АНАЛІЗ ІНСТРУМЕНТІВ МАШИННОГО НАВЧАННЯ ДЛЯ АНАЛІЗУ ВЕЛИКИХ МАСИВІВ ДАНИХ

ANALYSIS OF MACHINE LEARNING TOOLS FOR ANALYSIS OF LARGE DATA SETS

Пронін Сергій Вікторович, канд. техн. наук, доцент кафедри комп'ютерних технологій і мехатроніки, Харківський національний автомобільно-дорожній університет. psv59777@gmail.com, sergiy9977@ukr.net, тел.: 050-181-22-74, 057-707-37-43. 61002, Україна, м. Харків, вул. Ярослава Мудрого, 25.

Усиченко Олена Юріївна, кандидат технічних наук, доцент, професор кафедри будівництва та експлуатації доріг, Національний транспортний університет. fbbk@ukr.net, тел. +380442803942, Україна, 01010, м. Київ, вул. М. Омеляновича-Павленка, 1, к. 138.

https://orcid.org/0000-0002-7482-8420

Анотація: Розглядається одна з можливих ідей застосування парадигми великих даних - можливість створення програм для аналізу різноманітної інформації. Для рішення цієї задачі у статті розглядаються різні інструменти машинного навчання які дозволяють створювати кінцеві додатки в сфері аналізу даних. Аналіз включає в себе огляд мов програмування і їх функціональні можливості для створення готових додатків в сфері аналізу даних.
Ключові слова: Машинне навчання, аналіз даних, бібліотека scikit-learn

Введення

Світовий обсяг цифрованої інформації зростає по експоненті. За даними компанії IBS, до 2003 року світ накопичив 5 ексабайт. До 2008 року цей обсяг зріс до 0,18 зеттабайт до 2011 року - до 1,76 зеттабайт, до 2013 року - до 4,4 зеттабайт. У травні 2015 року глобальна кількість даних перевищила 6,5 зеттабайт. На 2020 рік, за прогнозами, людство сформує 40-44 зеттабайт інформації.

За розрахунками IBS, в 2013 році тільки 1,5 % накопичених масивів даних мало інформаційну цінність. У зв'язку з цим активно розвиваються технології обробки великих даних. Вони дозволять структурувати інформацію і отримати з цього користь. Аналіз великих даних дозволить побачити приховані закономірності, непомітні обмеженому людському сприйняттю. Це дає безпрецедентні можливості оптимізації всіх сфер нашого життя: державного управління, медицини, телекомунікації, фінансів, транспорту, виробництва і так далі.

Методи машинного навчання застосовуються в різноманітних областях і допомагають вирішувати безліч завдань: від виявлення спаму і актів шахрайства до розпізнавання і генерації зображень і музичних композицій.

Технологія машинного навчання на основі аналізу даних бере початок в 1950 році, коли почали розробляти перші програми для гри в шашки. За минулі десятиліття загальний принцип не змінився. Зате завдяки бурхливому зростанню обчислювальних потужностей комп'ютерів багаторазово ускладнилися закономірності і прогнози, створювані ними, і розширилося коло проблем і завдань, що вирішуються з використанням машинного навчання [1].

Щоб запустити процес машинного навчання, для початку необхідно зібрати інформацію і сформувати інформаційний масив придатний для алгоритму який буде обробляти запити. Процес навчання триває і після виданих прогнозів, чим більше даних ми проаналізували програмою, тим більш точно вона розпізнає потрібні зображення.

Таким чином основна ідея машинного навчання полягає в тому, щоб навчити комп'ютер "вчитися", тобто виокремлювати з будь-яких даних корисні знання.

У зв'язку з швидким зростанням обсягу інформації виникає гостра необхідність в обробці та структуруванні цієї інформації для її подальшого використання в навчанні моделі, яка на виході буде давати результат. Першим етапом побудови моделі класифікації є збір і попередня обробка даних – цей процес знаходиться на стику таких розділів як: Великі данні(Big data) [2] та інтелектуальний аналіз даних (Data Mining) [3].

Big data або Великі данні - це загальна назва для великих масивів даних і методів їх обробки. Такі дані ефективно обробляються за допомогою масштабованих програмних інструментів, які з'явилися в кінці 2000-х років і стали альтернативою традиційних баз даних і рішень Business Intelligence [3]. Аналіз великих даних проводять для того, щоб отримати нову, раніше невідому інформацію. Подібні відкриття називають інсайтом, що означає осяяння, здогад, раптове розуміння.

