhadu pravděpodobnosti, že je příznak duplicitní. Čím vyšší je pravděpodobnost duplicity, tím více se sníží hodnota $fitness$ pro daný příznak. To znamená, že 100% duplicitní příznak má nulovou hodnotu fitness. Naopak jistě neduplicitnímu příznaku není hodnota fitness snížena vůbec.

Meta-learningový model se skládá z predikčních modelů pro každý z faktorů užitečnosti příznaků, na základě kterých jsou seřazeny příznaky k nápočtu.

## 3.1 Použité technologie

Pro vývoj meta-learningového modelu byl zvolen nástroj Jupyter Notebook (Ipython Notebook). Jedná se o grafické interaktivní rozhraní k jazyku Python (a některých dalších). Toto rozhraní je založené na achitektuře klient-server. Jako klient je použit webový prohlížeč a server je součástí nástroje IPython.

Nástroj Ipython Notebook umožňuje vývoj v tzv. diářích (notebooks), jejichž struktura se skládá z jednotlivých buněk. Do těchto buněk je možné psát kód, deklaraci metod či tříd a každou buňku je možné ihned vyhodnotit. Výhodou tohoto nástroje je, že umožňuje rychlé testování napsaného kódu a podporuje knihovny specializované na strojové učení.

Základní knihovny pro práci s daty v jazyku Python:

1. scikit-learn — obsahuje implementace algoritmů pro strojové učení.
2. Scipy — poskytuje funkce pro matematické a vědecké výpočty.
3. Numpy — obsahuje různé objektové struktury pro práci s daty.
4. Matplotlib — knihovna nabízející široký výběr pro vykreslení grafů a statistik.

## 3.2 Meta-příznaky

Sekce obsahuje výčet napočítaných meta-příznaků, na základě kterých byly učeny a laděny predikční algoritmy.

### 3.2.1 Míra Chi2

Míra Chi2 je jedním z faktorů predikující užitečnost příznaků pro následnou klasifikaci či regresi. Napočítat ji lze nad jednotlivými sloupci tabulky *mainsample* získané z procesu propositionalizace.

Jelikož je tato míra určena pro kategorické proměnné, je třeba převést spojité atributy na diskrétní hodnoty. Pro diskretizaci každého příznaku (včetně cílového sloupce v případě datasetu určeného pro regresní problém) bylo použito 11 kategorií:

1. Kategorie -1 — reprezentuje chybějící hodnoty.
2. Kategorie 1 až 10 — u každého spojitého atributu je podle jeho rozsahu hodnot vytvořeno 10 stejně širokých intervalů, do kterých jsou následně jednotlivé hodnoty sloupce zařazeny.

Po diskretizaci dat je pro každý příznak vytvořena kontingenční tabulka s cílovým atributem a z této tabulky je spočtena hodnota Chi2.

Během nápočtu míry Chi2 se současně zapisuje meta-atribut třída Chi2 indikující, zda je hodnota Chi2 nulová či nikoliv. Tento meta-atribut má sloužit pro rozšíření predikčního meta-modelu o binární klasifikátor nulových hodnot Chi2.

Při nápočtu příznaků během propositionalizace vznikají i duplicitní sloupce sloupců, které jsou v tabulce *mainsample* již napočtené. Duplicity se do cílové tabulky *mainsample* nevkládají, informace o jejich nápočtu se pouze uloží do tabulky *journal*. Užitečnost těchto sloupců je stejná jako u sloupce z tabulky *mainsample*, pouze byly napočtené později. Je tedy žádoucí tyto prediktory přidat do učícího datasetu.

Při nápočtu hodnoty Chi2 pro konkrétní prediktor se proto ještě dohledají všechny jeho duplicity v tabulce *journal*. Tyto duplicitní příznaky jsou přidány do učícího datasetu, jsou označeny jako duplicity a je jim přiřazena napočtená míra Chi2 od daného prediktoru.

### 3.2.2 Nápočet meta-příznaků

#### 3.2.2.1 Jednoduché meta-příznaky

Jedna část meta-příznaků je získána pomocí SQL dotazu do několika databázových zdrojů, které mohou obsahovat důležité informace pro následnou predikci užitečnosti příznaků:

1. Tabulka *journal* — tato tabulka je jedním z výstupů procesu propositionalizace v nástroji Predictor Factory. Obsahuje informace o jednotlivých příznacích a průběhu jejich nápočtu. Z tabulky jsou použity informace, které by byly k dispozici ještě před nápočtem. Dále jsou z této tabulky získána data pro nápočet landmarkovacích příznaků.

| **Meta-příznak** | **Popis** |
|---|---|
| predictor_name | Název příznaku. |
| table_name | Název tabulky, ze které má být příznak napočítán. |
| column_list | Seznam sloupců, ze kterých má být příznak napočítán. |
| parameter_list | Seznam parametrů pro nápočet příznaku. |

| | |
|---|---|
| run_time | Délka běhu nápočtu. Použito pouze pro landmarkovací příznaky. |
| is_duplicate | Identifikátor, zda je příznak duplicitní s jiným příznakem. Použito pouze pro landmarkovací příznaky. |
| duplicate_name | Název prediktoru, se kterým je příznak duplicitní. Použito pouze pro landmarkovací příznaky. |

Tabulka 3.1: Meta-příznaky z tabulky *journal*

2. MySQL tabulka *information_schema.tables*, která obsahuje meta-data o tabulkách v databázi.

| **Meta-příznak** |
|---|
| table_catalog |
| table_schema |
| table_name |
| table_type |
| engine |
| version |
| row_format |
| table_rows |
| avg_row_length |
| data_length |
| max_data_length |

| **Meta-příznak** |
|---|
| index_length |
| data_free |
| auto_increment |
| create_time |
| update_time |
| check_time |
| table_collation |
| checksum |
| create_options |
| table_comment |

Tabulka 3.2: Meta-příznaky z tabulky *information_schema.tables*

3. MySQL tabulka *information_schema.columns*, která obsahuje meta-data o jednotlivých sloupcích v konkrétní tabulce.

| Meta-příznak |
|---|
| item table_catalog |
| table_schema |
| table_name |
| column_name |
| ordinal_position |
| column_default |
| is_nullable |
| data_type |
| character_maximum_length |
| character_octet_length |
| numeric_precision |

| Meta-příznak |
|---|
| numeric_scale |
| datetime_precision |
| character_set_name |
| collation_name |
| column_type |
| column_key |
| extra |
| privileges |
| column_comment |
| generation_expression |

Tabulka 3.3: Meta-příznaky z tabulky *information_schema.columns*

4. MariaDB tabulka *mysql.column_stats*, která obsahuje statistické informace o konkrétních sloupcích.

| Meta-příznak | Popis |
|---|---|
| db_name | |
| table_name | |
| column_name | |
| min_value | Minimální hodnota ve sloupci. |
| max_value | Maximální hodnota ve sloupci. |
| nulls_ratio | Poměr NULL hodnot. |
| avg_length | Průměrná délka hodnoty v daném sloupci. |
| avg_frequency | Průměrný počet záznamů se stejnou hodnotou. |
| hist_size | |
| hist_type | |
| histogram | |

Tabulka 3.4: Meta-příznaky z tabulky *mysql.column_stats*

Jelikož některé zdrojové tabulky obsahují částečně stejné informace, jsou po nápočtu meta-příznaků promazány duplicitní sloupce.

#### 3.2.2.2 Landmarkovací meta-příznaky

Druhá část meta-příznaků je dopočítána na základě landmarkovacích dat. Landmarkovací meta-příznaky se napočítávají pro predikci Chi2, třídy Chi2, délky běhu a duplicity. V meta-příznacích jsou označeny předponou *landmark*.

Příznaky, ze kterých se dopočítávají landmarkovací data, jsou vybírány podle použité šablony pro nápočet v procesu propositionalizace. Každý typ prediktoru je napočítán podle jedné konkrétní šablony. Podmínky pro výběr šablon k nápočtu landmarkovacích dat:

1. Jsou výpočetně nenáročné.
2. Zlepšují prediktivní přesnost modelu.
3. Pokrytí nejčastějších datových typů (numerické, char, datum).

Pro nápočet landmarkovacích příznaků bylo vybráno 7 šablon z 31 (Tabulka 4.2):

1. Direct field
2. Aggregate
3. Time Since
4. WOE
5. Count
6. Aggregate WOE
7. Time aggregate

Šablony byly vybrané tak, aby téměř pro každý sloupec ve vstupním datasetu byl napočítaný některý landmarkovací příznak. Tato skutečnost umožňuje pro zbylé prediktory odhadnout hodnoty landmarkovacích meta-příznaků jako průměr z již napočítaných landmarkovacích dat na stejném sloupci. Pokud pro sloupec žádný landmarkovací příznak neexistuje, spočítá se odhad jako průměr landmarkovacích hodnot v odpovídající tabulce. Tabulka 3.5 obsahuje seznam meta-příznaků napočtených na základě landmarkovacích dat.

| **Meta-příznak** | **Popis** |
|---|---|
| landmark_chi2 | Odhadovaná hodnota Chi2 na základě landmarkovacích dat. |
| landmark_chi2_class | Odhadovaná pravděpodobnost nenulové hodnoty Chi2 na základě landmarkovacích dat. |
| landmark_run_time | Odhadovaná hodnota délky běhu na základě landmarkovacích dat. |
| avg_duplication_for_the_used_columns | Landmarkovací příznak spočten nad použitými sloupci v příznaku. Udává průměrná duplicitu příznaků napočtených na daných sloupcích. |

Tabulka 3.5: Landmarkovací meta-příznaky

### 3.2.3 Předzpracování meta-příznaků

Na napočtená meta-data byly aplikovány dvě metody předzpracování dat:

1. Transformace nečíselných atributů na numerické hodnoty:
   a) Datumové sloupce jsou převedeny na počet sekund od 1.1.1970 00:00.
   b) Na atributy typu char je aplikována metoda one-hot-encoding, která transformuje kategorické proměnné ve sloupci na binární proměnné. Pro každou položku z kategorie je vytvořen nový sloupec s hodnotami 0 a 1, které reprezentují výskyt proměnné. Atributy typu char, které mají příliš mnoho unikátních hodnot, jsou z učícího datasetu vynechány.
2. Nahrazení chybějících hodnot ve spojitých atributech průměrem z daného sloupce. Pro každý atribut s chybějícími hodnotami je do datasetu přidán binární sloupec identifikující, zda v dané buňce hodnota chybí (hodnota 1) či nikoliv (hodnota 0). Tento binární sloupec je pojmenován názvem sloupce, který identifikuje, a příponou *bad*.

## 3.3 Predikce hodnot Chi2

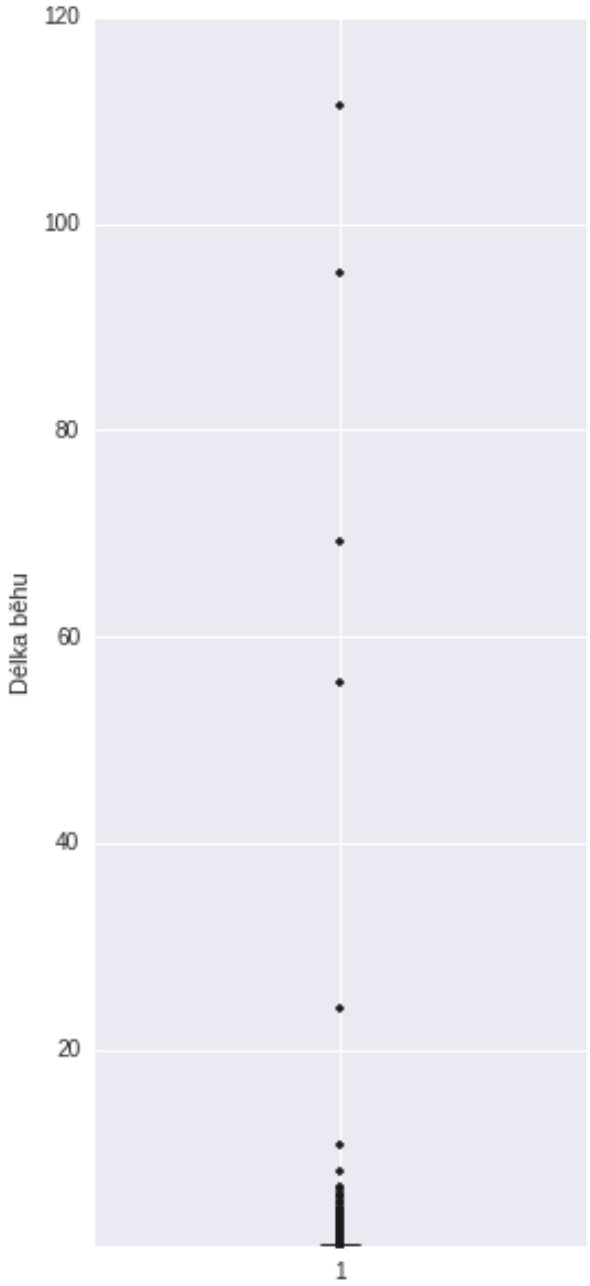
Predikce pro odhad míry Chi2 je navržena jako kombinace regresního a klasifikačního modelu. Regresní model slouží pro odhad konkrétní hodnoty Chi2 a je naučen pouze nad trénovacími daty, které mají nenulovou hodnotu Chi2. Pro odhad příznaků s nulovou hodnotou Chi2 je vytvořen binární klasifikační model, který by měl predikovat, zda příznak má nulovou či nenulovou hodnotu Chi2.

Napočtené hodnoty Chi2 se pohybují v řádu jednotek, ale i stovek a ve vysoce relevantních příznacích i přes hodnotu tisíc. Proto je regresní model trénován na zlogaritmovaných a normalizovaných hodnotách Chi2. Pro normalizaci je použita metoda MinMaxScaler, která přeškáluje hodnoty do intervalu [0; 1].

Pro regresní model jsou testovány 2 algoritmy s odlišným přístupem predikce — ElasticNet (definice 4) a XGBoost (definice 5). Pro klasifikační model je použit algoritmus pro binární klasifikaci logistická regrese (definice 2). Optimalizace a testování predikčních modelů je popsáno v sekci 4.

## 3.4 Predikce délky běhu

Některé datasety obsahovaly odchylky v cílovém sloupci délky běhu. Rozložení hodnot tohoto atributu je znázorněno na grafu typu boxplot 3.1. Aby němely tyto vychýlené hodnoty negativní dopad na učení modelu, byly z datasetu odstraněny řádky s hodnotou délky běhu vyšší než 20s.




Obrázek 3.1: Grafické vykreslení odchylek délky běhu v datech

Regresní model predikce délky běhu používá algoritmus XGBoost. Optimalizace a testování algoritmu je popsáno v sekci 4.

## 3.5 Predikce duplicitních prediktorů

Při nápočtu příznaků v průběhu propositionalizace vznikají duplicitní sloupce. Meta-learningový nástroj by měl zohlednit, aby se přednostně napočítávaly pravděpodobně neduplicitní sloupce. Proto je rozšířen o predikci pravděpodobnosti duplicitních příznaků pomocí binárního klasifikátoru používajícího logistickou regresi. Meta-příznaky použité pro binární klasifikaci:

1. dupl_prob (definice 3.2)
2. pattern_name
3. avg_duplication_for_the_used_columns (definice v tabulce 3.5)

Během nápočtu příznaků jsou průběžně získávány meta-data:

1. *col_used* — počet již spočtených prediktorů, které používají stejný sloupec
2. *col_dupl* — počet již spočtených prediktorů, které používají stejný sloupec a jsou duplicitní

Z těchto dvou čísel se ještě nenapočteným prediktorům spočte meta-příznak *dupl_prob*, který odhaduje pravděpodobnost duplicity jako:

$$dupl\_prob = \frac{col\_dupl}{col\_used} \tag{3.2}$$

Hodnoty *col_dupl* jsou inicializovány na 0 a hodnoty *col_used* na 1, aby se předešlo problému dělení nulou. Inicializace *col_used* na 1 způsobí pouze to, že ze začátku bude pravděpodobnost duplicit nepatrně nižší, ovšem s vyšším počtem napočtených prediktorů to bude bezvýznamný rozdíl.

Sloupců může být v prediktoru použito více, sloupec je tedy identifikován názvem tabulky a sjednocením názvů použitých sloupců (sloupce jsou řazeny vždy abecedně).

Jako první se při nápočtu prediktorů zpracují landmarkovací příznaky (Direct Field, Aggregate, Time Since, WOE, Count, Aggregate WOE, Time aggregate), u kterých se duplicita nepredikuje, a na základě jejich meta-dat se natrénuje binární klasifikátor.

### 3.5.1 Algoritmus pro predikci duplicity

Pro posílání prediktorů k nápočtu se používá vzorkování dat (data sampling) — ve while cyklu se vždy vybere $N$ příznaků s nejvyšší fitness hodnotou. Hodnota $N$ je na začátku malá, a poté se postupně zvětšuje.

1. While cyklus, dokud není množina prediktorů k nápočtu prázdná.
2. Dataset s příznaky k nápočtu se seřadí sestupně podle metriky $\frac{Chi2}{délka\ běhu}$.

3. Vybere se prvních $N$ příznaků.
4. Pokud vybrané příznaky již byly penalizovány, pošlou se k nápočtu.
5. U zbylých vybraných příznaků se spočte meta-příznak *dupl_prob*.
6. Predikuje se pravděpodobnost duplicity příznaků *dupl_prediction* pomocí binárního klasifikátoru.
7. Vygeneruje se náhodné číslo *n_random* od 0 do 1.
8. Prediktory, jejichž hodnota *dupl_prediction* je menší než *n_random*, se pošlou k nápočtu. To znamená, že pokud má prediktor například 70% pravděpodobnost, že je duplicitní, pošle se k nápočtu s 30% pravděpodobností. Naopak pokud má pouze 20% pravděpodobnost duplicity, napočte se s 80% pravděpodobností.
9. Příznakům, které se k nápočtu nepošlou, se sníží fitness hodnota (definice 3.1) podle jejich pravděpodobnosti duplicity. Příznaky se vrátí do množiny ještě nenapočtených prediktorů a označí se jako penalizované, aby se příště již rovnou poslaly k nápočtu.

Tímto procesem se docílí toho, že duplicitní příznaky se upozadí, ovšem stále mají šanci, že budou napočítány (což je žádoucí, protože odhad, že příznak je duplicitní, může být chybný a může se jednat o vysoce relevantní příznak).

Kapitola 4

# Experimentální část

Tvorbě predikčních modelů předcházela příprava trénovacích dat. Použité datasety a typy transformací pro převod relačních dat do podoby jedné tabulky jsou vypsány v sekci 4.1 a 4.2.

Sekce 4.3 a 4.4 se věnují selekci napočítaných meta-příznaků, ladění a optimalizování jednotlivých predikčních meta-learningových modelů, i jejich kombinacím.

## 4.1 Použitá data

Pro účely vývoje a testování byl jako zdroj dat použit repozitář The CTU Prague Relational Learning Repository [23]. Tento repozitář obsahuje relační data pro vývoj strojového učení a momentálně obsahuje kolem sedmdesáti datasetů. Databáze jsou hostovány na MySQL serveru relational.fit.cvut.cz.