Великі дані покликані здійснювати три операції. По-перше, обробляти великі в порівнянні з «стандартними» обсягами даних. По-друге, вміти працювати з швидко динамічними даними в дуже великих обсягах. Тобто даних не просто багато, а їх постійно стає все більше і більше. По-третє, вони повинні вміти працювати зі структурованими і погано структурованими даними паралельно в різних аспектах. Великі дані припускають, що на вхід алгоритми отримують потік не завжди структурованої інформації і що з нього можна витягти більше ніж одну ідею.

В якості визначальних характеристик для великих даних традиційно виділяють «три V»: обсяг (англ. Volume, в сенсі величини фізичного обсягу), швидкість (velocity в сенсах як швидкості приросту, так і необхідності високошвидкісної обробки і отримання результатів), різноманіття (variety, в сенсі можливості одночасної обробки різних типів структурованих і напів структурованих даних) надалі були створені різні варіації і інтерпретації цієї ознаки [2].

Data Mining або інтелектуальний аналіз даних. Data Mining - це процес виявлення в "сирих" даних раніше невідомих нетривіальних практично корисних і доступних інтерпретації знань, необхідних для прийняття рішень в різних сферах людської діяльності. Data Mining є одним з кроків Knowledge Discovery in Databases [3].

Інформація, знайдена в процесі застосування методів Data Mining, повинна бути нетривіальною і раніше невідомої, наприклад, середні продажі є такими. Знання повинні описувати нові зв'язки між властивостями, передбачати значення одних ознак на основі інших. Знайдені знання повинні бути застосовні і на нових даних з деякою мірою вірогідності. Корисність полягає в тому, що ці знання можуть приносити певну вигоду при їх застосуванні. Знання повинні бути в зрозумілій для користувача не математичному вигляді. Наприклад, найпростіше сприймаються людиною логічні конструкції "якщо ... то ...". Більш того, такі правила можуть бути використані в різних СУБД в якості SQL-запитів. У разі, коли витягнуті знання непрозорі для користувача, повинні існувати методи обробки поста, що дозволяють привести їх до інтерпретованих.

Ця стаття розглядає одну з можливих ідей застосування парадигми великих даних - можливість створення програм для аналізу різноманітної інформації Успішне вирішення такого завдання може бути використано, зокрема, в сфері інтернет-комерції — знання про те, який товар краще для покупця, допомагають ефективніше організувати контекстні пропозиції товарів, що в свою чергу веде в збільшення ефективності бізнесу в цілому.

Метою даної роботи буде огляд сучасних інструментів для створення систем аналізу великих масивів даних.

Серед завдань дослідження можна виділити:

- виявлення етапів процесу аналізу даних;
- вибір мови програмування;
- огляд бібліотек для роботи з даними.

Етапи інтелектуального аналізу даних

Для процесу інтелектуального аналізу даних існують цілком певні стандарти процесу, наприклад Cross-Industry Standart Process for Data Mining (CRISP-DM) або SEMMA. У загальному випадку, обидва цих стандарту схожі, за винятком, можливо, іменування етапів роботи [4].

Стандарт передбачає п’ять глобальних фаз процесу аналізу даних (рис.1):

1. Business Understanding - первісна фаза, ставить перед собою розуміння поставленого завдання з точки зору бізнесу. Також, на цій фазі формується проблема аналізу даних, яка буде вирішуватися, ставляться завдання, які будуть виконані в процесі досягнення бізнес-мети.

2. Data Understanding - фаза фокусується на первинному аналізі даних з точки зору проблем їх збору, проблем якості даних. Також в цій фазі робляться початкові припущення про способи аналізу, характер прихованих законів і ін.

3. Data Preparation - фаза покриває процеси отримання початкової вихідної інформації, транформування її в підсумкову вибірку, яка буде подана на вхід моделям аналізу. Проводяться як відбір ознак, так і рішення проблем неякісних даних — відновлення пропусків і позбавлення від викидів в даних.

4. Modeling - фаза описує використання різних методик побудови моделей аналізу, а також процеси настройки параметрів моделей для досягнення оптимального результату. Процес вибору моделі для розв'язання прикладної задачі аналізу користувачів, а також теоретичні положення побудови моделі і її настойки будуть описані далі в поточному розділі.

5. Evaluation - дана фаза фокусується на оцінці результатів моделювання з точки зору аналізу даних. Проводиться перевірка всіх положень і теорій, які використовуються в процесі аналізу, наводяться критерії успішного вирішення бізнес-завдання. Поставлені в фазі Business Understanding мети повинні бути досягнуті.

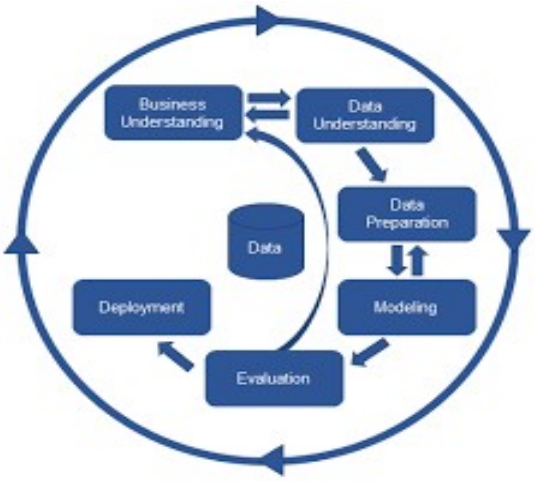

Рисунок 1 – Фази інтелектуального аналізу даних
Figure 1 - Phases of data mining

Задачі які вирішуються за допомогою машинного навчання

Серед основних задач машинного навчання можна виділити наступні:

– задача регресії — прогноз на основі вибірки об'єктів з різними ознаками;

– задача класифікації — отримання категоріальної відповіді на основі набору ознак;

– задача кластеризації — розподіл даних на групи;

– задача зменшення розмірності — зведення великого числа ознак до меншого (зазвичай 2-3) для зручності їх подальшої візуалізації (наприклад, стиснення даних);

– задача виявлення аномалій — відділення аномалій від стандартних випадків. На перший погляд вона збігається із завданням класифікації, але є одна істотна відмінність: аномалії - явище рідкісне, і навчальних прикладів, на яких можна машино навчити модель на виявлення таких об'єктів, або мало, або просто немає, тому методи класифікації тут не працюють. На практиці таким завданням є, наприклад, виявлення шахрайських дій з банківськими картами;

– задача прогнозування — об'єктами є відрізки часових рядів, що обриваються в той момент, коли потрібно зробити прогноз на майбутнє. Для вирішення завдань прогнозування часто вдається пристосувати методи регресії або класифікації, причому в другому випадку мова йде скоріше про завдання прийняття рішень[3].

Вище названі задачі в системах машинного навчання вирішуються за допомогою таких методів як:

– нейронні мережі;

– логістична і пробіт-регресія;

– дерева рішень;

– метод найближчого сусіда;
– метод опорних векторів;
– дискримінантний аналіз;
– байєсівська(наївна) класифікація.

Вибір мови програмування

На сьогодні в сфері інтелектуальних систем найбільш поширеними мовами програмування є Python і Java. Це можна пояснити, з одного боку, особливостями мови Python, що дозволяє вирішувати завдання машинного навчання, а з іншого популярністю і поширеністю мови Java на якому створюється велика кількість проектів. Крім функціоналу для застосування машинного навчання мову Python [5] дає, крім іншого, такі переваги як вільне поширення, широка підтримка і кроссплатформенность, що дає можливість створювати при-розкладання для проектів, написаних іншою мовою. З чого можна зробити висновок, що в якості мови для створення систем аналізу даних краще використовувати Python.

Найбільш поширеними інструментами для роботи з аналізом великих даних на нинішній час-ний день є мови програмування R і Python, а також набір пов'язаних з цими мовами бібліотек машинного навчання і роботи з даними [5].

Мова програмування R - мова програмування, призначена для статистичної обробки даних і роботи з графікою, але в той же час це вільна програмне середовище з відкритим вихідним кодом, що розвивається в рамках проекту GNU. R отримав широке поширення в сферах, де проводиться робота з даними. Основна обчислювальна потужність R полягає в статистичному аналізі, проте він також володіє широким функціоналом для первинного аналізу даних (побудова графіків і таблиць спряженості) і математичного моделювання.