Pro vývoj byly použité datasety s reálnými daty. Celý výčet použitých datasetů se nachází v tabulce 4.1. Sloupec s počtem řádků představuje celkový počet řádků v datasetu, sloupec s počtem instancí repzrezentuje počet řádku v cílové tabulce datasetu. Pro poslední sloupec tabulky je využito značení K — klasifikace a R — regrese.

| **Název datasetu** | **#tabulek** | **#atributů** | **#řádků** | **#instancí** | **Úkol** |
|---|---|---|---|---|---|
| Financial | 8 | 55 | 1086274 | 682 | K |
| Accidents | 3 | 43 | 1433638 | 483558 | K |
| CCS | 6 | 28 | 5347501 | 1000 | R |
| AustralianFootball | 4 | 77 | 138057 | 3036 | K |
| BasketballMen | 9 | 195 | 143787 | 1536 | R |
| Biodegradability | 5 | 17 | 22054 | 328 | R |
| Carcinogenesis | 6 | 23 | 27570 | 329 | K |
| Chess | 2 | 45 | 2052 | 295 | K |
| CORA | 3 | 6 | 57884 | 2708 | K |

| | | | | | |
|---|---|---|---|---|---|
| Hepatitis | 7 | 26 | 12927 | 500 | K |
| Mondial | 40 | 167 | 21497 | 204 | K |
| Nations | 3 | 118 | 11004 | 14 | K |
| PremiereLeague | 4 | 217 | 10716 | 380 | K |
| AustralianFootball | 38 | 76 | 29762 | 299 | K |
| StudentLoan | 10 | 15 | 5288 | 1000 | K |
| VisualGenome | 6 | 20 | 2483807 | 1313454 | K |
| Walmart | 4 | 27 | 4628497 | 4607680 | R |
| WebKP | 3 | 6 | 80592 | 877 | K |
| World | 3 | 24 | 5411 | 239 | K |

Tabulka 4.1: Popis použitých datasetů z The CTU Prague Relational Learning Repository. Pro poslední sloupec tabulky je využito značení K — klasifikace a R — regrese.

## 4.2 Použité šablony v PF

K převodu relačních dat do jedné tabulky je použit nástroj Predictor Factory. Příznaky jsou nad relačními daty napočítány různými typy transformací, které jsou definovány pomocí šablon. Seznam použitých šablon obsahuje tabulka 4.2.

| **Název šablony** | **Popis** | **Typ proměnných** | **Kardinalita** |
|---|---|---|---|
| Aggregate | Aplikace agregačních funkcí (stddev_samp, avg, min, max, sum). | Numerické | N |
| Aggregate distinct | Agregační funkce (stddev_samp, avg, sum) aplikovaná nad unikátními hodnotami. | Numerické | N |
| Aggregate frame | Agregační funkce (stddev_samp, avg, min, max, sum) aplikovaná nad časově omezenými hodnotami z numerických sloupců. | Numerické | N |
| Aggregate text length | Agregační funkce (stddev_samp, avg, min, max, sum) aplikovaná nad počtem znaků v proměnných typu char. | Nominální | N |
| Aggregate range | Rozdíl maximální a minimální hodnoty. | Numerické | N |
| Direct field | Vrací všechny sloupce z tabulky. | Všechny | 1 |

| | | | |
|---|---|---|---|
| Count | Počet záznamů v tabulce po aplikaci grupovací funkce. | | N |
| Coefficient of variation | Poměr směrodatné odchylky ku průměru hodnot. | Numerické | N |
| Correlation | Pearsonův korelační koeficient nad numerickými sloupci s datumovým sloupcem. | Numerické | N |
| Distinct count | Počet unikátních hodnot. | Nomilnální | N |
| Duplicate ratio | Poměr celkového počtu záznamů ku počtu duplicit. | Nomilnální, numerické | N |
| Intercept | Čas přechodu z negativní hodnoty na pozitivní (a naopak). | Numerické | N |
| Log product | Logaritmus z produktu hodnot numerických sloupců. Nefunguje pro negativní hodnoty. | Numerické | N |
| Existential count | Počet výskytů konkrétní hodnoty. Ignorují se hodnoty NULL, které jsou řešeny šablonou Null ratio. Momentálně se tato hodnota napočítává pro 20 nejvíce frekventovaných hodnot. | Nomilnální | N |
| Null ratio | Poměr NULL hodnot ve sloupci. | Všechny | N |
| Slope | Směrnice přímky z lineární regrese. | Numerické | N |
| Text length | Počet znaků v proměnné typu char. | Nominální | 1 |
| Time day part | Určuje část dne (ráno, poledne, odpoledne, večer, noc) z datumové sloupce, pokud obsahuje i časovou složku. | Datum | 1 |
| Time diff | Rozdíl mezi 2 datumovými sloupci. | Datum | 1 |
| Time range | Rozdíl mezi maximální a minimální hodnotou. | Datum | N |
| Time frequency | Poměr počtu záznamů ve sloupci ku době trvání (tj. hodnota Time range). Pro vyvárování se dělení nulou je k délce trvání vždy připočtena hodnota 1. | Všechny | N |

| | | | |
|---|---|---|---|
| Time is weekend | Klasifikuje hodnotu buď do pracovních dnů nebo na víkend (výsledek je závislý na nastavení začátku týdne v databázi). | Datum | 1 |
| Time part | Umožňuje specifikovat hodinu/den/den v týdnu/měsíc z datumového sloupce. | Datum | 1 |
| Time since | Vyjadřuje čas od nějakého data v minulosti do data predikce. | Datum | 1 |
| Time aggregate | Agregační funkce (stddev_samp, min, max, avg, sum) aplikovaná na datumových atributech. | Datum | N |
| Time aggregate diff | Agregační funkce (stddev_samp, min, max, avg, sum) aplikovaná nad šablonou Time diff. | Datum | N |
| Time aggregate since | Agregační funkce (stddev_samp, min, max, avg) aplikovaná nad šablonou Time since. | Datum | N |
| Time aggregate since event | Agregační funkce (stddev_samp, min, max, avg) pracující s časem od určité události. | Datum | N |
| WOE (Weight of evidence) | Metrika pro vyjádření prediktivní síly nezávislé proměnné vůči závislé proměnné. Příznak funguje pouze pro binární cílový atribut. | Nominální, ID | 1 |
| Aggregate WOE | Agregační funkce aplikovaná nad WOE. | Nominální, ID | N |
| Time WOE | WOE pro datumové sloupce. | Datum | N |

Tabulka 4.2: Šablony pro nápočet příznaků používané v Predictor Factory

## 4.3 Meta příznaky

Zvolené meta-příznaky pro popis napočtených prediktorů z procesu propositionalizace jsou popsané v sekci 3.2. Během optimalizace predikčních modelů byla pro každý model analyzována podmnožina nejlepších meta-příznaků.

### 4.3.1 Predikce Chi2

Pro predikci hodnot míry Chi2 byly testovány 2 algoritmy — ElasticNet a XGBoost.

Tabulka 4.3 obsahuje výpis koeficientů jednotlivých příznaků, které byly naměřeny pomocí regresního algoritmu ElasticNet při predikci hodnot Chi2 (implementace algoritmu ElasticNet z knihovny scikit-learn neposkytuje k jednotlivým koeficientům p-hodnoty). Podle váhy příznaků se dá orientačně určit jejich důležitost při predikci.

Data v tabulce jsou seřazena sestupně podle absolutních hodnot koeficientů. Příznaky, jejichž váha byla nulová a tudíž nemají na výsledek predikce žádný vliv, nejsou v tabulce uvedeny. Nominální příznaky jsou v tabulce uvedeny po transformaci pomocí algoritmu One Hot Encoding (tzv. dummy kódování).

Jelikož samotný algoritmus ElasticNet zajišťuje penalizaci příznaků, byly při testování kvality predikce Chi2 použity všechny meta-příznaky. Tabulka 4.3 pouze reflektuje váhu příznaků při použití algoritmu ElasticNet.

| **Příznak** | **Váha příznaku** |
|---|---|
| landmark_chi2 | 0.52 |
| ohe_create_options=avg_row_length=50 | -0.06 |
| ohe_pattern_name=time aggregate diff | -0.04 |
| ohe_pattern_name=aggregate woe | 0.03 |
| ohe_pattern_name=existential count | -0.03 |
| ohe_pattern_name=WOE | 0.03 |
| ohe_collation_name=utf8_bin | -0.02 |
| ohe_table_collation=utf8_bin | -0.02 |
| landmark_is_duplicate=bad | -0.02 |
| ohe_column_type=time | 0.01 |
| ohe_data_type=time | 0.01 |
| ohe_column_key | 0.01 |
| ohe_pattern_name=null ratio | -0.01 |
| ohe_pattern_name=aggregate distinct | 0.01 |
| ohe_character_set_name=utf8 | -0.01 |
| ohe_column_key=uni | -0.01 |
| ohe_column_type=varchar | -0.01 |
| ohe_data_type=varchar | -0.01 |
| auto_increment=bad | -0.01 |

Tabulka 4.3: Váha příznaků pro predikci Chi2 z algoritmu ElasticNet

Během učení modelu používající algoritmus XGBoost je každému příznaku spočítána váha, která reprezentuje jeho přínos pro predikci. Výčet prediktorů

a jejich koeficientů je uveden v tabulce 4.4, data jsou seřazena sestupně podle váhy příznaků.

| **Příznak** | **Váha příznaku** |
|---|---|
| landmark_chi2 | 282 |
| landmark_run_time | 190 |
| avg_frequency | 102 |
| avg_length | 81 |
| table_rows | 74 |
| ohe_pattern_name=aggregate text length | 70 |
| avg_row_length | 68 |
| ohe_pattern_name=existential count | 63 |
| ordinal_position | 54 |
| avg_duplication_for_the_used_columns | 43 |
| landmark_is_duplicate=bad | 42 |
| nulls_ratio | 40 |
| ohe_pattern_name=distinct count | 39 |
| landmark_is_duplicate | 36 |
| character_octet=length | 33 |
| character_maximum_length | 29 |
| ohe_pattern_name=null ratio | 29 |
| ohe_pattern_name=duplicate ratio | 29 |
| data_length | 28 |
| ohe_pattern_name=time frequency | 27 |
| index_length | 27 |
| landmark_chi2_class | 25 |
| ohe_pattern_name=coefficient of variation | 23 |
| column_count | 23 |
| ohe_pattern_name=time aggregate since event | 20 |
| ohe_pattern_name=time woe | 20 |
| ohe_pattern_name=text length | 20 |
| table_count | 16 |
| ohe_pattern_name=log product | 15 |
| ohe_pattern_name=aggregate woe | 14 |
| ohe_pattern_name=aggregate distinct | 14 |
| ohe_pattern_name=time is weekend | 14 |
| ohe_create_options=avg_row_length=50 | 12 |
| auto_increment | 10 |
| ohe_pattern_name=time aggregate diff | 10 |
| ohe_pattern_name=time aggregate since | 9 |
| ohe_table_collation=utf8_bin | 7 |
| ohe_pattern_name=correlation | 7 |

| | |
|---|---|
| ohe_pattern_name=time part | 7 |
| ohe_pattern_name=time day part | 7 |
| ohe_pattern_name=intercept | 6 |
| ohe_pattern_name=time diff | 6 |
| ohe_pattern_name=aggregate | 5 |
| datetime_precision=bad | 5 |
| ohe_pattern_name=aggregate range | 5 |
| ohe_create_options=row_format=compact | 4 |
| ohe_is_nullable=no | 4 |
| character_maximum_length=bad | 4 |
| ohe_pattern_name=time aggregate | 4 |
| data_free | 4 |
| ohe_column_key=mul | 4 |
| ohe_column_default=0 | 3 |
| ohe_column_default=member | 3 |
| ohe_table_collation=utf8_general_ci | 3 |
| ohe_collation_name=utf8_bin | 3 |
| ohe_is_nullable=yes | 3 |
| numeric_precision | 3 |
| ohe_create_options | 3 |
| ohe_data_type=datetime | 2 |
| ohe_table_collation=latin1_swedish_ci | 2 |
| ohe_character_set_name=latin1 | 2 |
| ohe_data_type=date | 2 |
| character_octet=length=bad | 2 |
| ohe_column_type=datetime | 2 |
| ohe_character_set_name=utf8 | 2 |
| ohe_column_key=pri | 2 |
| numeric_precision=bad | 2 |
| ohe_column_key= | 1 |
| ohe_data_type=tinyint | 1 |
| ohe_column_default= | 1 |
| ohe_column_type=float | 1 |
| ohe_collation_name=utf8_general_ci | 1 |
| ohe_pattern_name=direct field | 1 |
| ohe_data_type=char | 1 |
| auto_increment=bad | 1 |
| ohe_collation_name=latin1_swedish_ci | 1 |
| ohe_column_key=uni | 1 |

Tabulka 4.4: Důležitost příznaků pro predikci Chi2 z algoritmu XGBoost

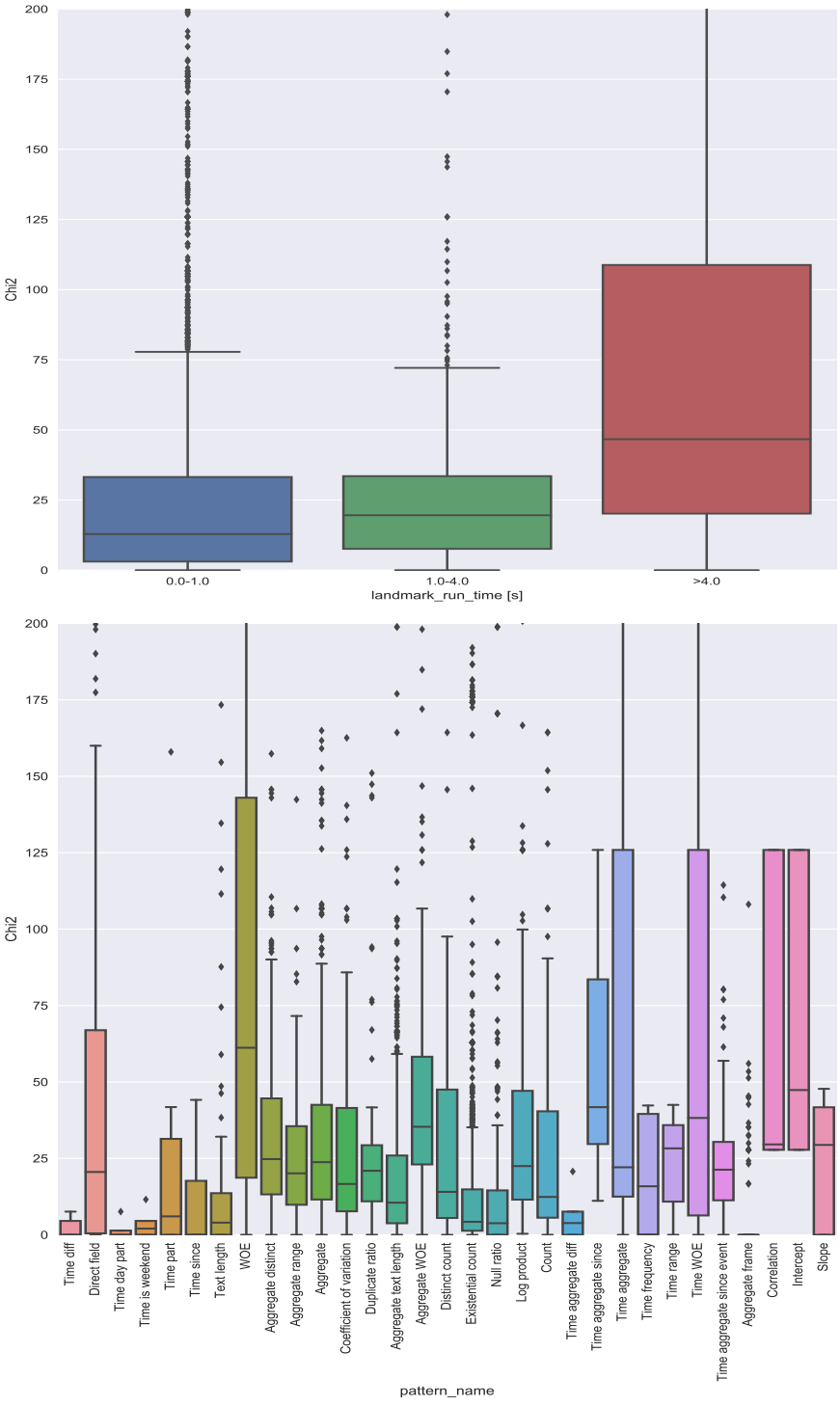
Obsah tabulky 4.4 slouží pouze pro orientační zjištění důležitosti meta-příznaků. Z této tabulky byla vybrána množina nejlepších meta-příznaků, jejíž vliv na kvalitu predikce byl dál testován. Zvolen byl iterativní postup, kdy výsledná množina obsahovala na začátku pouze 2 nejlepší meta-příznaky a po jednom do ní byly přidávány ostatní meta-příznaky. Přidaný příznak byl ponechán, pokud se díky němu zvýšila hodnota míry KI při testování na většině datasetech.

Pro výběr nejvhodnějších meta-příznaků nebyla použita metoda forward selection, protože knihovna scikit-learn implementaci této metody neposkytuje. Knihovna scikit-learn nabízí pro selekci příznaků metodu RFE (Recursive Feature Elimination). Ta ovšem nelze použít pro algoritmus XGBoost, protože nevrací váhu atributů z učení v parametru $\_coef$.

Ve výsledné množině nejlepších meta-příznaků pro predikci Chi2 byly ponechány následující příznaky:

1. landmark_chi2 — Odhadovaná hodnota Chi2 na základě landmarkovacích příznaků.
2. landmark_run_time — Příznaky, které se složitěji (déle) napočítávají mohou mít výšší hodnotu míry Chi2.
3. pattern_name — Důležitý meta-příznak, jelikož nápočet některých šablon může být v regeresi či klasifikaci užitečnější než ostatní šablony.
4. avg_length — Průměrná délka řádků ve sloupci.
5. table_rows — Počet řádků v dané tabulce.
6. avg_row_length — Průměrná délka řádků v tabulce.
7. avg_frequency — Tento meta-příznak udává průměrný počet záznamů se stejnou hodnotou.
8. data_type — Datový typ sloupce.
9. numeric_precision — Numerická přesnost.
10. index_length
11. landmark_chi2_class
12. table_count — Počet tabulek ve schématu.
13. auto_increment — Zda se jedná o sloupec s automaticky inkrementovanými hodnotami.
14. is_nullable — Zda může sloupec obsahovat hodnoty NULL.

Tato podmnožina meta-příznaků je dále používána při měření kvality algoritmu XGBoost pro predikci Chi2. Vztah mezi cílovým atributem Chi2 a meta-příznaky landmark_run_time a pattern_name je graficky zobrazen na obrázku 4.1.




Obrázek 4.1: Grafy zobrazující vztah mezi některými meta-příznaky a cílovým atributem Chi2. Hodnoty míry Chi2 na ose y jsou kvůli odlehlým hodnotám omezeny na interval [0;200].

### 4.3.2 Predikce třídy Chi2

Pro predikci třídy Chi2 je použit binární klasifikační algoritmus logistická regrese. Pro zjištění nejlepších meta-příznaků bylo možné použít metodu RFE (Recursive Feature Elimination) z knihovny scikit-learn. Mezi nejlepší meta-příznaky patří:

1. pattern_name
2. landmark_chi2_class
3. avg_length
4. numeric_precision
5. table_count

### 4.3.3 Predikce délky běhu

Tabulka 4.5 obsahuje naměřené váhy meta-příznaků pro predikci doby nápočtu jednotlivých příznaků pomocí algoritmu XGBoost. Data jsou seřazena sestupně podle váhy meta-příznaků.

| **Příznak** | **Váha příznaku** |
|---|---|
| landmark_run_time | 149 |
| character_octet_length | 44 |
| ohe_pattern_name=aggregate text length | 34 |
| landmark_is_duplicate=bad | 27 |
| ohe_pattern_name=null ratio | 19 |
| landmark_is_duplicate | 19 |
| landmark_chi2 | 15 |
| ohe_pattern_name=existential count | 15 |
| ohe_pattern_name=distinct count | 13 |
| ohe_pattern_name=time woe | 12 |
| table_rows | 10 |
| avg_row_length | 10 |
| ohe_pattern_name=duplicate ratio | 9 |
| ohe_pattern_name=time aggregate since event | 9 |
| ohe_pattern_name=aggregate frame | 8 |
| avg_frequency | 7 |
| ohe_pattern_name=aggregate distinct | 7 |
| character_maximum_length | 6 |
| ordinal_position | 6 |
| ohe_pattern_name=aggregate range | 6 |
| avg_duplication_for_the_used_columns | 6 |
| ohe_pattern_name=time frequency | 5 |

| | |
|---|---|
| ohe_data_type=enum | 5 |
| table_count | 5 |
| create_time | 4 |
| ohe_create_options=avg_row_length=50 | 4 |
| nulls_ratio | 4 |
| ohe_pattern_name=time aggregate since | 4 |
| ohe_pattern_name=intercept | 3 |
| ohe_pattern_name=text length | 3 |
| column_count | 3 |
| data_length | 3 |
| data_free | 3 |
| ohe_column_key_mul | 3 |
| ohe_table_collation=utf8_bin | 2 |
| ohe_data_type=datetime | 2 |
| index_length | 2 |
| ohe_column_comment=players from premier league | 2 |
| ohe_create_options=row_format=compact | 2 |
| ohe_pattern_name=aggregate | 2 |
| ohe_pattern_name=coefficient of variation | 2 |
| ohe_is_nullable=no | 2 |
| ohe_column_default=member | 2 |
| avg_length | 2 |
| ohe_table_collation=utf8_general_ci | 2 |
| ohe_data_type=date | 2 |
| ohe_column_default=f | 2 |
| ohe_data_type=varchar | 2 |
| ohe_pattern_name=log product | 1 |
| ohe_pattern_name=aggregate woe | 1 |
| ohe_pattern_name=correlation | 1 |
| ohe_pattern_name=time part | 1 |
| auto_increment | 1 |
| ohe_pattern_name=slope | 1 |
| landmark_chi2_class | 1 |
| ohe_column_type=enum('T','F') | 1 |
| character_octet_length=bad | 1 |

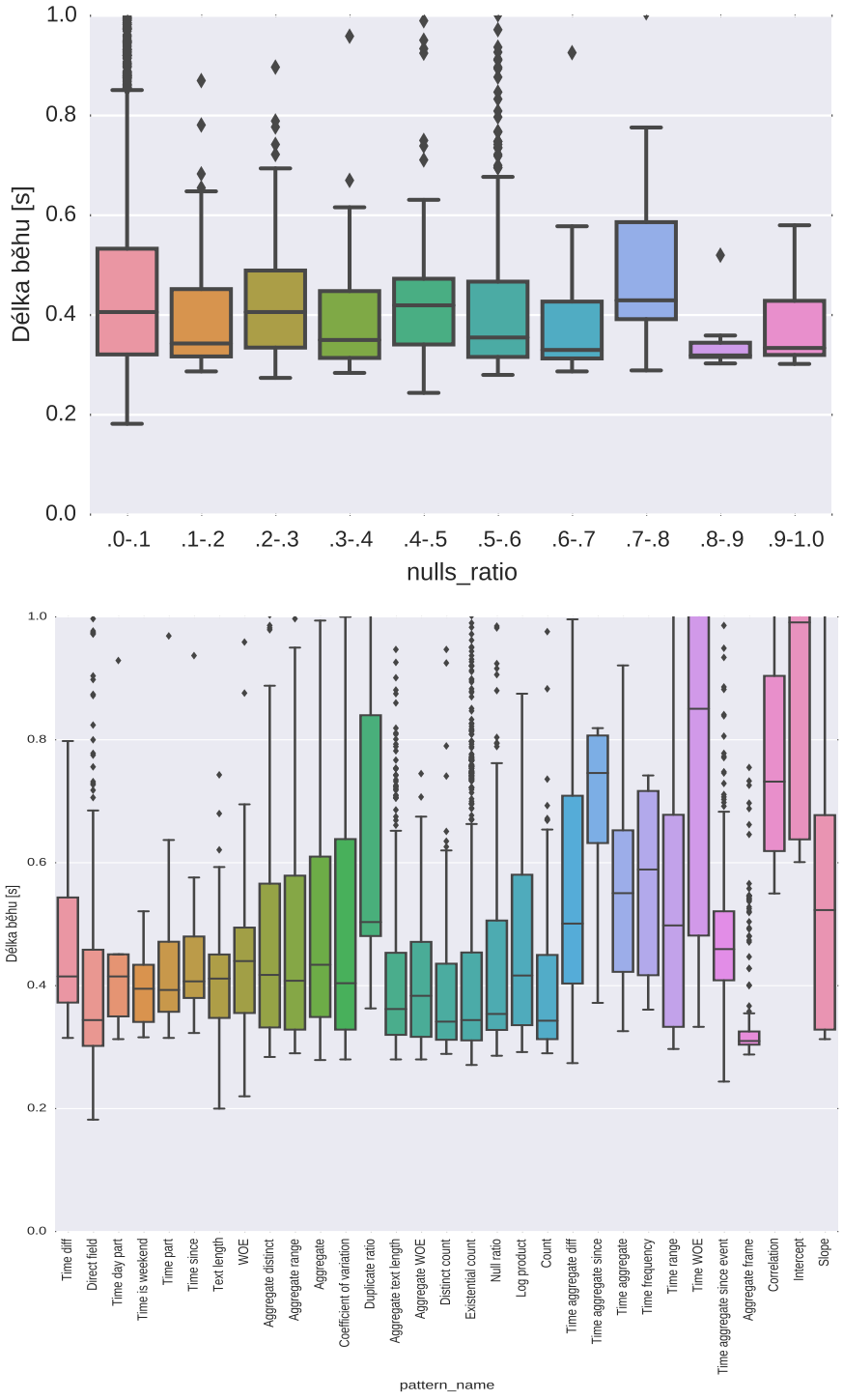
Tabulka 4.5: Důležitost příznaků pro predikci délky běhu z algoritmu XGBoost

Obdobně jako při výběru nejvhodnějších meta-příznaků pro predikci hodnot míry Chi2 (při použití algoritmu XGBoost) byla vytvořena množina meta-příznaků pro predikci délky běhu na základě tabulky 4.5. Do výsledné množiny

nejlepších meta-příznaků pro predikci délky běhu byly vybrány ty, které zlepšovaly přesnost míry Spearmanův korelační koeficient:

1. landmark_run_time — Napočítaná odhadovaná doba délky běhu z landmarkovacích příznaků.
2. landmark_chi2 — Vyšší odhadované hodnoty Chi2 mohou znamenat vyšší náročnost nápočtu příznaku a tedy delší délku běhu.
3. pattern_name — Několik konkrétních šablon je výpočetně náročnějších na nápočet příznaku.
4. nulss_ratio — Poměr null hodnot ve sloupci může ovlivnit dobu nápočtu.
5. avg_row_length — Průměrná délka řádku v tabulce.

Vztah mezi vybranými meta-příznaky a cílovým atributem Délka běhu je zobrazen na grafu 4.2.




Obrázek 4.2: Grafy zobrazující vztah mezi některými meta-příznaky a cílovým atributem Délka běhu. Rozsah hodnot na ose y je omezen na interval [0;1] kvůli odlehlým hodnotám.

## 4.4 Prediktivní modely

Parametry jednotlivých prediktivních algoritmů byly optimalizovány metodou Grid Search. Tato metoda funguje tak, že pro každý optimalizovaný parametr se nadefinuje množina hodnot, kterých může nabývat. Následně je měřena přesnost modelu na celém vstupním datasetu pro každou kombinaci hodnot zvolených parametrů. Zvolené hodnoty parametrů jsou uvedeny v následujích podsekcích.

### 4.4.1 Měření kvality predikčních modelů

#### 4.4.1.1 K-násobná křížová validace

Tento typ křížové validace funguje tak, že celý učící dataset se rozdělí do $k$ podmnožin. Predikční model je iterativně naučen vždy na $k$-1 různých podmnožinách a $k$-tá podmnožina slouží jako testovací množina během učení. Celková prediktivní přesnost modelu se počítá jako průměr z měření přesnosti modelů v jednotlivých iteracích. Tento typ měření lze použít pro vyhodnocení přesnosti základních prediktivních modelů.

#### 4.4.1.2 Míra KI

Tato míra (definice 1) je měřena na jednotlivých datasetech a to tak, že predikční model je natrénován na všech datasetech vyjma toho, u kterého je měřena kvalita predikce. Tento typ měření je použit jak u základních prediktivních modelů, tak při měření výsledné přesnosti při kombinaci více modelů.

Míra KI vyjadřuje poměr dvou obsahů — optimálního a predikovaného. Pro změření hodnoty KI jsou potřeba 3 křivky, jenž jsou definovány součtem hodnot skutečné naměřené míry Chi2 (osa y) a součtem skutečné naměřené délky běhu (osa x) příznaků.

V případě vykreslení míry Chi2 bylo nutné řešit duplicitu příznaků, jelikož je žádoucí rozlišit první napočtený příznak ze skupiny duplicit od jeho následných duplicit. Tento fakt je řešen tak, že hodnota Chi2 je přiřazena prvnímu příznaku v pořadí ze skupiny duplicit, a později napočteným duplicitním příznakům je přiřazena nulová hodnota Chi2.

Definice použitých křivek:

1. Náhodná křivka — představuje náhodně vygenerované pořadí příznaků k nápočtu. Data pro tuto křivku jsou získána následujícím postupem:
   a) $N$-krát se vygeneruje náhodné pořadí příznaků.
   b) V každé iteraci se podle duplicity upraví hodnota Chi2 jednotlivým příznakům. To znamená, že v každé skupině duplicit je hodnota Chi2 přiřazena pouze prvnímu příznaku v pořadí, ostatním duplicitním příznakům se nastaví nulová hodnota Chi2.

c) Po vygenerování $n$ náhodných pořadí příznaků se hodnoty Chi2 a délka běhu (které jsou potřeba pro vykreslení náhodné křivky) z jednotlivých iterací na stejném pořadí seskupí pomocí průměru. Tím jsou získány hodnoty pro vykreslení náhodné křivky.

d) Z $n$ vygenerovaných náhodných pořadí se získá predikční interval hodnot Chi2 na hladině významnosti $\alpha = 0.05$.

2. Optimální křivka — příznaky jsou seřazené podle funkce pro optimální pořadí příznaků $fitness$ (definice 3.1) dle skutečně naměřených hodnot.
3. Predikovaná křivka — příznaky jsou seřazené pouze podle predikovaných hodnot:

a) Model pro predikci hodnot Chi2 — příznaky jsou seřazeny sestupně dle predikovaných hodnot Chi2.

b) Model pro predikci délky běhu — příznaky jsou seřazeny vzestupně dle predikovaných hodnot délky běhu.

c) Kombinace predikce Chi2 a délky běhu — příznaky jsou seřazeny sestupně dle predikovaných hodnot doplněných do metriky $\frac{Chi2}{délka\_běhu}$.

d) Kombinace predikce Chi2, délky běhu a duplicity — příznaky jsou seřazené dle funkce $fitness$ pro optimální pořadí příznaků (definice 3.1) doplněné predikovanými hodnotami.

Čím blíže je výsledek míry KI hodnotě 1, tím lepší je predikce pořadí příznaků pro nápočet.

### 4.4.2 Model pro predikci Chi2

#### 4.4.2.1 Algoritmus ElasticNet

Pro algoritmus ElasticNet byly pomocí metody Grid Search optimalizovány parametry *alpha* a *l1_ratio*. Výsledek optimalizace je vypsán v tabulce 4.6.

| **Parametr** | **Popis** | **Testované hodnoty** | **Nejlepší hodnota** |
|---|---|---|---|
| alpha | Míra penalizace pro L1 a L2. Čim nižší je hodnota parametru alpha, tim více model penalizuje méně prediktivní příznaky. | 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001 | 0.0001 |
| l1_ratio | Udává poměr mezi L1 a L2 penalizací $(... + \alpha * l1\_ratio * L1 + \alpha * (1 - l1\_ratio) * L2...)$. | 0, 0.5, 0.7, 0.8, 0.9, 1 | 0.5 |

Tabulka 4.6: Optimalizace parametrů algoritmu ElasticNet pomocí metody Grid Search

Naměřené výsledky predikčního modelu používajícího metodu ElasticNet pomocí 10-násobné křížové validace jsou vypsané v tabulce 4.7.

| **Míra** | **Naměřené hodnoty** |
|---|---|
| MSE | 52605973 |
| Pearsonův korelační koeficient | 0.85 [p-hodnota 9.10e-68] |
| Spearmanův korelační koeficient | 0.79 [p-hodnota 1.44e-118] |

Tabulka 4.7: Naměřené hodnoty modelu pro predikci hodnot Chi2 pomocí 10-násobné křížové validace. Použitým algoritmem byl ElasticNet.

#### 4.4.2.2 Algoritmus XGBoost

Pro algoritmus XGBoost byly metodou Grid Search optimalizované parametry *max_depth*, *n_estimators* a *learning_rate*. Výsledek optimalizace obsahuje tabulka 4.8.

| **Parametr** | **Popis** | **Testované hodnoty** | **Nejlepší hodnota** |
|---|---|---|---|
| max_depth | Definuje maximální hloubku rozhodovacích stromů. | 2, 3, 4, 6, 8, 10 | 4 |
| n_estimators | Počet vygenerovaných slabých modelů (rozhodovacích stromů). | 50, 80, 100, 150 | 150 |
| learning_rate | Koeficient pro zmenšení váhy příznaků po každé iteraci. | 0.01, 0.1, 0.2, 0.5 | 0.2 |

Tabulka 4.8: Optimalizace parametrů algoritmu XGBoost pomocí metody Grid Search

Naměřené hodnoty predikčního modelu pro Chi2 používající algoritmus XGBoost pomocí 10-násobné křížové validace jsou zobrazeny v tabulce 4.9.

| **Míra** | **Naměřené hodnoty** |
|---|---|
| MSE | 52966557 |
| Pearsonův korelační koeficient | 0.84 [p-hodnota 1.88e-43] |
| Spearmanův korelační koeficient | 0.91 [p-hodnota 2.89e-201] |

Tabulka 4.9: Naměřené hodnoty algoritmu XGBoost z 10-násobné křížové validace

#### 4.4.2.3 Porovnání algoritmů ElasticNet a XGBoost

Tabulka 4.10 obsahuje naměřené hodnoty míry KI na jednotlivých datasetech za použití algoritmu XGBoost nebo ElasticNet při predikci hodnot Chi2.

Ačkoliv algoritmus XGBoost dosahuje lepší přesnosti při 10-násobné křížové validaci v případě Spearmannova korelačního koeficientu, při testování na jednotlivých datasetech vychází lépe algoritmus ElasticNet.

Testování na jednotlivých datasetec na základě míry KI lépe reflektuje měření kvality predikce, proto byl pro predikci míry Chi2 v dalších experimentech zvolen algoritmus ElasticNet.

| **Dataset** | **XGBoost** | **ElasticNet** |
|---|---|---|
| Accidents | 0.55 | **0.85** |
| AustralianFootball | 0.48 | **0.49** |
| Basketball_men | -0.35 | **-0.13** |
| Biodegradability | **0.47** | -0.13 |
| Carcinogenesis | 0.10 | **0.15** |
| ccs | **0.97** | 0.96 |
| Chess | 0.83 | 0.83 |
| CORA | 0.19 | **0.32** |
| financial | 0.08 | **0.48** |
| Hepatitis_std | **0.71** | 0.33 |
| Mondial | 0.42 | **0.46** |
| nations | 0.78 | **0.86** |
| PremierLeague | -0.07 | **0.67** |
| PTE | **0.73** | 0.37 |
| Student_loan | 0.61 | **0.75** |
| VisualGenome | 0.80 | **0.92** |
| Walmart | -0.25 | **0.95** |
| WebKP | 0.58 | -0.60 |
| world | 0.11 | **0.45** |
| Počet vítězství | 4 | **13** |

Tabulka 4.10: Porovnání predikčního modelu Chi2 při použití algoritmu XGBoost nebo ElasticNet pomocí naměřených hodnot KI na jednotlivých datasetech

### 4.4.3 Model pro predikci třídy Chi2

Pro klasifikaci třídy Chi2 byl použit algoritmus logistická regrese. Tabulka 4.11 obsahuje optimalizované parametry *penalty* a $C$ pomocí metody Grid Search.

| **Parametr** | **Popis** | **Testované hodnoty** | **Nejlepší hodnota** |
|---|---|---|---|
| penalty | Typ penalizace, která se má aplikovat. | l1, l2 | l1 |
| C | Inverzní hodnota pro parametr $\lambda$ ($C = \frac{1}{\lambda}$). | 0.001, 0.01, 0.1, 1, 10, 100, 1000 | 0.1 |

Tabulka 4.11: Optimalizace parametrů pro logistickou regresi pomocí metody Grid Search

Naměřené hodnoty z klasifikace třídy Chi2 pomocí algoritmu logistická regrese za použití 10-násobné křížové validace jsou vypsány v tabulce 4.12. Naměřené hodnoty jsou přílíš optimistické, jelikož ve vstupních datech dominuje třída Chi2 s hodnotou 1 (repzentující hodnoty Chi2 větší než 0). Na tento fakt jsou citlivé především míry F-Measure, senzitivita a přesnost.

| **Míra** | **Naměřené hodnoty** |
|---|---|
| F-Measure | 0.97 |
| AUC | 0.95 |
| Senzitivita | 1.00 |
| Přesnost | 0.96 |

Tabulka 4.12: Naměřené hodnoty klasifikace z 10-násobné křížové validace

Tabulka 4.13 obsahuje naměřené hodnoty klasifikace při použití modelu trénovaném na datech s vyváženým poměrem tříd.

| **Míra** | **Naměřené hodnoty** |
|---|---|
| F-Measure | 0.89 |
| AUC | 0.94 |
| Senzitivita | 0.97 |
| Přesnost | 0.82 |

Tabulka 4.13: Naměřené hodnoty klasifikace z 10-násobné křížové validace po vybalancování tříd při učení modelu

### 4.4.4 Model pro predikci Chi2 a třídy Chi2

Zkombinování těchto 2 prediktivních modelů funguje následovně:

1. Na vstupní dataset se aplikuje regresní algoritmus ElasticNet pro predikci hodnoty Chi2.
2. Následně se pro vstupní dataset predikuje třída Chi2 pomocí klasifikačního algoritmu logistická regrese.