Мова програмування Python - високорівнева універсальна інтерпретована мова сценаріїв. При розробці мовою Python велика увага приділяється простоті і зрозумілості синтаксису, що не тільки скорочує час вивчення його основ, але і підвищує швидкість розробки в цілому [8]. Це далеко не всі переваги даного мови, основні з них:

- об'єктно-орієнтованість;
- вільне поширення і широка підтримка;
- кросплатформенність;
- розвинені функціональні можливості.

Кросплатформеність мови досягається завдяки його реалізації на переносимому ANSI C, що дозволяє програмам, написаним на мові Python, однаково добре компілюватиметься і виконується на будь-яких платформах, де встановлена сумісна версія Python.

Гібридна природа Python об'єднує в собі простоту і зручність мов сценаріїв і потужності компілює мов, що робить Python зручним засобом розробки додатків різного типу. Однак найбільша ефективність мови досягається при вирішенні задач аналізу даних і автоматизації процесів. Python широко використовується в дослідницьких роботах. Мова програмування Python має потужний вбудованим інструментарієм (вбудовані типи об'єктів і динамічна типізація, автоматичне керування пам'яттю) і можливістю використання зовнішніх бібліотек і утиліт сторонніх розробників для вирішення більш вузькоспеціалізованих завдань.

Python використовується не тільки окремими користувачами, але і компаніями, включаючи комерційне використання. наприклад:

- компанія Google широко використовує Python в своїй пошуковій системі і для створення фреймворка App Engine. Також Google розроблена вільно розповсюджувальна бібліотека для машинного навчання - Tensor Flow;

- служба колективного використання відеоматеріалів YouTube в значній мірі реалізована на мові Python. - виробники електронних пристроїв і комп'ютерних компонентів (такі, як Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm і IBM) використовують Python для тестування апаратного забезпечення.

- компанія з виробництва геоінформаційних систем (Environmental Systems Research Institute) використовує Python в якості інструменту налаштування своїх програмних продуктів під потреби кінцевого користувача.

Для вирішення завдання проаналізуємо доступні бібліотеки по роботі з даними на мові Python.

Фреймворк Apache Spark

Так як дані часто мають неструктурований («сирий») вид для подальшої роботи з ними виникає необхідність в їх структуруванні. Це завдання вирішується за допомогою фреймворка Spark [6].

Apache Spark - це аналітичний движок з відкритим вихідним кодом, який використовується для робочих навантажень з великими даними. Він може обробляти як пакети, так і робочі навантаження аналітики і обробки даних в реальному часі. Apache Spark стартував у 2009 році як дослідницький проект в Каліфорнійському університеті в Берклі. Дослідники шукали спосіб прискорити обробку завдань в системах Hadoop. Він заснований на Hadoop MapReduce і розширює модель MapReduce, щоб ефективно використовувати її для більшої кількості типів обчислень, включаючи інтерактивні запити і потокову обробку. Spark надає власні прив'язки для мов програмування Java, Scala, Python і R. Хоча робочими мовами є Scala і Python, згодом в нього була додана значна частина коду на Java. Це пояснюється широким поширенням мови Java і великою кількістю написаних на ньому проектів. Це дає можливість використовувати Spark в проектах написаних на Java.

Spark має такі розширень, як:

- Spark SQL - SQL-запити над даними,
- Spark Streaming -надстройка для обробки потокових даних,
- Spark MLlib -Набір бібліотек машинного навчання;
- GraphX - розподілена обробка графів.

Основним поняттям в Spark'e є RDD (Resilient Distributed Dataset), який представляє собою Dataset, над яким можна робити різні перетворення. За великим рахунком початковий RDD вдає із себе набір «сирих» які потім за допомогою функціоналу Spark перетворюються до потрібного вигляду.

Результатом застосування різних операцій до початкового RDD є новий Dataset. Серед основних операцій можна виділити:

.map (function) - застосовує функцію function до кожного елементу датасета

.filter (function) - повертає всі елементи датасета, на яких функція function повернула справжнє значення

.distinct ([numTasks]) - повертає датасета, який містить унікальні елементи ис-перехідного датасета

Також варто відзначити про операції над множинами, зміст яких зрозумілий з назв:

.union (otherDataset)

.intersection (otherDataset)

.cartesian (otherDataset) - новий датасет містить в собі всілякі пари (A, B), де перший елемент належить вихідного датасета, а другий - датасета-аргументу

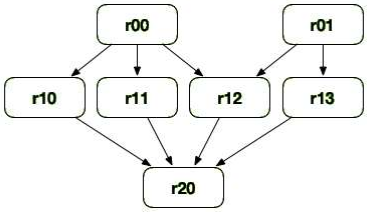

Рисунок 2 – Приклад формування нових датасетів
Figure 2 - Example of new datasets forming

Библиотека scikit-learn

На даний момент існує ряд мов програмування, за допомогою яких можна розроблювати додатки для машинного навчання та аналізу даних.