3. Predikované hodnoty Chi2 z kroku 1 jsou přepsány na hodnotu 0, pokud byly v kroku 2 klasifikovány do nulové třídy Chi2.

Tabulka 4.14 obsahuje srovnání predikce Chi2 pomocí regresního modelu a regresního modelu rozšířeného o klasifikaci nulových hodnot.

Během ladění klasifikačního algoritmu (jak meta-příznaků, tak parametrů algoritmu) se nepodařilo natrénovat klasifikační model tak, aby se zlepšila klasifikace nulových hodnot. Naučený predikční model tedy převážně klasifikuje do nenulové třídy Chi2, proto se touto kombinací nepodařilo zvýšit predikční přesnost.

| **Dataset** | **Chi2** | **Kombinace Chi2 a třídy Chi2** |
|---|---|---|
| Accidents | 0.85 | 0.85 |
| AustralianFootball | 0.49 | 0.49 |
| Basketball_men | -0.13 | -0.13 |
| Biodegradability | -0.13 | -0.13 |
| Carcinogenesis | **0.15** | 0.13 |
| ccs | 0.96 | 0.96 |
| Chess | 0.83 | 0.83 |
| CORA | 0.24 | **0.32** |
| financial | **0.48** | 0.47 |
| Hepatitis_std | 0.33 | 0.33 |
| Mondial | 0.46 | 0.46 |
| nations | **0.86** | 0.62 |
| PremierLeague | 0.67 | 0.67 |
| PTE | 0.37 | 0.37 |
| Student_loan | 0.75 | 0.75 |
| VisualGenome | 0.92 | 0.92 |
| Walmart | 0.95 | 0.95 |
| WebKP | **-0.60** | -0.61 |
| world | 0.45 | 0.45 |
| Počet vítězství | 4 | 1 |

Tabulka 4.14: Porovnání predikčního modelu Chi2 a modelu Chi2 rozšířeného o klasifikaci třídy Chi2 pomocí naměřených hodnot KI na jednotlivých datasetech

### 4.4.5 Model pro predikci délky běhu

Tabulka 4.15 obsahuje výčet optimalizovaných parametrů pro algoritmus XGBoost metodou Grid Search.

| **Parametr** | **Popis** | **Testované hodnoty** | **Nejlepší hodnota** |
|---|---|---|---|
| max_depth | Definuje maximální hloubku rozhodovacích stromů. | 2, 3, 6, 8, 10 | 3 |
| n_estimators | Počet vygenerovaných slabých modelů (rozhodovacích stromů). | 50, 80, 100, 150 | 100 |
| learning_rate | Koeficient pro zmenšení váhy příznaků po každé iteraci. | 0.01, 0.1, 0.2, 0.5 | 0.1 |

Tabulka 4.15: Optimalizace parametrů pro XGBoost pomocí metody Grid Search

Naměřené hodnoty predikce délky běhu pomocí algoritmu XGBoost za použití 10-násobné křížové validace je zobrazena v tabulce 4.16.

| **Míra** | **Naměřené hodnoty** |
|---|---|
| MSE | 0.09 |
| Pearsonův korelační koeficient | 0.66 [p-hodnota 7.25e-40] |
| Spearmanův korelační koeficient | 0.71 [p-hodnota 1.11e-90] |

Tabulka 4.16: Naměřené hodnoty algoritmu XGBoost z křížové validace

## 4.5 Kombinace predikčních modelů

Testovány byly dvě kombinace predikčních modelů. Jedna se skládá z predikce Chi2 a délky běhu, kdy predikované pořadí příznaků je sestupně setříděno podle poměru $\frac{Chi2}{délka\_běhu}$.

Druhá kombinace vychází z té první, je ale rozšířena ještě o predikci duplicity.

Naměřené výsledky kombinací se nachází v tabulce 4.18.

## 4.6 Vyhodnocení výsledků

Tabulka 4.18 obsahuje naměřené hodnoty míry KI predikčních modelů a jejich kombinací na 19 datasetech. Vykreslené grafy ke všem testovaným metalearnigovým modelům se nachází v příloze B.

První sloupec tabulky obsahuje horní mez predikčního intervalu náhodného pořadí ($\alpha = 0.05$).

Nejdůležitějším faktorem pro určení užitečnosti příznaků je míra Chi2. To je potvrzené naměřenými výsledky, kde predikce optimálního pořadí pouze na základě hodnoty míry Chi2 vychází lépe na 14 datasetech, než pokud by se použilo náhodné pořadí. Zatímco predikce pořadí pouze pomocí modelu pro délku běhu přináší zlepšení jen na 4 datasetech oproti pořadí náhodnému.

Podle naměřených výsledků rozšíření predikce Chi2 o klasifikaci nulových hodnot nepřináší žádné zlepšení. Proto nebyla klasifikace třídy Chi2 v dalších testovaných kombinacích modelů použita.

Meta-learningový model skládající se z kombinace predikce Chi2 a délky běhu zvýší přesnost míry KI na 9 datasetech oproti použití samotné predikce Chi2. Dá se tedy říci, že kombinace Chi2 s délkou běhu je přínosná.

Ještě lepší výsledek je naměřen při kompletní kombinaci faktorů (míra Chi2, délka běhu, duplicita) a tato kombinace tak vychází nejlépe.

## 4.7 Vyhodnocení pomocí binomiálního testu

Binomiální test se používá k vyhodnocení experimentů, ve kterých výsledek může nabývat dvou hodnot (úspěch/neúspěch). Tento test je použit pro ověření tvrzení, že predikované pořadí dosahuje významně lepších hodnot KI než horní mez predikčního intervalu ($\alpha = 0.05$) náhodného pořadí. Testován je nejlepší meta-learningový model, tedy kombinace míry Chi2, délky běhu a duplicity, kde bylo naměřeno 15 nejvyšších zlepšení na 19 datasetech.

P-hodnota binomiálního testu byla naměřena pomocí metody binom_test z matematické knihovny Scipy.

| | |
|---|---|
| Nulová hypotéza $H_0$ | Rozdíl mezi náhodným pořadí příznaků a predikovaným je nulový. |
| Alternativní (jednostranná) hypotéza $H_1$ | Rozdíl mezi náhodným pořadí příznaků a predikovaným je větší než nula. |
| Počet úspěchů | 15 |
| Počet pozorování | 19 |
| Hladina významnosti $\alpha$ | 0.025 |
| P-hodnota testované hypotézy | 3.28e-21 |
| **Závěr** | Na základě naměřené p-hodnoty je nulová hypotéza na hladině významnosti $\alpha = 0.025$ zamítnuta. |

Tabulka 4.17: Aplikace binomiálního testu na úspěšnost predikce

Na základě aplikovaného binomiálního testu 4.17 lze tvrdit, že naměřené hodnoty KI z predikce optimálního pořadí příznaků pomocí kombinace Chi2,

délky běhu a duplicity jsou významně lepší než u náhodného pořadí.

| **Dataset** | **Horní mez PI náhodného pořadí** ($\alpha = 0.05$) | **Chi2** | **Chi2+třída Chi2** | **Délka běhu** | **Chi2+délka běhu** | **Chi2+délka běhu+duplicita** |
|---|---|---|---|---|---|---|
| Accidents | 0.33 | 0.85 | 0.85 | 0.04 | 0.85 | **0.88** |
| AustralianFootball | 0.36 | 0.49 | 0.49 | 0.21 | 0.50 | **0.55** |
| Basketball_men | 0.11 | -0.13 | -0.13 | 0.09 | -0.07 | **0.28** |
| Biodegradability | **0.31** | -0.13 | -0.13 | 0.24 | 0.00 | 0.15 |
| Carcinogenesis | **0.32** | 0.13 | 0.13 | -0.10 | 0.12 | 0.11 |
| ccs | 0.94 | 0.96 | 0.96 | -0.40 | 0.96 | **0.98** |
| Chess | 0.29 | 0.83 | 0.83 | 0.03 | 0.84 | **0.86** |
| CORA | **0.56** | 0.32 | 0.32 | 0.37 | 0.44 | 0.42 |
| financial | 0.32 | 0.48 | 0.47 | -0.30 | 0.46 | **0.50** |
| Hepatitis_std | 0.28 | 0.33 | 0.33 | 0.13 | 0.34 | **0.45** |
| Mondial | 0.11 | 0.46 | 0.46 | 0.14 | 0.46 | **0.53** |
| nations | 0.34 | 0.86 | 0.62 | 0.39 | 0.86 | **0.88** |
| PremierLeague | 0.17 | 0.67 | 0.67 | 0.04 | 0.73 | **0.74** |
| PTE | 0.20 | 0.37 | 0.37 | -0.19 | 0.35 | **0.42** |
| Student_loan | 0.39 | 0.75 | 0.75 | -0.08 | 0.75 | **0.79** |
| VisualGenome | 0.82 | 0.92 | 0.92 | -0.63 | 0.92 | **0.96** |
| Walmart | 0.42 | 0.95 | 0.95 | 0.82 | **0.96** | **0.96** |
| WebKP | **0.54** | -0.61 | -0.61 | -0.16 | -0.61 | -0.52 |
| world | 0.28 | 0.45 | 0.45 | 0.32 | 0.46 | **0.47** |
| Počet vítězství | 4 | 0 | 0 | 0 | 2 | 15 |

Tabulka 4.18: Naměřené hodnoty KI prediktivních modelů a jejich kombinací

# Závěr

Cílem práce v teoretické části bylo vytvořit rešerši zabývající se využitím meta-learningu pro účely klasifikace. Většina literatury v oblasti meta-learningu se věnuje doporučování klasifikačních algoritmů na základě vstupních datasetů. V kapitole2 byly analyzovány 4 nástroje využívající meta-learning ve spojení s klasifikací, přičemž byla snaha vybrat trochu odlišné přístupy k meta-learningu. Z rešerše se dalo inspirovat použitými typy meta-příznaků.

Cílem práce v praktické části bylo navrhnout, implementovat a vyhodnotit meta-learningový model, který predikuje užitečnost příznaků v procesu propositionalize, na základě které se odhadne optimální pořadí příznaků k výpočtu.

Velká část práce byla věnována přípravě dat. Data v relační podobě bylo potřeba transformovat procesem propositionalizace do podoby jedné tabulky, k tomuto účelu byl využit nástroj Predictor Factory.

Součástí přípravy dat byla identifikace meta-příznaků a jejich výpočet. Meta-příznakům je věnována sekce 3.2. Byly zvoleny dvě skupiny meta-příznaků:

1. Jednoduché meta-příznaky, které popisují vlastnosti datasetu (jednotlivých tabulek a sloupců).
2. Landmarkovací meta-příznaky, které byly dopočítány jako průměr hodnot z rychle napočtených příznaků. V experimentální části se ukázalo, že v predikčních modelech jsou velice přínosné.

Na připravených trénovacích datech byly vyvíjené predikční meta-learningové modely, této implementační části jsou věnovány kapitoly 3 a 4. V průběhu vývoje byla jako vyhodnocovací metrika užitečnosti příznaků stanovena funkce $fitness$ 3.1. Výsledný meta-learningový model se proto skládá z predikce hodnot míry Chi2 (reprezentující relevanci), doby výpočtu a duplicity (redundance).

Testování a vyhodnocení predikčních modelů se nachází v kapitole 4. Stanovenou mírou pro vyhodnocení kvality predikovaného pořadí příznaků je míra KI. Hodnoty této míry byly naměřeny pro každý testovací dataset. Z naměřených výsledků funguje nejlépe meta-learningový model založený na kombinaci

predikce míry Chi2, délky běhu a duplicity, přičemž největší vliv na výsledek má predikce míry Chi2. Na základě naměřených výsledků a binomiálního testu s hladinou významnosti $\alpha = 0.025$ bylo ověřeno, že predikované pořadí dosahuje významně lepších hodnot KI než horní mez predikčního intervalu ($\alpha = 0.05$) náhodného pořadí příznaků.

# Literatura


[1] Lemke, C.; Budka, M.; Gabrys, B.: Metalearning: a survey of trends and technologies. *Artificial Intelligence Review*, ročník DOI: 10.1007/s10462-013-9406-y, 06 2013.

[2] Fawcett, T.: Introduction to ROC analysis. *Pattern Recognition Letters*, ročník 27, 06 2006: s. 861–874.

[3] Melo, F.: Area under the ROC Curve. In *Encyclopedia of Systems Biology*, Springer New York, 2013, ISBN 978-1-4419-9863-7, s. 38–39.

[4] Wood, T.: Using Mean Absolute Error for Forecast Accuracy *[online]*. 2012, [cit. 2017-08-28]. Dostupné z: `http://canworksmart.com/using-mean-absolute-error-forecast-accuracy/`

[5] Team, T. B.: Time Series with the BigML Dashboard. 2017.

[6] Stellwagen, E.: A Guide to Forecast Error Measurement Statistics and How to Use Them *[online]*. 2017, [cit. 2017-08-28]. Dostupné z: `http://www.forecastpro.com/Trends/forecasting101August2011.html`

[7] Lane, D. M.: Introduction to Statistics: An Interactive eBook *[online]*. [cit. 2017-08-28]. Dostupné z: `http://onlinestatbook.com/2/describing_bivariate_data/pearson.html`

[8] Lund, A.; Lund, M.: Spearman's Rank-Order Correlation *[online]*. [cit. 2017-10-27]. Dostupné z: `https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php`

[9] Chi-Square Test for Independence *[online]*. 2017, [cit. 2017-07-28]. Dostupné z: `https://stattrek.com/chi-square-test/independence.aspx`

[10] Brandenburger, T.; Furth, A.: Cumulative Gains Model Quality Metri. *Journal of Applied Mathematics and Decision Sciences*, 2009: str. 14.

[11] Duda, R. O.; Hart, P. E.; Stork, D. G.: Pattern Classification. *Wiley Interscience*, ročník xx, 01 2001.

[12] Zou, H.; Hastie, T.: Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 04 2005: s. 301 – 320.

[13] Introduction to Boosted Trees *[online]*. 2015-2016, [cit. 2017-07-28]. Dostupné z: `http://xgboost.readthedocs.io/en/latest/model.html`

[14] Reif, M.; Shafait, F.; a spol.: Automatic classifier selection for non-experts. *Pattern Analysis & Applications*, ročník 17, 02 2014.

[15] Doan, T.; Kalita, J.: Predicting run time of classification algorithms using meta-learning. *International Journal of Machine Learning and Cybernetics*, ročník 8, 07 2016.

[16] Lichman, M.: UCI Machine Learning Repository *[online]*, 2013, [cit. 2017-06-23]. Dostupné z: `http://archive.ics.uci.edu/ml`

[17] Romero, C.; Olmo, J.; Ventura, S.: A meta-learning approach for recommending a subset of white-box classification algorithms for Moodle datasets. 2013, [cit. 2017-06-30].

[18] Orriols-Puig, A.; Macia, N.; a spol.: The data complexity library, DCoL *[online]*. 2010, [cit. 2017-06-30]. Dostupné z: `https://github.com/nmacia/dcol`

[19] Iman, R. L.; Davenport, J. M.: Approximations of the critical region of the Friedman statistic. *Communications in Statistics-Theory and Methods*, ročník 9, 01 1980: s. 571–595.

[20] Lam, W.; Lai, K.: A Meta-Learning Approach for Text Categorization. 2001.

[21] Brazdil, P. B.; anf J. P. da Costa, C. S.: Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results. *Machine Learning*, ročník 50, 03 2003: s. 251–277.

[22] Motl, J.; Schulte, O.: The CTU Prague Relational Learning Repository *[online]*. 2015, [cit. 2017-07-28]. Dostupné z: `https://relational.fit.cvut.cz`

[23] Motl, J.; Schultes, O.: The CTU Prague Relational Learning Repository *[online]*. [cit. 2017-11-17]. Dostupné z: `https://arxiv.org/abs/1511.03086`

Příloha **A**

# Seznam použitých zkratek

**MLP** Multilayer perceptron

**SVM** Support vector machine

**SVR** Support vector regression

**MARS** Multi-Variate Adaptive Regression Splines

**KNN** Knearest Neighbor Regression

**PCA** Principal Components Analys

**RMSE** Root Mean Squared Error

**MAE** Mean Absolute Error

**MAD** Mean Absolute Deviation

**CPU** Central processing unit

**RAM** Random Access Memory

**AUC** Area Under an ROC Curve

**MSE** Mean squared error

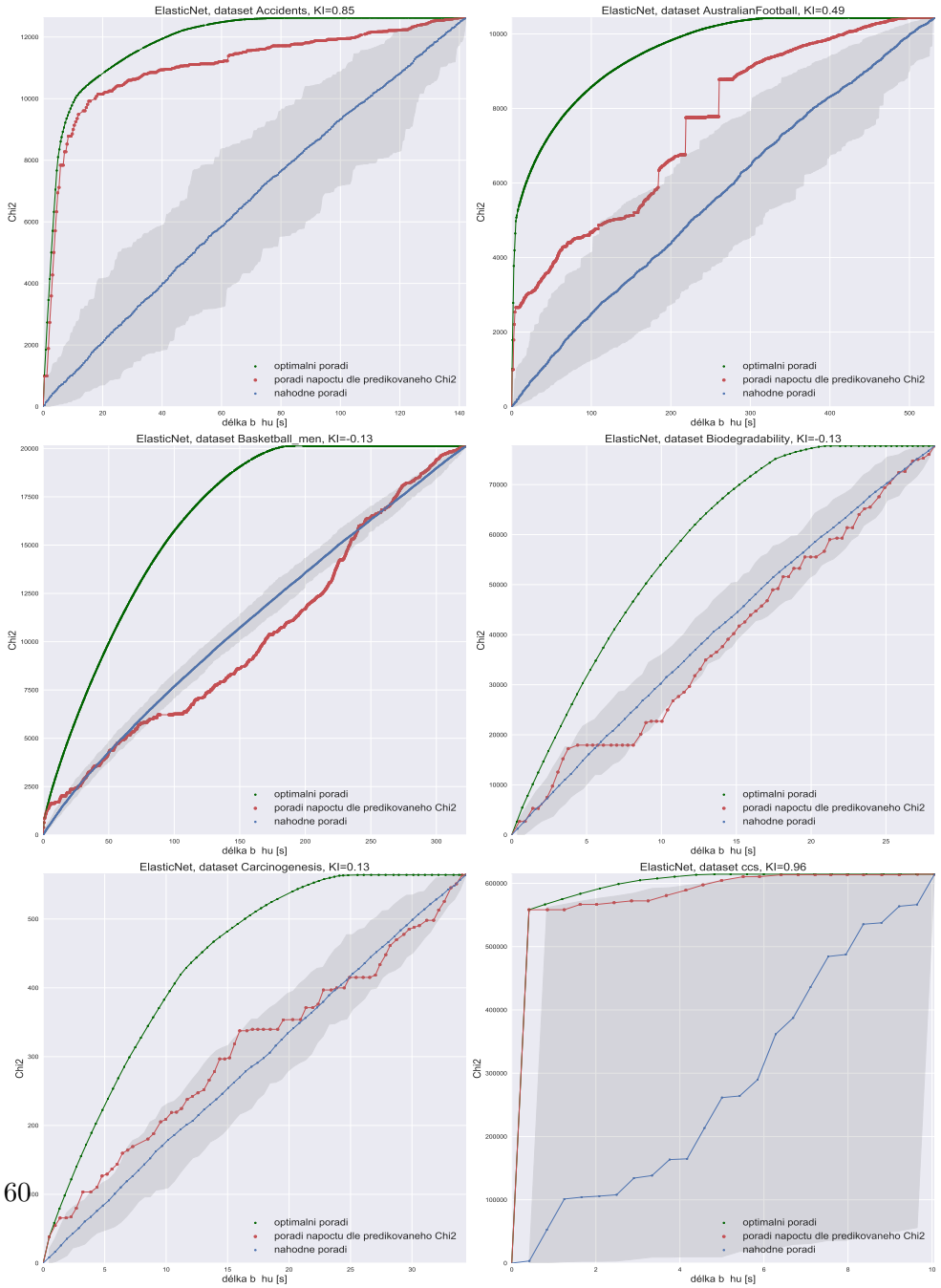
**XGBoost** Extreme Gradient Boosting

# Příloha B

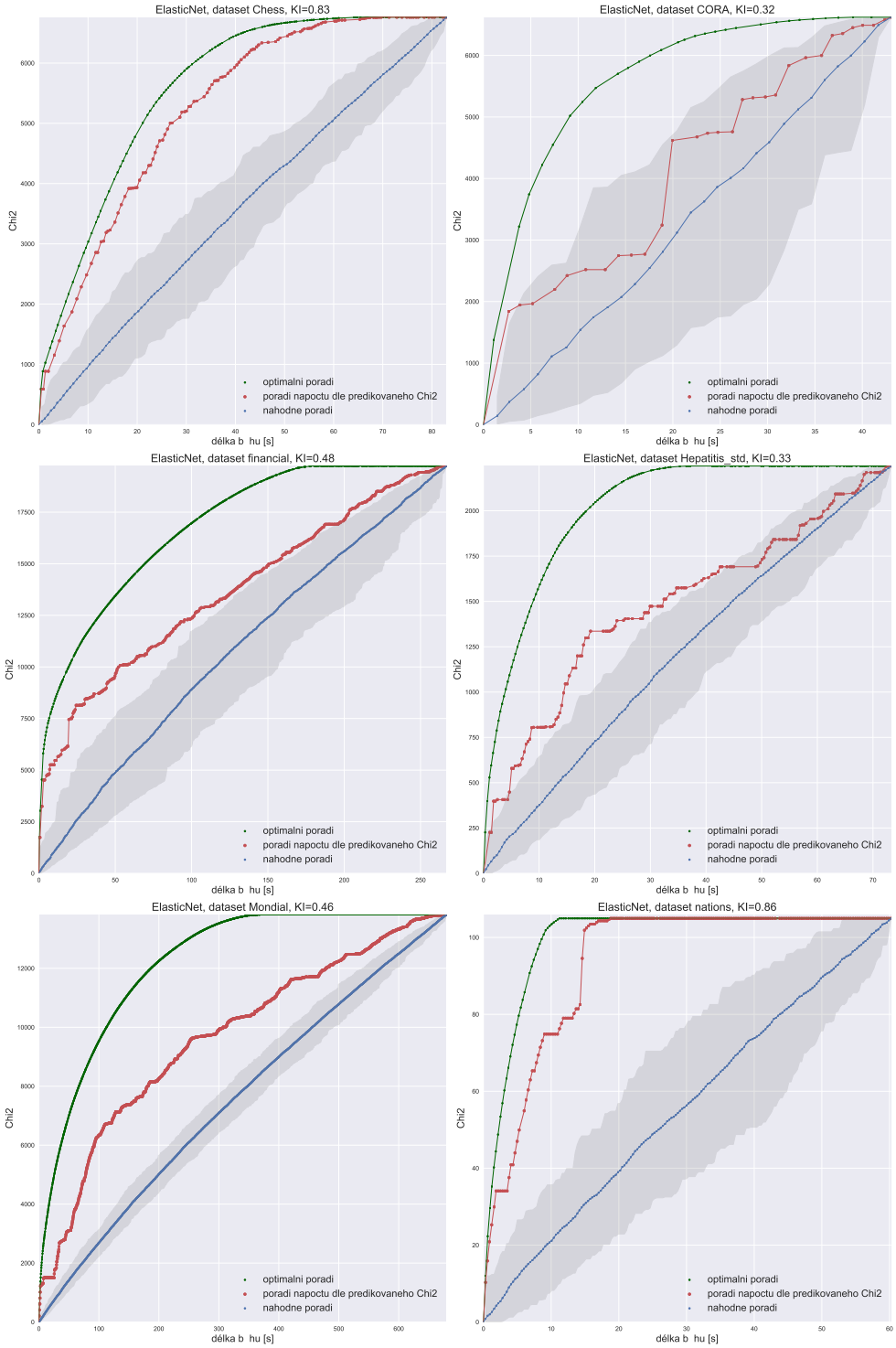
# Naměřené výsledky na všech datasetech

## B.1 Chi2

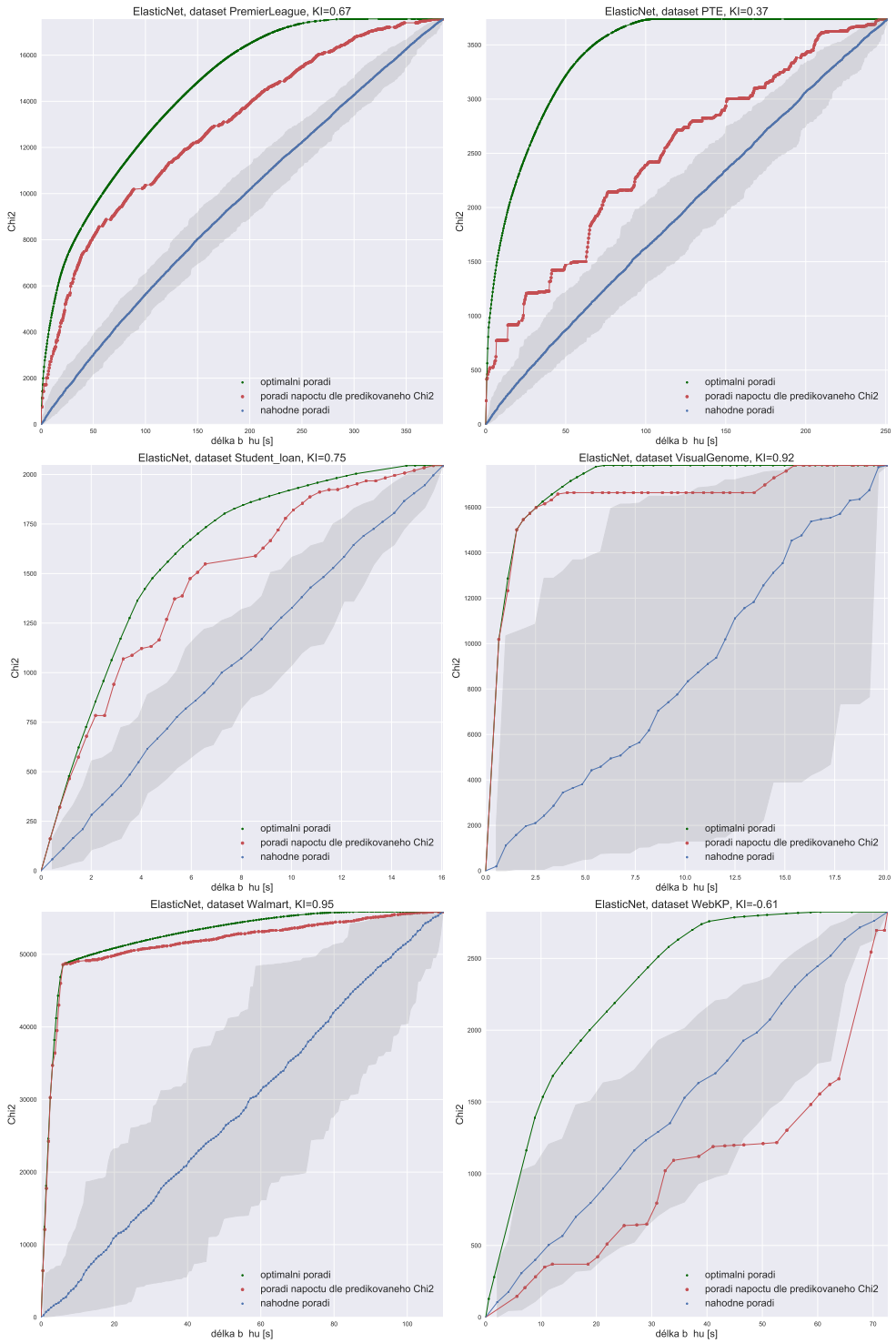
### B.1.1 Použití algoritmu ElasticNet



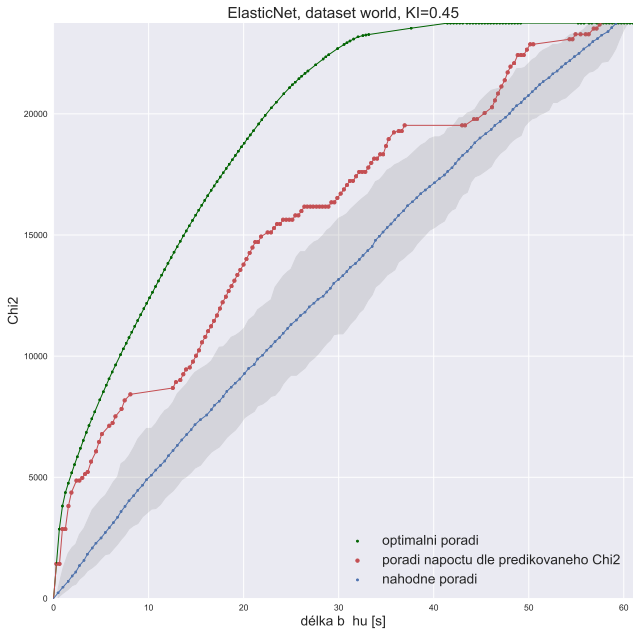

Obrázek B.1: Měření výsledků z algoritmu ElasticNet




Obrázek B.2: Měření výsledků z algoritmu ElasticNet

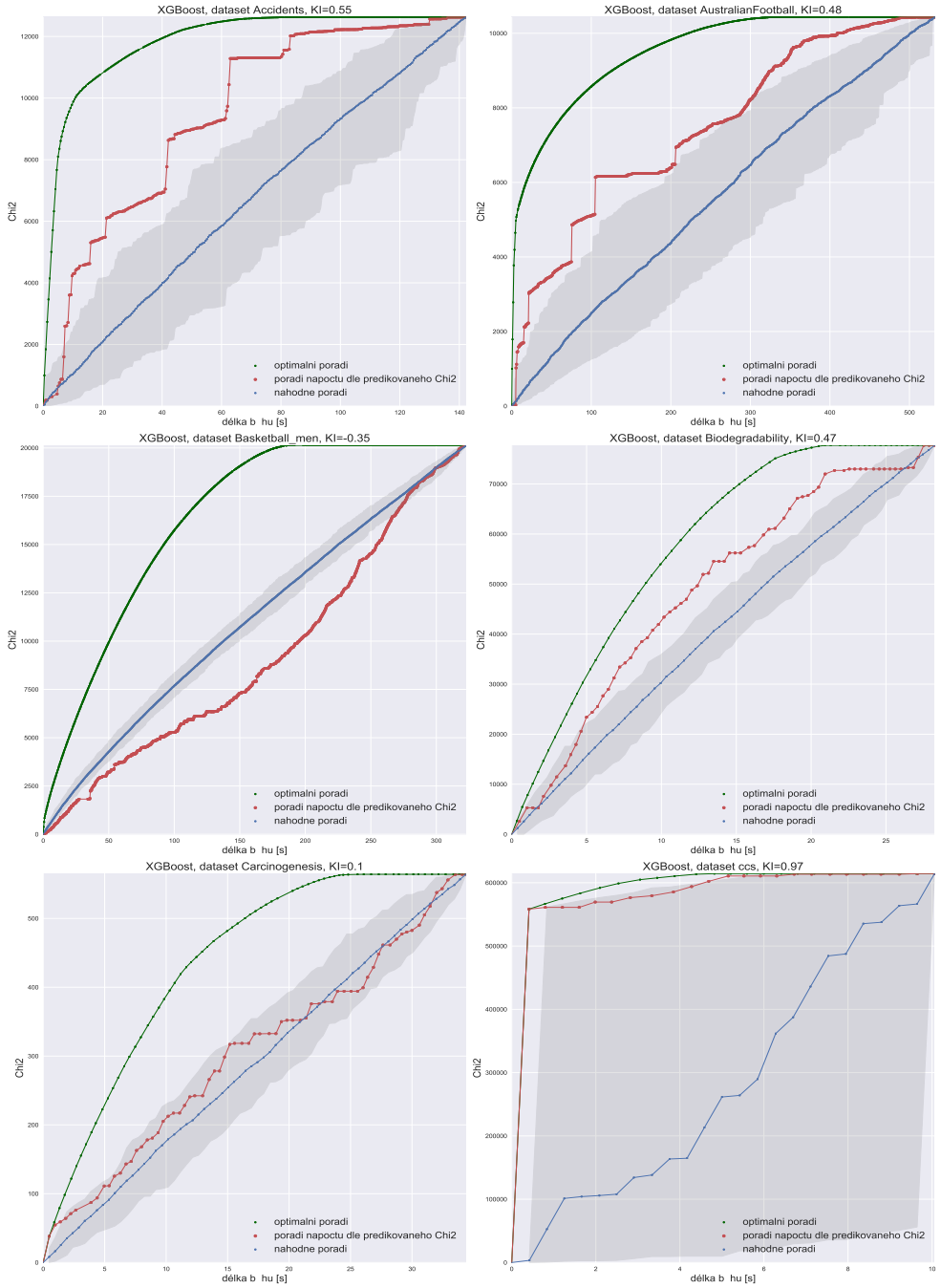