Завдяки широкому поширенню Python зібрав навколо себе активна спільнота розробників, які в рамках різних проектів розробляють модулі для вузькоспеціалізованих завдань [7].

Одна з основних причин, чому Python використовується для машинного навчання полягає в тому, що у нього є безліч фреймворків, які спрощують процес написання коду і скорочують час на розробку. Обговорімо, які саме бібліотеки і фреймворки Python використовуються в машинному навчанні. У наукових розрахунках використовується Numpy, SciPy, в добуванні і аналізі даних - SciKit-Learn. Ці бібліотеки часто використовують разом з TensorFlow, CNTK і Apache Spark за допомогою яких проектуються нейронні мережі. Крім того простий синтаксис мови Python допомагає розробнику тестувати складні алгоритми з мінімальною витратою часу на їх реалізацію.

Одним з таких проектів стала бібліотека scikit-learn.

Бібліотека scikit-learn надає реалізацію ряду алгоритмів як для навчання з учителем (Supervised learning), так і для навчання без (Unsuoervised learning).

Scikit-learn побудована на основі стека SciPy (Scientific Python), який включає в себе:

- NumPy додає підтримку великих багатовимірних масивів і матриць, а також бібліотеку високоруровневих математичних функцій для операцій з ними.

- SciPy - відкрита бібліотека високоякісних наукових інструментів для мови программирования Python.

- Matplotlib - бібліотека для візуалізації двовимірної і тривимірної графіки.

- IPython - інтерактивна оболонка для мови програмування Python, яка надає рас-ширення интроспекцию, додатковий командний синтаксис, підсвічування коду і автоматичне доповнення Sympy - бібліотека для роботи з символьними обчисленнями

- Pandas реалізує різні структури даних і аналіз.

Бібліотека scikit-learn складається з 35 модулів. Кожен модуль складається з класів і функцій і вирішує такі завдання, як:

- кластеризація (Clustering) - угруповання нерозмічену даних.

- перехресна перевірка (Cross Validation) - оцінка ефективності роботи моделі на незалежних даних.

- набори даних (Datasets) - для зберігання тестових наборів даних і для генерації наборів даних з певними властивостями для дослідження поведінкових властивостей моделі.

- скорочення розмірності (Dimensionality Reduction) - набір алгоритмів для зменшення кількості атрибутів для візуалізації та відбору ознак (Feature Selection), наприклад, метод головних компо-нент (Principal Component Analysis).

- алгоритмічні композиції (Ensemble Methods) - набір методів для комбінування прогнозів декількох моделей.

- витяг ознак (Feature Extraction) - процес визначення атрибутів в даних.

- відбір ознак (Feature Selection) - набір алгоритмів для виявлення значущих атрибутів на основі яких буде побудована модель.

- оптимізація параметрів алгоритму (Parameter Tuning) - методи для отримання максимально еф-ність віддачі від моделі.

- множинне навчання (Manifold Learning) - підхід нелінійного скорочення розмірності даних.

Окремо слід виділити методи реалізують навчання з учителем (Supervised Models). Даний набір методів включає в себе:

- узагальнені лінійні моделі (Generalized Linear Models);

методи дискримінантного аналізу (Discriminate Analysis);

- наївний байесовский класифікатор (Naive Bayes);

- нейронні мережі (Neural Networks);

метод опорних векторів (Support Vector Machines):

- дерева прийняття рішень (Decision Trees).

Висновок

Розглянуто підхід щодо створення систем машинного навчання та аналізу даних. Так як дані часто представлені в неструктурованому "сиром" вигляді необхідно використовувати спеціальні інструменти для їх структурування. Це завдання можна вирішити застосовуючи фреймворки Spark. Також даний фреймворк можливо застосовувати на етапах первинному аналізі даних, початкових припущеннях про способи аналізу, отриманні початкової вихідної інформації, транформування її в підсумкову вибірку, яка буде подана на вхід моделям аналізу. Також за допомогою фреймворку Spark проводяться відбір ознак і рішення проблем неякісних даних (відновлення пропусків і позбавлення від викидів в даних). Для більш детального аналізу краще використовувати библиотеку scikit-learn. Використання цієї бібліотеці даст можливість побудови моделей аналізу, настройки їх параметрів для досягнення мети, та оцінці результатів.

Література

1. Weka: Data Mining URL: https://www.cs.waikato.ac.nz/~ml/weka/ (дата звернення: 23.11.2020).
2. Big Data URL: https://www.it.ua/ru/knowledge-base/technology-innovation/big-data-bolshie-dannye (дата звернення: 23.11.2020).
3. А. А. Барсегян, Методы и модели анализа данных / А. А. Барсегян, М. С. Куприянов – С.-Петербург: БХВ Петербург – 2004 г. – 134 с.

4. Shearer C. The CRISP-DM model: the new blueprint for data mining. – JData Warehousing. – 2000. – C.22.
5. Documentation Python.org. URL: . -https://www.python.org/
6. Spark Application Overview URL: https://docs.cloudera.com/documentation/enterprise/5-6-x/topics/cdh_ig_spark_apps.html (дата звернення: 12.11.2020).
7. Scikit-learn. Machine Learning in Python. URL: . http://scikit-learn.org (дата звернення: 12.11.2020).

ANALYSIS OF MACHINE LEARNING TOOLS FOR ANALYSIS OF LARGE DATA SETS

Pronin Sergey Viktorovich, Cand. Eng. Sci. (Ph.D), Associate Professor of Computer Technology and Mechatronics Kharkiv National Automobile and Road University, Kharkiv National Automobile and Road University, psv59777@gmail.com, sergiy9977@ukr.net, тел.: 050-181-22-74, 057-707-37-43.61002, Ukraine, Kharkiv, st. Yaroslav the Wise, 25
Usychenko O., PhD, Candidate of Technical Sciences, Associate Professor, Professor, Department of Transportation Construction and Property Management, National Transport University, fbbk@ukr.net, тел. +380442803942, Ukraine, 01010, Kyiv, M. Omelianovycha-Pavlenka Str., 1, office 138, https://orcid.org/0000-0002-7482-8420

Abstract. One of the possible ideas of applying the big data paradigm is considered - the possibility of creating programs for the analysis of various information. A successful solution to this problem can be used, in particular, in the field of e-commerce - knowing which product is best for the buyer, helps to better organize the contextual offers of goods, which, in turn, leads to increased efficiency of the business as a whole. Problem. Due to the rapid growth of information, there is an urgent need to process and structure this information for further use in teaching a model that will yield results. The goal of this work will be a review of modern tools for creating systems for analyzing large data sets. The methodology of selection of tools for creation of data analysis systems is used. As a result, tools were identified to create the ultimate applications in the field of data analysis. The originality of the work lies in the use of specialized machine learning libraries to create data analysis systems. The practical value of the work lies in the possibility of creating data analysis systems built using specialized machine learning libraries
Keywords: Machine learning, data analysis, scikit-learn library

References

1. Weka: Data Mining Retrived from. URL:: https://www.cs.waikato.ac.nz/~ml/weka/ (accessed: 23.11.2020).
2. Big Data Retrived from. URL:: https://www.it.ua/ru/knowledge-base/technology-innovation/big-data-bolshie-dannye (accessed: 23.11.2020).
3. A. A. Barsegyan, Metody` i modeli analiza danny`kh [Data Analysis Methods and Models]. *S.-Peterburg: BKhV Peterburg* – 2004 g. – 134 s.
4. Shearer C. The CRISP-DM model: the new blueprint for data mining. – *JData Warehousing*. – 2000 g. – C.22.
5. Documentation Python.org Retrived from. URL:: https://www.python.org/ (accessed: 23.11.2020)
6. Spark Application Overview Retrived from. URL:: https://docs.cloudera.com/documentation/enterprise/5-6-x/topics/cdh_ig_spark_apps.html
7. Scikit-learn. Machine Learning in Python. Retrived from. URL: http://scikit-learn.org (accessed: 23.11.2020)