Obrázek B.3: Měření výsledků z algoritmu ElasticNet

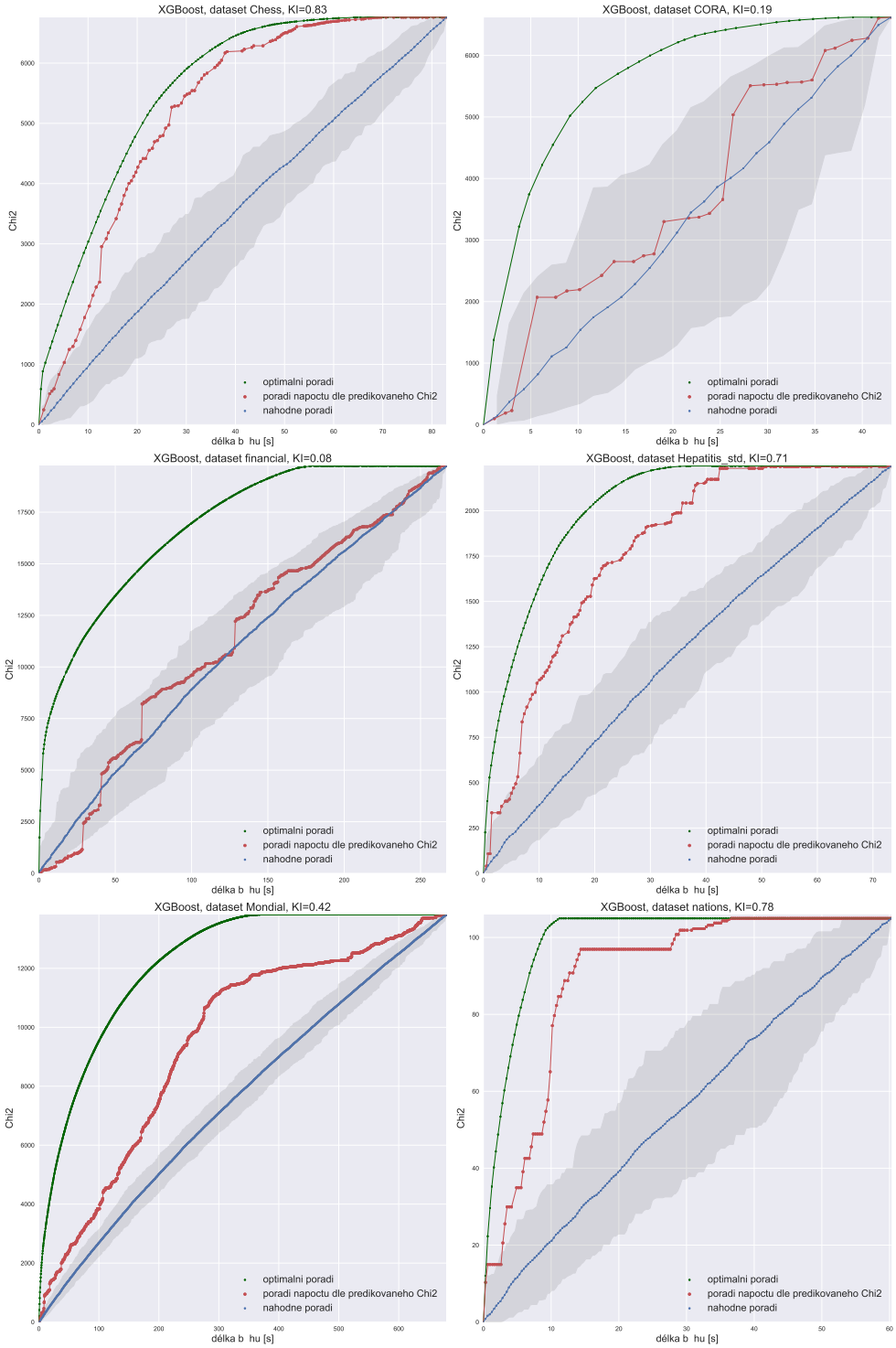



Obrázek B.4: Měření výsledků z algoritmu ElasticNet

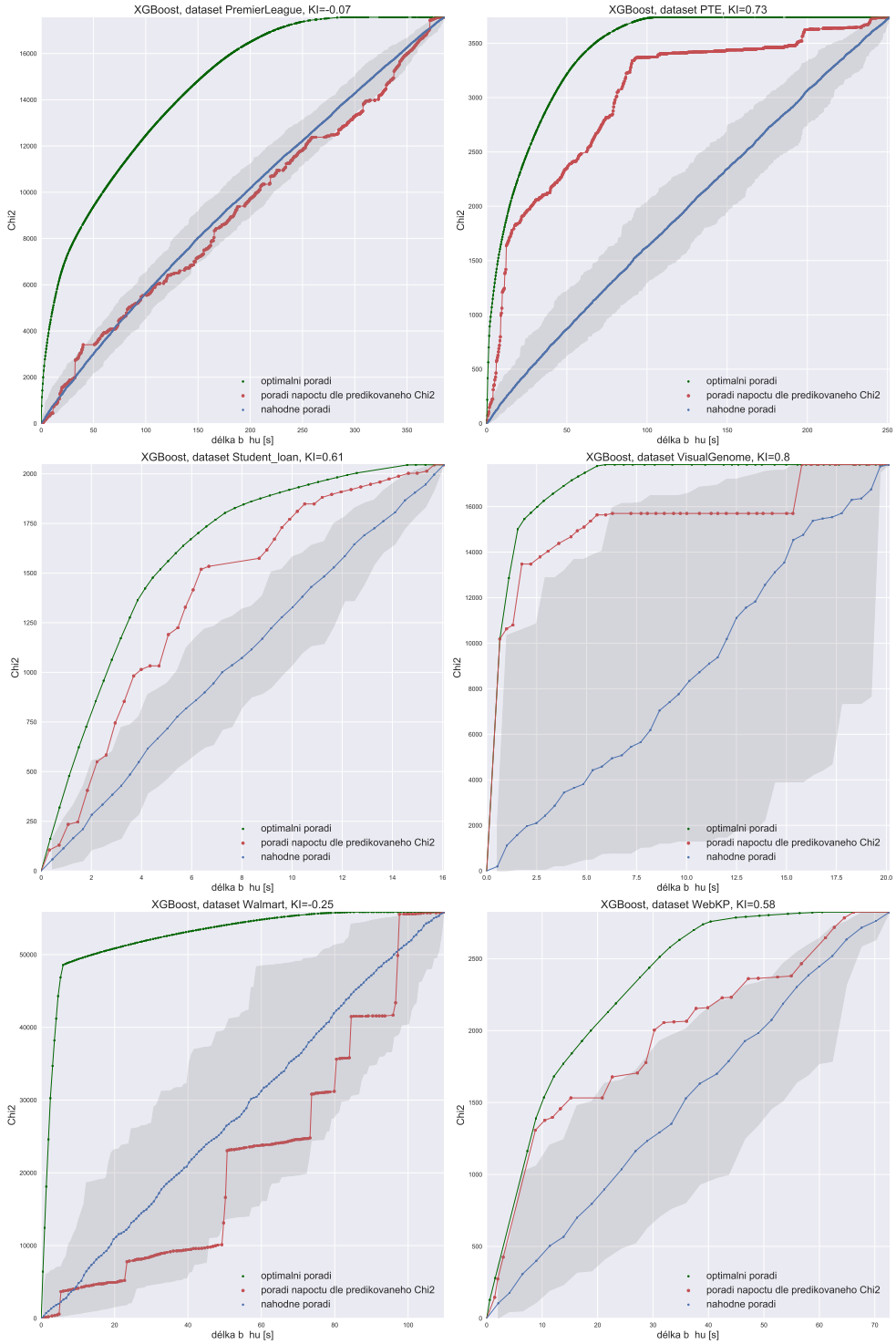
### B.1.2 Použití algoritmu XGBoost



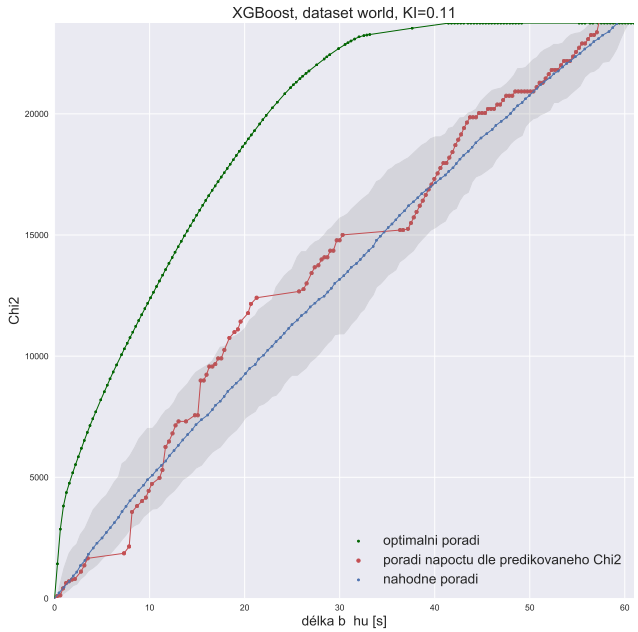

Obrázek B.5: Měření výsledků z algoritmu XGBoost




Obrázek B.6: Měření výsledků z algoritmu XGBoost

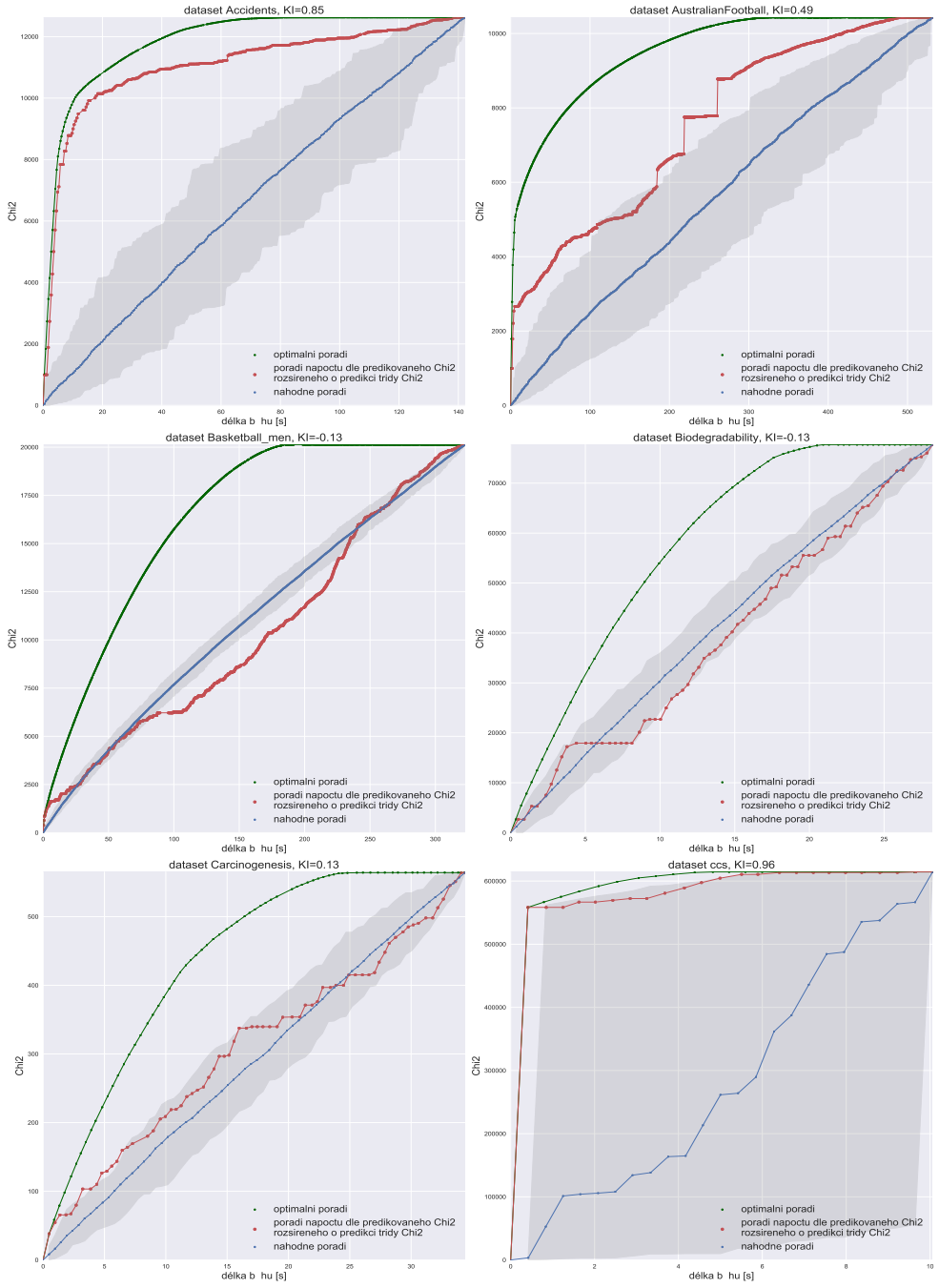



Obrázek B.7: Měření výsledků z algoritmu XGBoost

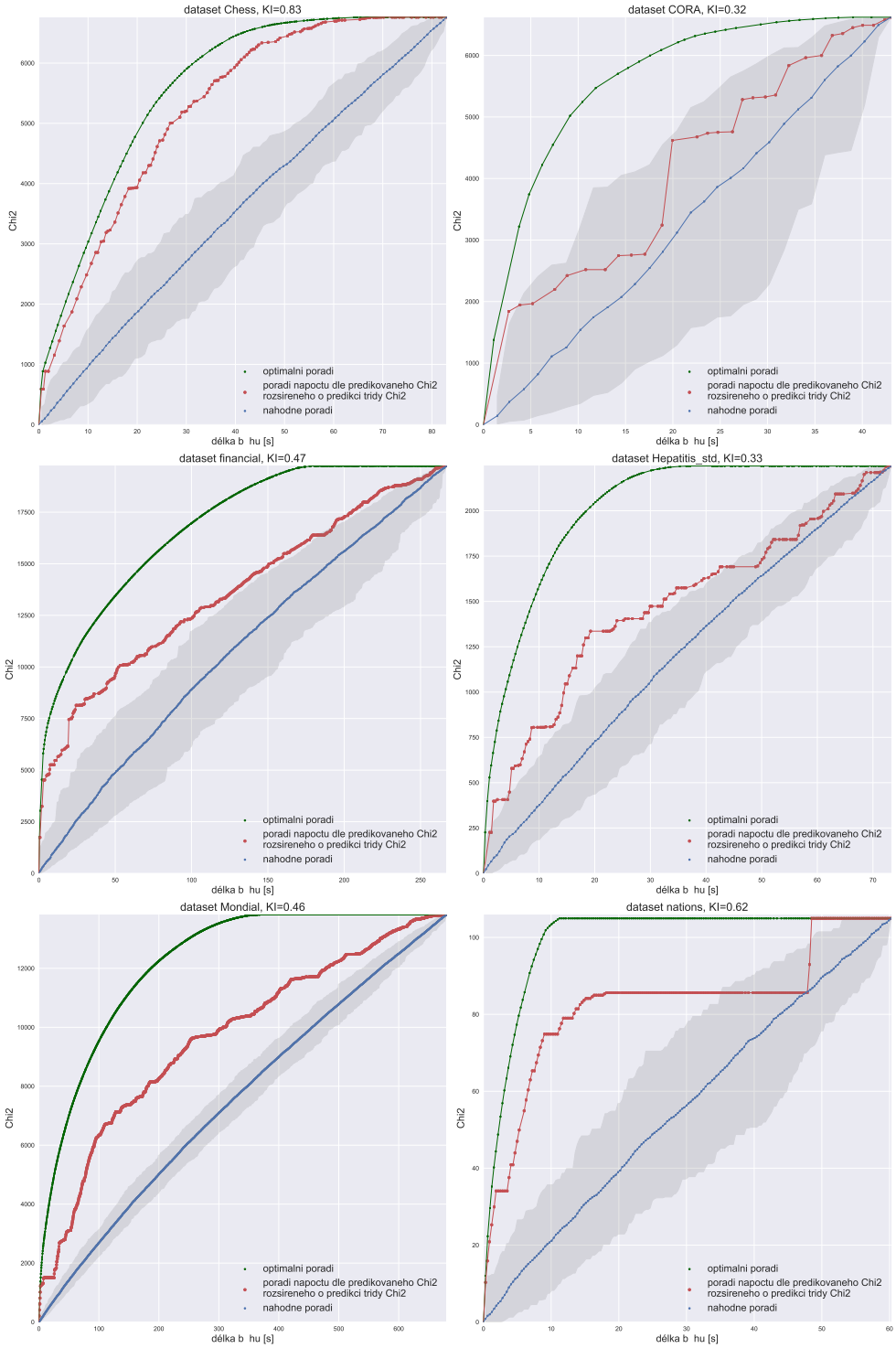



Obrázek B.8: Měření výsledků z algoritmu XGBoost

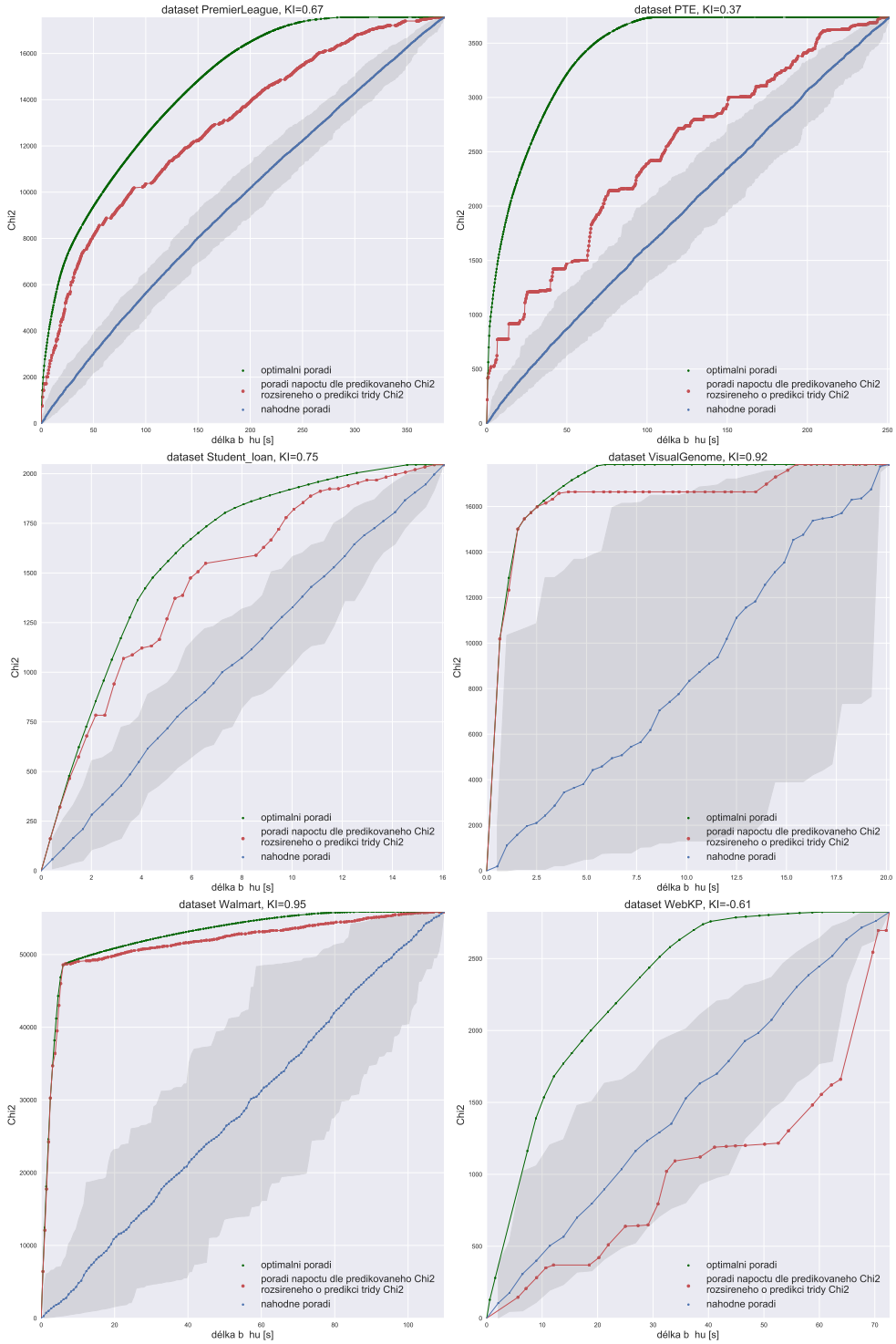
## B.2 Kombinace Chi2 a třídy Chi2



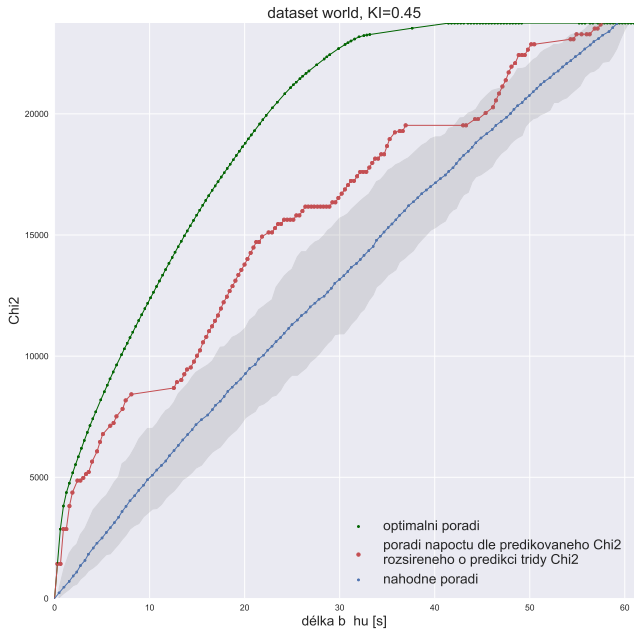

Obrázek B.9: Měření meta-learningového modelu za použití predikce míry Chi2 rozšířené o klasifikaci nulových hodnot




Obrázek B.10: Měření meta-learningového modelu za použití predikce míry Chi2 rozšířené o klasifikaci nulových hodnot

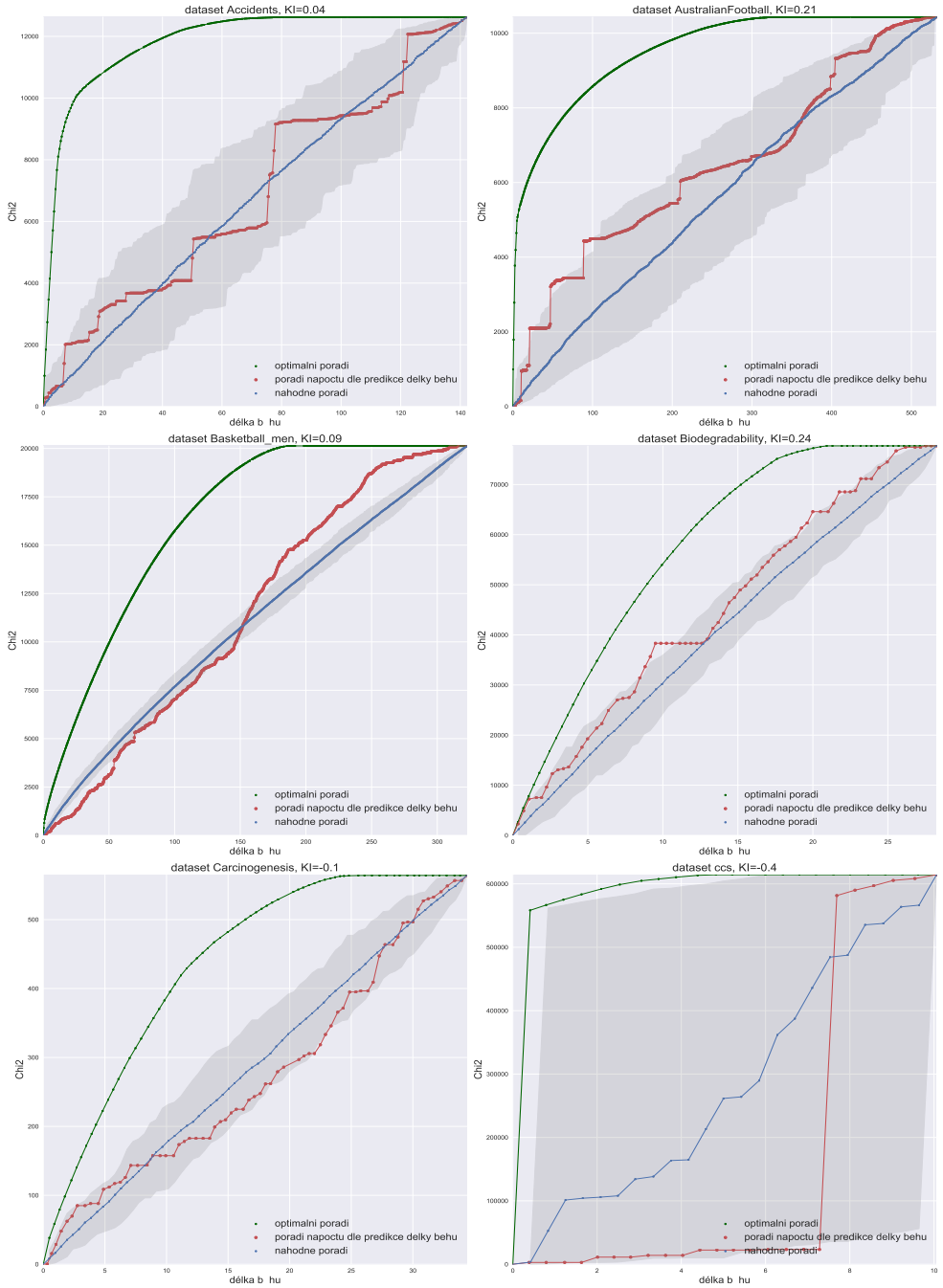



Obrázek B.11: Měření meta-learningového modelu za použití predikce míry Chi2 rozšířené o klasifikaci nulových hodnot

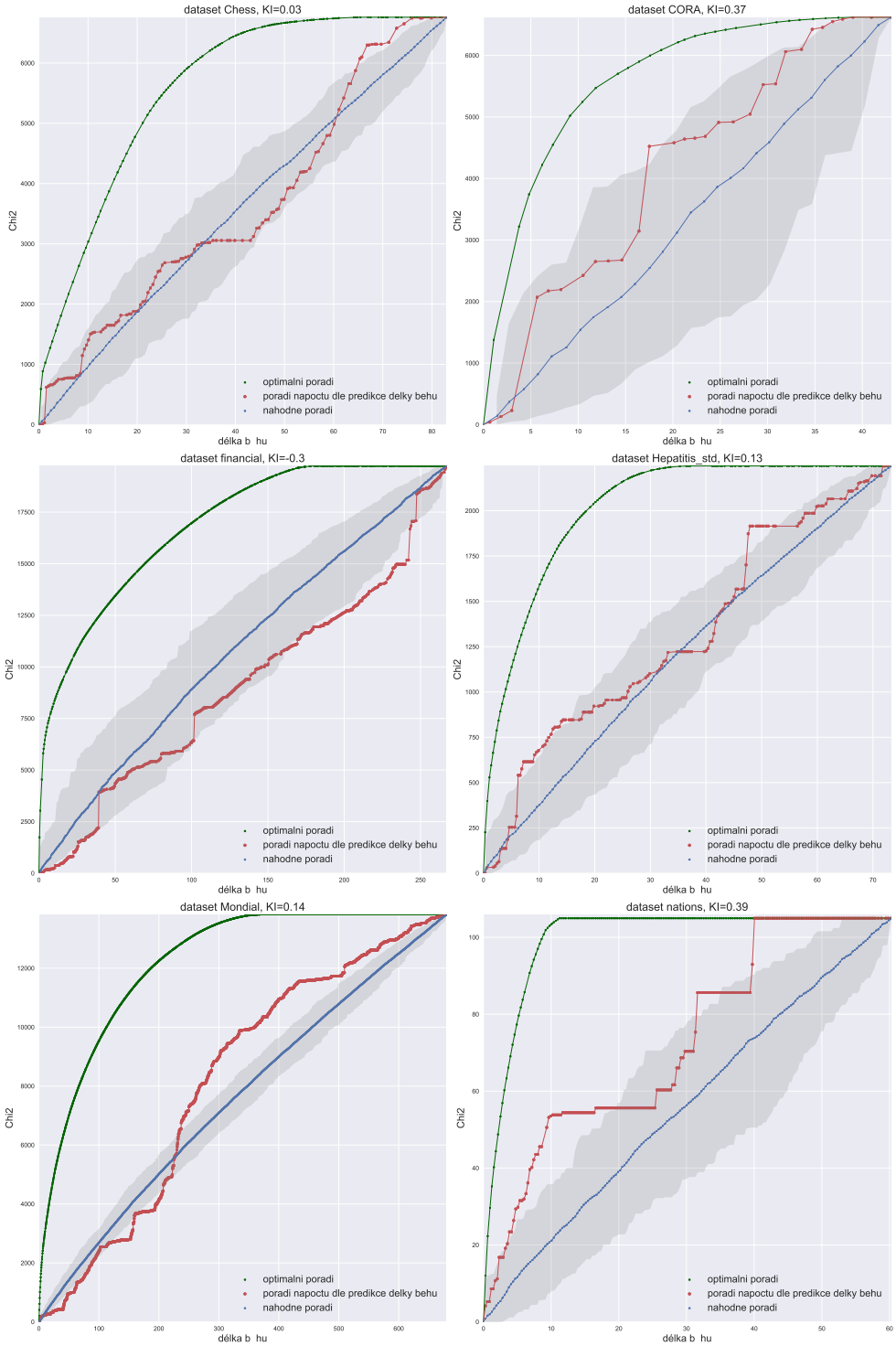



Obrázek B.12: Měření meta-learningového modelu za použití predikce míry Chi2 rozšířené o klasifikaci nulových hodnot

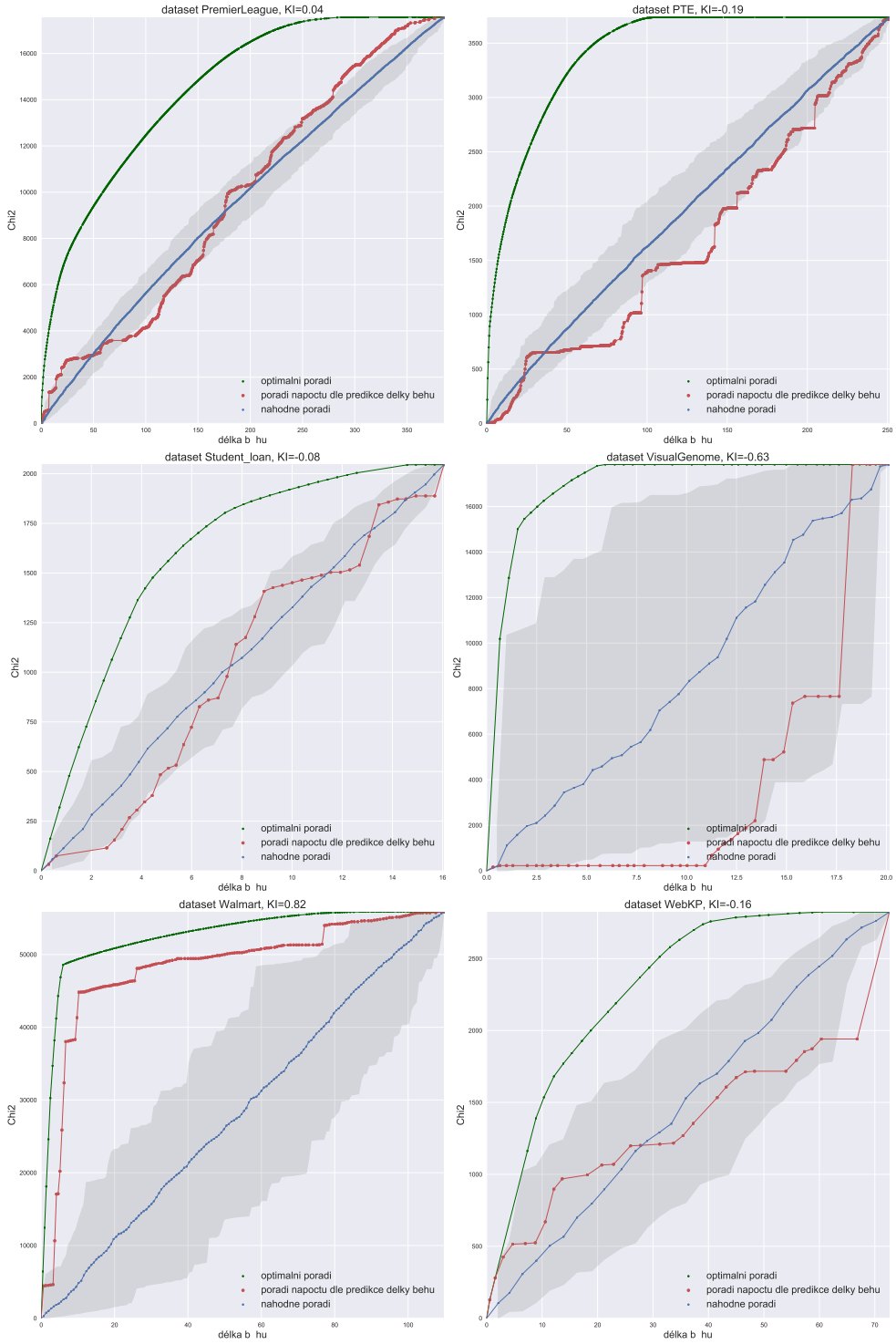
## B.3 Délka běhu



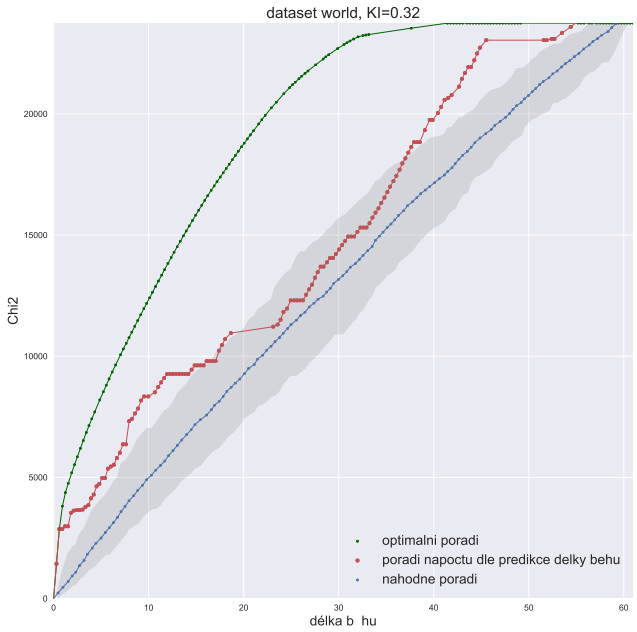

Obrázek B.13: Měření meta-learningového modelu za použití predikce délky běhu




Obrázek B.14: Měření meta-learningového modelu za použití predikce délky běhu

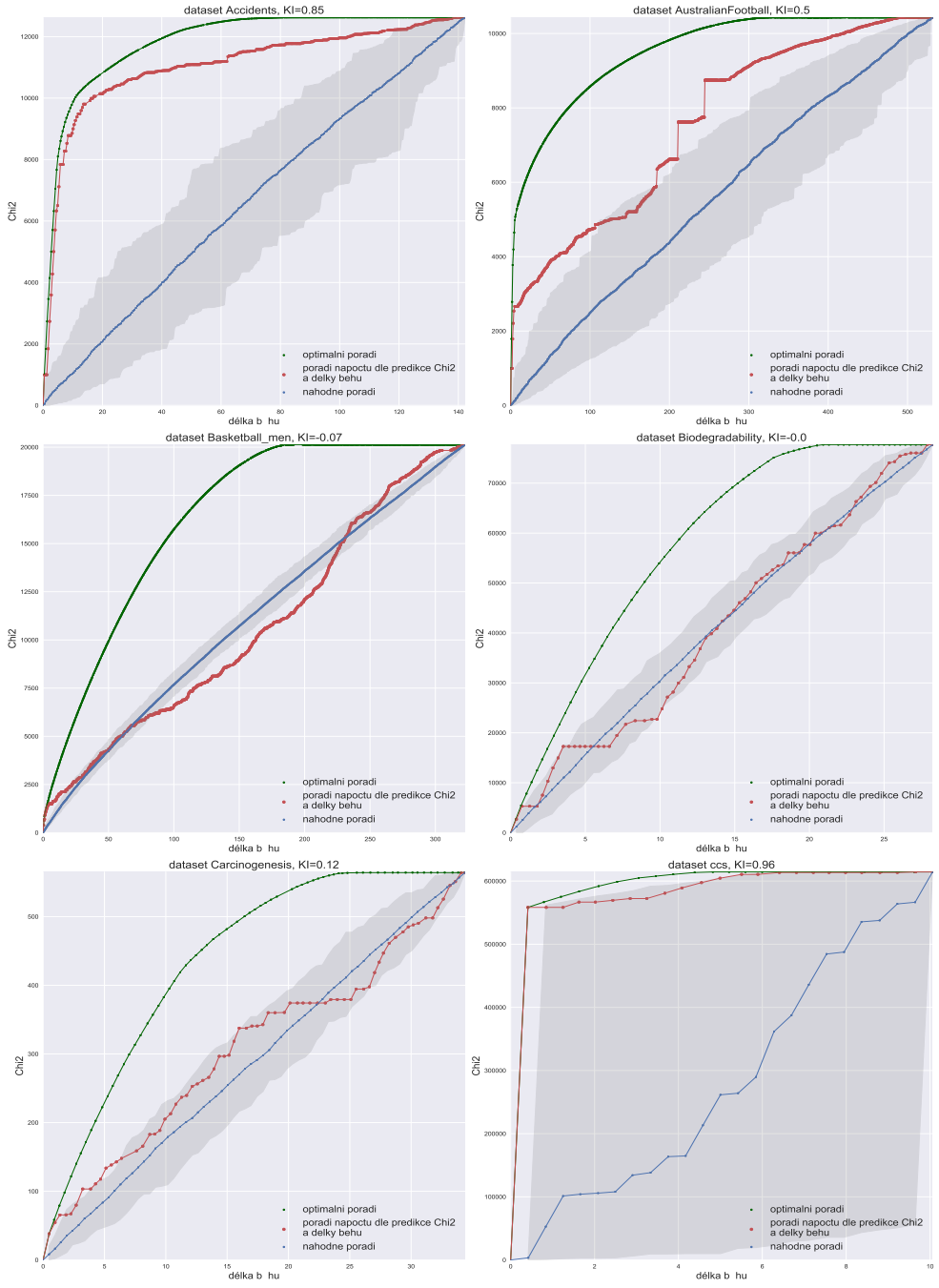



Obrázek B.15: Měření meta-learningového modelu za použití predikce délky běhu

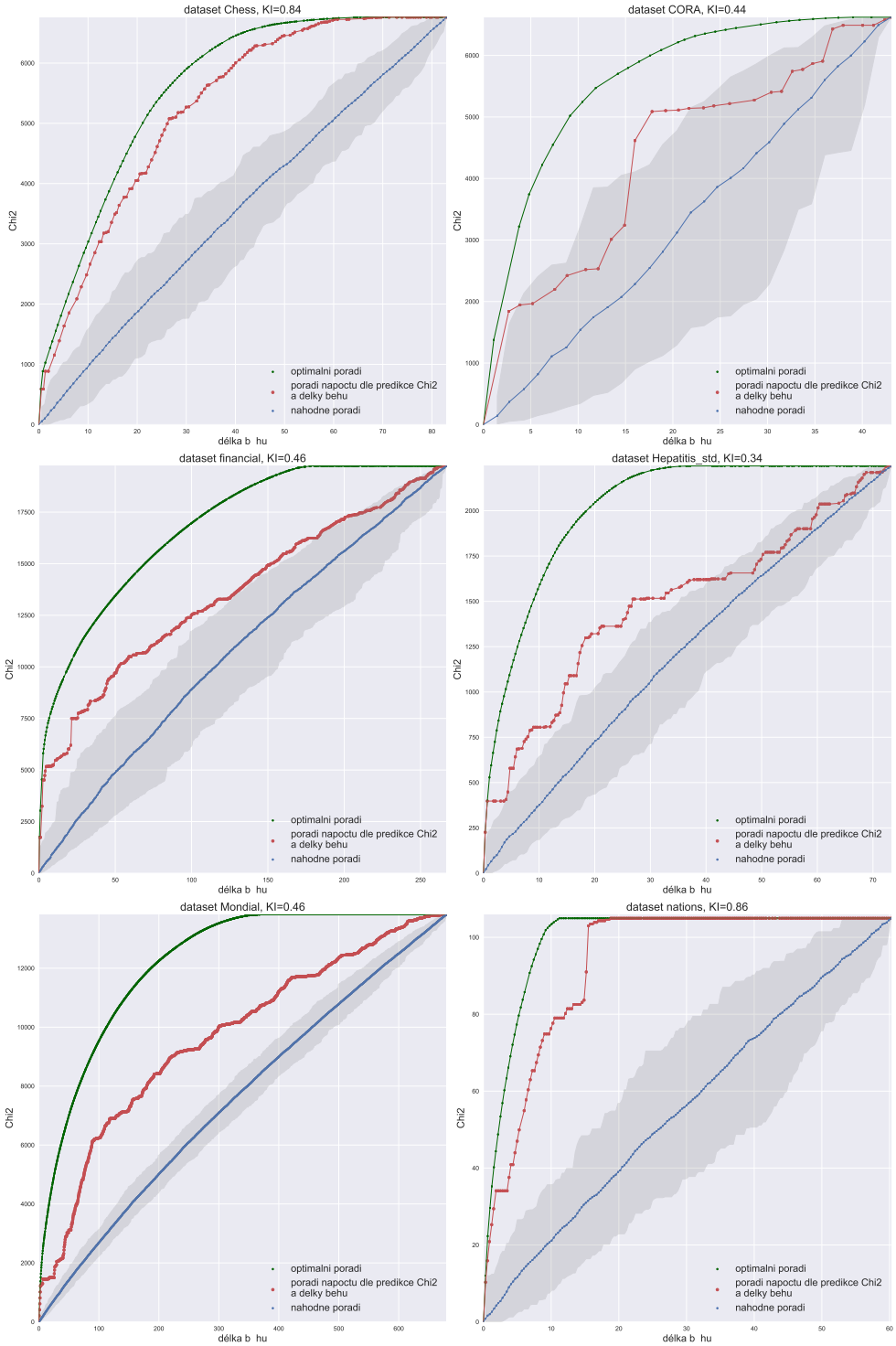



Obrázek B.16: Měření meta-learningového modelu za použití predikce délky běhu

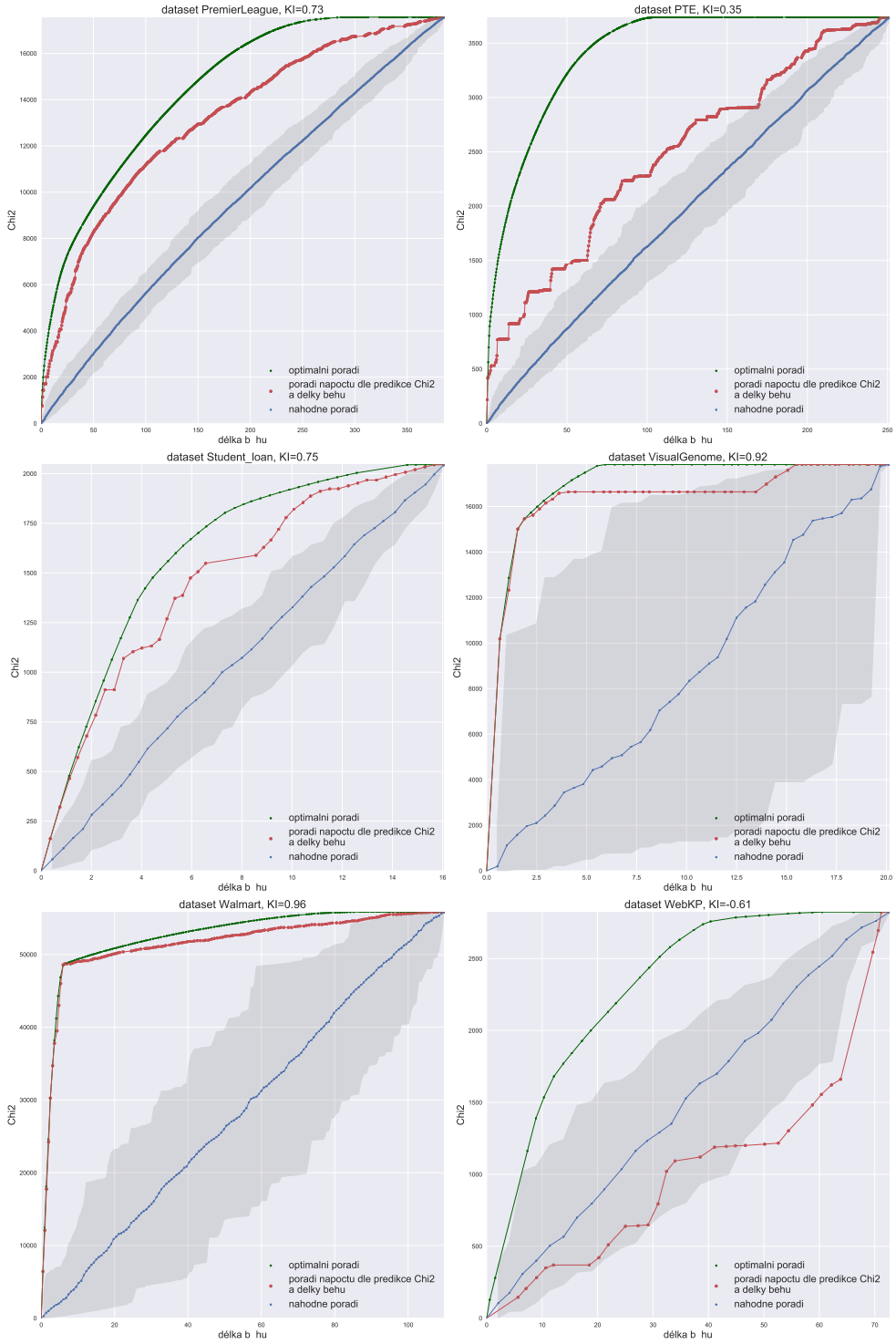
## B.4 Kombinace Chi2 a délky běhu



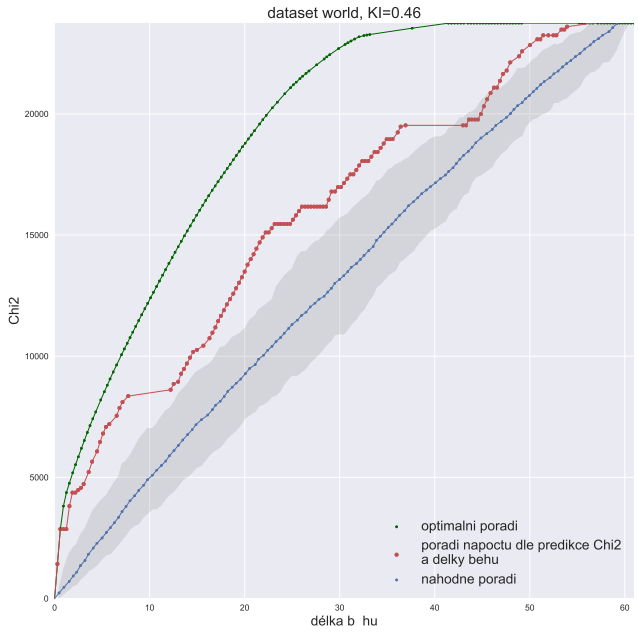

Obrázek B.17: Měření meta-learningového modelu za použití predikce míry Chi2 a délky běhu




Obrázek B.18: Měření meta-learningového modelu za použití predikce míry Chi2 a délky běhu

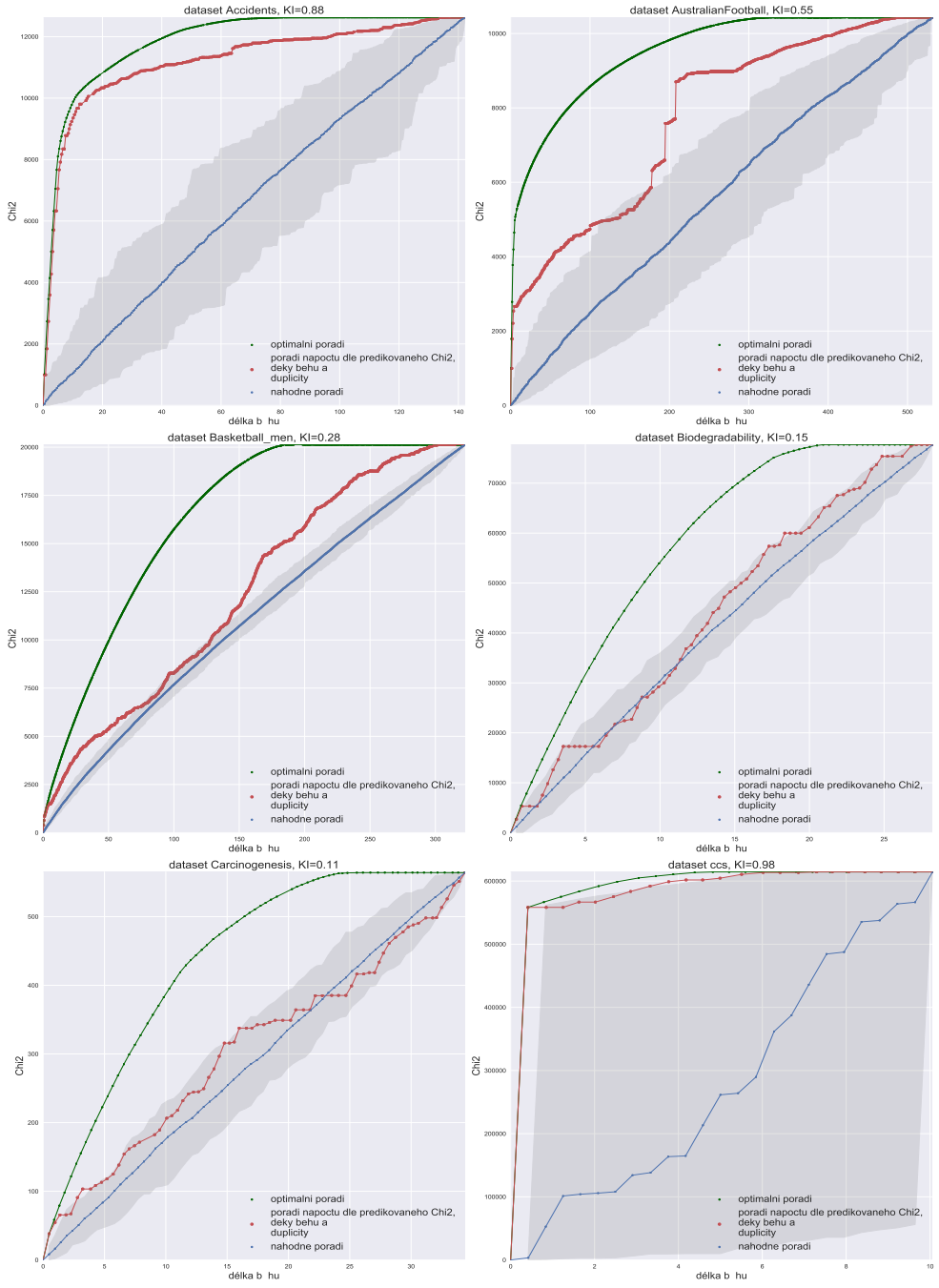



Obrázek B.19: Měření meta-learningového modelu za použití predikce míry Chi2 a délky běhu

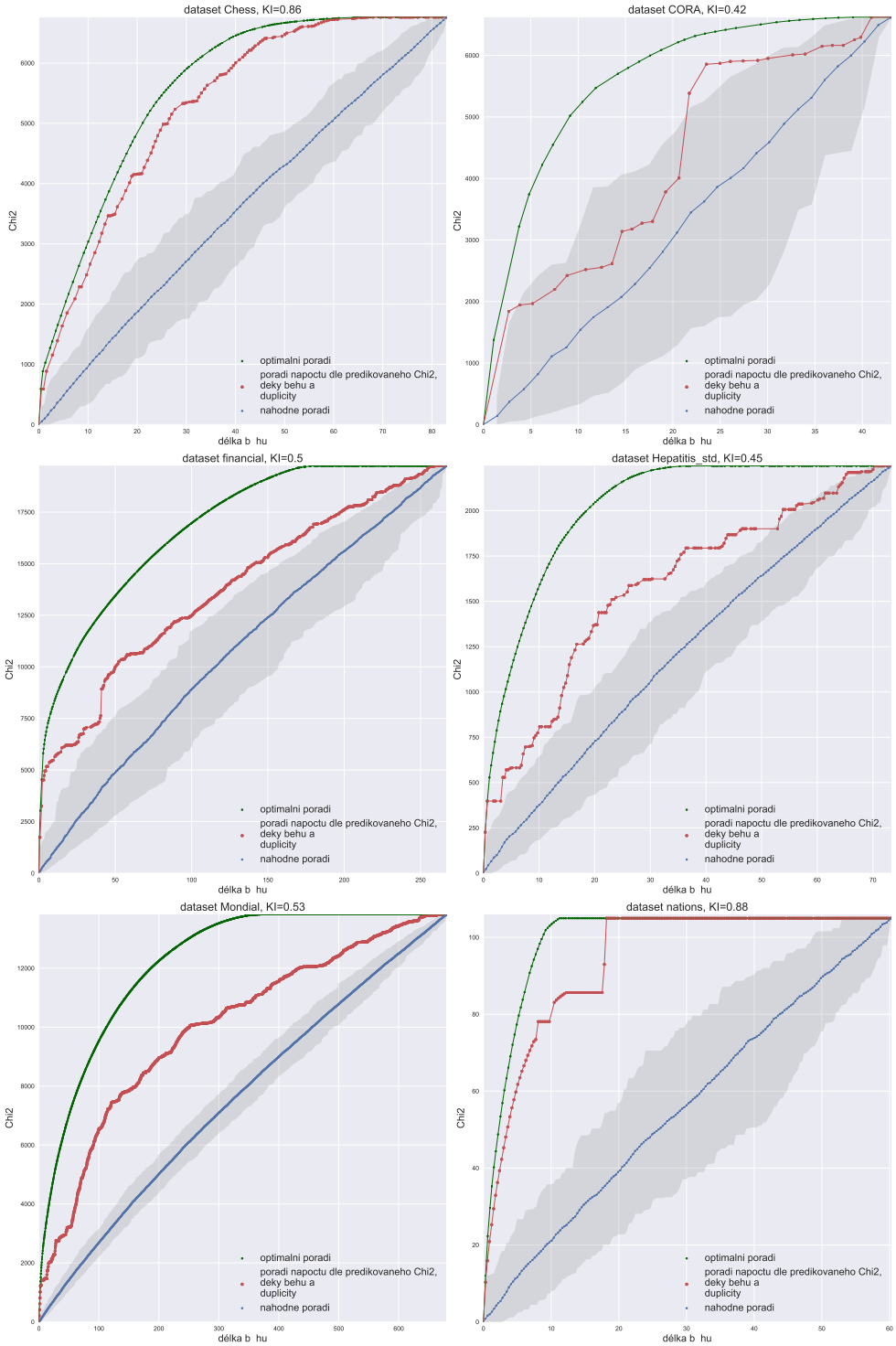



Obrázek B.20: Měření meta-learningového modelu za použití predikce míry Chi2 a délky běhu

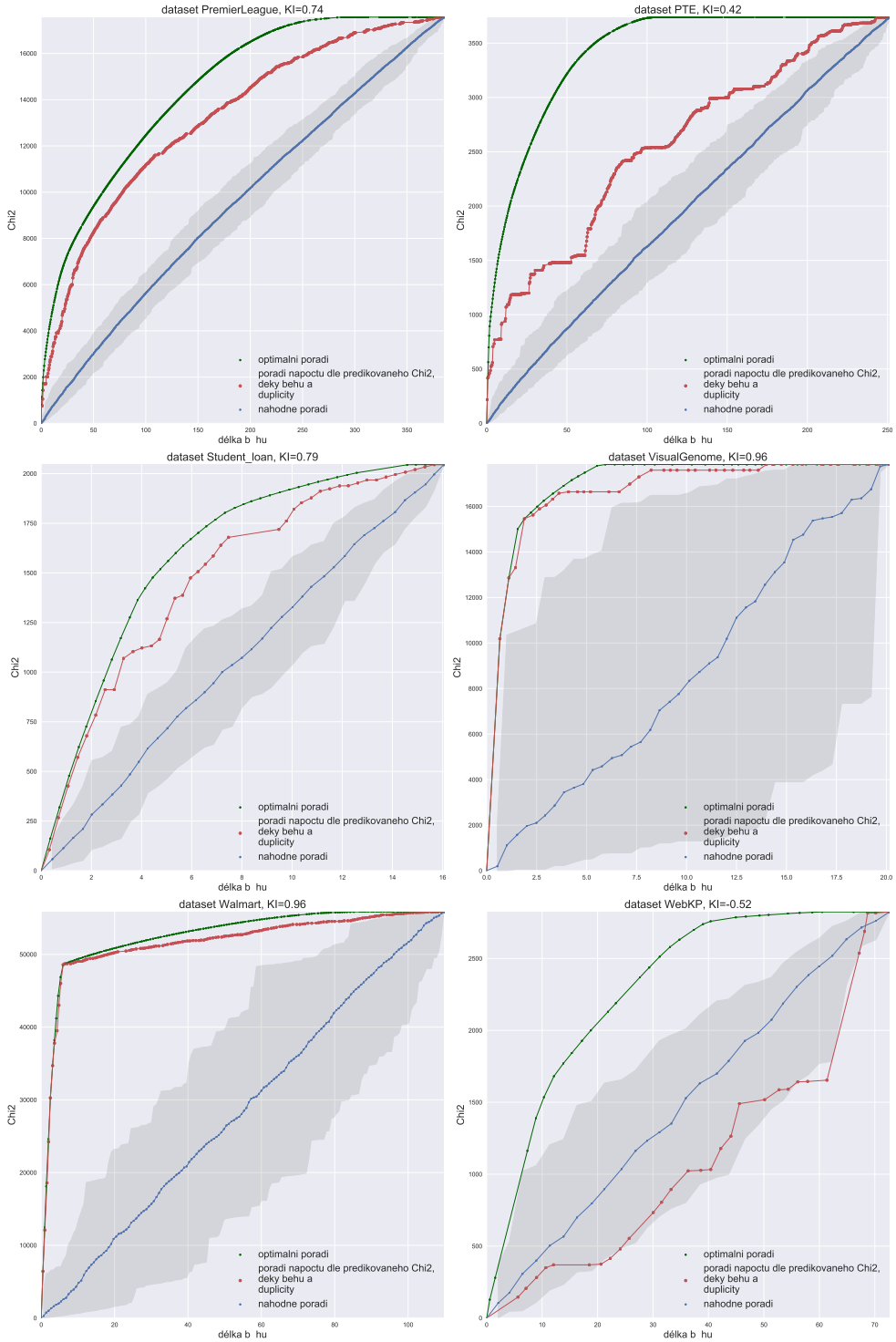
## B.5 Kombinace Chi2, délky běhu a duplicity



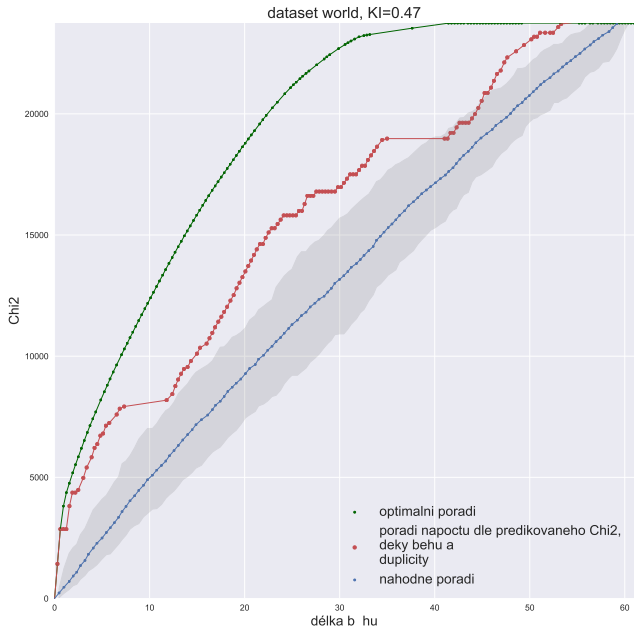

Obrázek B.21: Měření meta-learningového modelu za použití predikce míry Chi2, délky běhu a duplicity




Obrázek B.22: Měření meta-learningového modelu za použití predikce míry Chi2, délky běhu a duplicity




Obrázek B.23: Měření meta-learningového modelu za použití predikce míry Chi2, délky běhu a duplicity




Obrázek B.24: Měření meta-learningového modelu za použití predikce míry Chi2, délky běhu a duplicity

Příloha C

# Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `impl` ..... zdrojové kódy implementace
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF