Obraz zawierający tekst

Opis wygenerowany automatycznie

**Computer Science**

**Informatyka**

Data Science

**Szymon Wujec**

S20431

**“Data analysis in terms of resource usage based on “Bike Sharing” dataset”**

Master Thesis

Thesis supervisor

Dr. habil. Grzegorz Marcin Wójcik

Warsaw, ------------, 2024

Table of contents

[1 Introduction 5](#_Toc169924478)

[1.1 Thesis structure description 5](#_Toc169924479)

[1.2 Goals of the thesis 5](#_Toc169924480)

[1.3 Main thesis issues 5](#_Toc169924481)

[1.3.1 Bike - Sharing systems 5](#_Toc169924482)

[1.3.2 Analysis 10](#_Toc169924483)

[1.3.3 Machine learning 12](#_Toc169924484)

[1.3.4 Refactoring 15](#_Toc169924485)

[2 Description of analysis. 17](#_Toc169924486)

[2.1 Description introduction 17](#_Toc169924487)

[2.2 Manual analysis description 17](#_Toc169924488)

[2.2.1 Seasonal analysis 17](#_Toc169924489)

[2.2.2 Annual analysis 17](#_Toc169924490)

[2.2.3 Impact of working days and holidays 17](#_Toc169924491)

[2.2.4 Weather influence 18](#_Toc169924492)

[2.2.5 Monthly analysis 18](#_Toc169924493)

[2.2.6 Influence of humidity and wind speed 18](#_Toc169924494)

[2.2.7 Analysis by days of the week 18](#_Toc169924495)

[2.2.8 Comparison of registered and casual users 18](#_Toc169924496)

[2.2.9 The impact of holidays on rentals 18](#_Toc169924497)

[2.3 Automated analysis description 19](#_Toc169924498)

[2.3.1 Data upload 19](#_Toc169924499)

[2.3.2 Data preparation 19](#_Toc169924500)

[2.3.3 Modeling 19](#_Toc169924501)

[2.3.4 Model results 20](#_Toc169924502)

[2.3.5 Visualization of results 20](#_Toc169924503)

[2.3.6 Chart analysis 20](#_Toc169924504)

[2.4 Dataset 21](#_Toc169924505)

[2.5 Tools 23](#_Toc169924506)

[2.5.1 IntelliJ IDEA 23](#_Toc169924507)

[2.5.2 Excel 25](#_Toc169924508)

[2.5.3 Python 26](#_Toc169924509)

[2.5.4 Pandas 27](#_Toc169924510)

[2.5.5 NumPy 28](#_Toc169924511)

[2.5.6 Matplotlib 29](#_Toc169924512)

[2.5.7 Seaborn 30](#_Toc169924513)

[2.5.8 Scikit-learn 31](#_Toc169924514)

[3 Functionality process assumptions 32](#_Toc169924515)

[4 Implementation. 37](#_Toc169924516)

[4.1 Front-end code 37](#_Toc169924517)

[4.1.1 Home page 37](#_Toc169924518)

[4.1.2 Summary page 41](#_Toc169924519)

[4.2 Back-end code 42](#_Toc169924520)

[4.2.1 Initialization classes 42](#_Toc169924521)

[4.2.2 Callable classes 48](#_Toc169924522)

[5 Summary 58](#_Toc169924523)

[5.1 Application outcome 58](#_Toc169924524)

[5.2 Acknowledgments 58](#_Toc169924525)

[6 Bibliography 59](#_Toc169924526)

[7 Attachment 60](#_Toc169924527)

[7.1 Table of Figures 60](#_Toc169924528)

[7.2 Table of Pictures 60](#_Toc169924529)

[7.3 Table of Code 61](#_Toc169924530)

[7.4 What CD contains 62](#_Toc169924531)

# Introduction

## Thesis structure description

Głównym celem tej pracy jest analiza przygotowanego zbioru danych przy użyciu różnych technik lub dostępnych narzędzi aby lepiej zrozumieć czym kierują się docelowi klienci firm wypożyczających rowery miejskie lub jakie warunki sprzyjają lepszemu rozwojowi biznesu z tej branży usługowej. Do napisania takiej pracy skłoniły mnie własne doświadczenia, sam w wolnym czasie bardzo często korzystam z takiego środka komunikacji jakim jest rower. Za każdym razem gdy znajduje się przed miejska stacja rowerów publicznych i zamierzam wypożyczyć rower aby dostać się z punktu „A” do punktu „B” ciekawi mnie w jaki sposób firma dostarczająca rowery do użytku publicznego, rozplanowuje kiedy i gdzie powinna dostarczyć zwiększoną ilość tych pojazdów, a gdzie może sobie pozwolić aby było ich mniej. Aby taki proces oraz biznes funkcjonował poprawnie oraz aby był wydajny, wymagana jest dogłębna analiza danych użytkowania, którą mam zamiar przeprowadzić w tej pracy magisterskiej.

## Goals of the thesis

Głównym celem analizy zbioru danych „Bike sharing dataset” dotyczącym wypożyczania rowerów jest zrozumienie wzorców wypożyczeń, optymalizacji oferty oraz identyfikacji trendów czasowych i zachowan użytkowników. Umożliwia to prognozowanie popytu, segmentacje użytkowników, ocenę wydajności oraz wpływu czynników zewnętrznych, a co najważniejsze opłacalności biznesu.

## Main thesis issues

### Bike - Sharing systems

Na swiecie istnieje wiele programów potocznie zwanych „miejskimi rowerami”. Ta potoczna nazwa ma wiele wspólnego ze stanem rzeczywistym ponieważ znaczaca czesc organizacji swiadczacych usługi z branzy wypożyczalni rowerowych to wlasnie zarządy miejskie opłacane z pieniędzy podatnikow. Oczywiście, na przestrzeni lat gdy ten rynek znacząco zaczal się rozwijać, przez zwiększony ruch uliczny, zanieczyszczenie powietrza, a nawet powiększanie się problemu jakim jest globalne ocieplenie, zaczely pojawiać się na rynku firmy prywatne oferujące takie usługi. Lokalizacja rowerow do wypożyczenia, a także występowanie oraz możliwość skorzystania z programów bike-sharing stosowane jest zwykle w dużych aglomeracjach miejskich. Z początku wystepowaly w scislym centrum takich miast, natomiast aktualnie możemy je zauwazyc w lokalizacjach, gdzie ruch publiczny jest znacznie bardziej nasilony. Do takich miejsc naleza miedzy innymi stacje metra, ponieważ metro to bez watpienia miejski srodek transportu który posiada najwieksza ilość przetransportowanych pasazerow, kolejnymi popularnymi lokalizacjami sa obiekty gdzie dojazd samochodem jest znacznie utrudniony. Wyobraźmy sobie, ze w pewnym miescie aktualnie odbywa się mecz pilki nożnej, drużyn posiadających ogromna liczbe fanow. Oczywistym jest, ze stadion wyprzedałby wszystkie dostępne bilety na długo przed rozpoczęciem meczy, przez zwiększone zainteresowanie tym wydarzeniem. Najwiekszy stadion na swiecie jest w stanie pomiescic 114 000 osob na widowni, gdyby każdy chciał przyjechać własnym samochodem to przestrzen przeznaczona na parking musiałaby zajmować 137 hektarow, czyli prawie siedem razy więcej niż zajmuje sam stadion, przy zalozeniu, ze srednia powierzchnia potrzebna do zaparkowania samochodu osobowego to 12 metrow kwadratowych. W takich sytuacjach oraz miejscach swietnie sprawdzają się stacje „rowerow miejskich”, generujące znacznie mniej miejsca, nie wspominając o emisji spalin.



**Picture 1 Bike-sharing station**

Systemy Bike – Sharing mozemy podzielić na pięć generacji tych programów.

#### Generation zero

Stacje, które posiadaja obsluge przy wypożyczeniu roweru naleza do generacji zerowej Bike - Sharing systems. Charakteryzuje je to, ze takie lokalizacji lub stacje nie sa zautomatyzowane lecz prowadzone przez pracowników lub wolontariuszy. Jest to bez watpienia najstarszy model wypożyczalni dostępny na swiecie, aktualnie najbardziej rozpoznawane wypożyczalnie korzystające z tego modelu to siec sklepow z wyposażeniem o charakterystyce sportowej „Decathlon”



**Picture 2 Bike-Sharing generatoin 0**

#### First generation

Pierwszy przypadek bezobsługowej wypożyczalni miał swoje miejsce w Amsterdamie w 1965 roku. Holenderski projektant przemyslowy Luud Schimmelpennink, postanowil wraz ze swoimi znajomymi zebrac 50 rowerow, przemalować je na biało, a następnie rozstawić w mieście do darmowego użytku publicznego. Ten program został nazwany „White Bicycle Plan”. Niestety większość z tych rowerow została skradziona.

**Picture 3 Left side, Luud Schimmelpennink Picture 4 Right side, "White Bicycle Plan"**

#### Second generation

Autorami tego modelu sa Morten Sadolina i Ole Wessung. Opracowali oni model wypożyczalni rowerowej, który zakladal darmowe wypożyczenie pojazdu rowerowego w zamian za kaucje. Depozyt ten był w postaci monet, które następnie odblokowywaly dostep do rowerow, jest to podobny system do tego który znamy z hipermarketów oraz ich wózków sklepowych. Pierwsza pula dostepnych rowerow była zlokalizowana w Danii oraz liczyła 26 jednostek transportowych wraz z 4 stacjami w latach 1991 – 1993. Kolejnym dużym etapem w roku 1995 było wprowadzenie do obiegu 800 jednostek transportowych w Kopenhadze, a ten system nosil nazwe „Bycyklen”.



**Picture 5 Bike-Sharing second generation**

#### Third generation

Generacja trzecia sklada sie ze stacji dokujących gdzie można wypożyczyć pojazd rowerowy, a następnie odstawić go w dowolnej stacji należącej do tego samego Bike-Sharing system. Stacje sa wyposażone w stojaki, które posiadaja mechanizmy zwalniające oraz blokujące pozostawione tam rowery tylko i wyłącznie systemem komputerowym. Wypożyczenie roweru polega na identyfikacji zarejestrowanej osoby karta programu członkowskiego. Taki model wypożyczania został opracowany przez Hellmut Slachta i Paul Brandstätter w latach 1990 – 1992 jako „Public Velo”, natomiast po raz pierwszy został wdrożony pod nazwa „Bikeabout” w roku 1996 przez Uniwersytet Portsmouth i Radę Miasta Portsmouth w Anglii.



**Picture 6 Bike-Sharing Third generation**

#### Fourth generation

Generacja o ktorej mowa posiada swoje dwa warianty. Pierwszy z nich to w istocie rowery gotowe do wypożyczenia generacji czwartej czyli takiej gdzie rowery posiadaja swoje stacje dokujące tak jak w generacji drugiej lub trzeciej, natomiast pojazdy rowerowe sa wyposażone w blokady które umozliwiaja samodzielne podjecie decyzji przez użytkownika czy pozostawić rower w dedykowanej stacji dokującej lub zaparkowanie w dowolnym miejscu przez skorzystanie z wyposażonej blokady. Natomiast wspomniany wcześniej, drugi wariant generacji czwartej to rowery objęte programami typu bike sharing noszące miano generacji piatej. Roznica polega w tym, ze wariant nazwany generacja piata nie posiada dedykowanych stacji dokujących lecz posiada tylko i wyłącznie blokady, znane już z pierwszego wariantu generacji czwartej. Cala generacja czwarta została opracowana przez niemieckie przedsiębiorstwo kolejowe i logistyczne, „Deutsche Bahn” w 1998 roku w celu wykorzystania automatycznie wygenerowanych cyfrowych kodow uwierzytelniających służących do zautomatyzowanego blokowania i odblokowywania rowerow. Ich autorski system w roku 2000 nosil nazwe „Call a Bike”, polegal on na odblokowaniu roweru przy uzyciu wiadomości SMS lub polaczenia telefonicznego, aby w późniejszych latach zrestrukturyzować oraz uruchomić w pełni sprawna aplikacje do zainstalowania na telefon. W Polsce pierwsze programy posiadające w swojej flocie rowery pochodzące z generacji czwartej, pojawily się w roku 2015 w Krakowie pod nazwa „Wavelo” oraz w roku 2017 w Warszawie o nazwie „Acro-bike”



**Picture 7 Bike-Sharing Fourth generation**

(Bicycle-sharing system, 2024)

### Analysis

Analizą mozemy nazwac proces kiedy rozbijamy zlozony podmiot na czynniki pierwsze lub mniejsze części aby docelowo wyciagnac konkretne wnioski oraz uzyskac lepsze zrozumienie omawianej problematyki. Analiza to slowo pochodzące z okolo 1500 do 300 roku przed narodzeniem Chrystusa z jezyka Ancient Greek, ktore brzmialo „analusis”, a oznaczalo „breaking up” lub „an untying”. Przez ten zakres na osi czasu mogliśmy zaobserwować podzial na cztery okresy; „Mycenaean” Greek który trwal w przedziale 1500–1200 rokiem przed Chrystusem, „Dark Ages” którego czas przypadal na lata 1200–800 przed Chrystusem, the „Archaic or Epic period” obowiazujacy w latach 800–500 przed Chrystusem, aby finalnie zakonczyc okresem the Classical period który trwal najkrócej, rozpoczynając od  500 do 300 roku przed narodzeniem Chrystusa. Technika ta oficjalnie i pod swoja nazwa była wykorzystywana do nauki matematyki, rozwiazywania problemów logicznych, natomiast proces ten był wykorzystywany znacznie wcześniej lecz pod bliżej nieokreślona forma przez jakiekolwiek myślące istoty żyjące we wszechświecie. Przykladem jest polowanie lub obrona przed zostaniem upolowanym, proces wyboru odpowiedniego miejsca w którym dany osobnik powinien się znaleźć aby upolować ofiare, lub wybor bezpiecznego i ustronnego miejsca to również proces strategicznej analizy. Natomiast w literaturze i sztuce bardzo czesto pojawia sie pojecie „deconstructive analysis” lub „critical analysis”. Stwierdzenia te polegaja na poszukiwaniu znaczen docelowo ukrytych przez autorow lub artystow. Taki rodzaj analizy przeswietla dzielo sztuki w celu zrozumienia glebszego sensu lub znaczenia ukrytego pod oslona przenosni, cenzury, lub wielu innych podobnych sposobow maskowania glebszego znaczenia tworczosci artysty. Istnieje wiele rodzaji lub odmian analizy, wszystko zalezy od tego co poddajemy omawianemu procesowi analizy.

(Analysis, 2024)

#### Data Analysis

Termin „Data Analysis”, jak sama nazwa wskazuje odnosi sie do przetrwarzania danych w celu ich lepszego zrozumienia oraz na ich podstawie wyciagniecia celnych wnioskow. Do elementow wykorzystywanych w tym procesie mozemy zaliczyc inspekcje, transformacje, modelowanie, a nawet czyszczenie danych. Przeprowadzanie tego typu analizy pomaga w podejmowaniu decyzji, do podjecia ktorych niezbezne jest wziecie pod uwage wiele kryterii. Aby wykonac taka analize niezbedne jest wykonanie kilku krokow, ktore sa scisle egzekwowane na przyklad przez firmy swiadczace uslugi analityczne. Pierwszym etapem majacym wplyw na efekt koncowy jakim jest wynik przeprowadzonej analizy, to ustalenie wymagan potrzebych do skompletowania wymaganego zakresu danych tworzacych pozniej zbior danych. Przykladowo jezeli uzytkownik koncowy, czyli osoba zlecajaca analize, oczekuje konkretnych wniskow dotyczacych tematyki branzy rowerowej to w zbiorze danych mozemy oczekiwac pojawienie sie danych takich jak modele rowerow, ich wagi, rozmiarow i wiele innych. Natomiast nie powinnismy oczekiwac wystepowania informacji niedotyczacych zlecanej analizy, w tym przypadku moglyby to byc przykladowo dane medyczne, ktore maja malo wspolnego z tematyka rowerowa. Po ustaleniu wymagan dotyczacych jakie informacje powinien zawierac zbior danych, nastepuje kolekcjonowanie danych w oparciu o wymagania ustalone w poprzednim kroku. Dane powinny byc kolekcjonowane z wielu mozliwie dostepnych zrodel, w celu unikniecia z gory zalozonej stronniczosci wyniku analizy danych. Dane moga przybierac rozne formy, od tekstowych, przechodzac przez numeryczne dane, a konczac na danych graficznych. Kolejnych krokiem jest etap nazwany „Data processing”. Data processing polega na ulozeniu zebranych danych wynikajacych z poprzednich etapow, w taki sposob aby umozliwic przeprowadzenie analizy. Przykladowo, jezeli analiza dotyczy danych numerycznych, nalezy umiejscowic dane w wierszach i kolumnach w tabeli, wtedy mowimy o ustrukturyzowanych danych. Strukturyzacja danych zazwyczaj nastepuje w programach do przechowywania danych, stworzonych do celow analitycznych lub statystycznych, miedzy innymi takich jak program formy „Microsoft” z pakietu „Office” o nazwie „Excel”. Czyszczenie danych to kolejny bardzo wazny punkt przygotowywania danych do analizy, poniewaz zebrane dane moga zawierac liczne duplikaty, a nawet bledy, ktore w pozniejszym etapie maga miec ogromny wplyw na wynik analizy, uniemozliwajac dotarcie do slusznych wnioskow z przeprowadzanej analizy. Przykladowo, w momencie zestawienia danych finansowych, potrzebnych do przeprowadzenia analizy portfela klienta zlecajacego analize, moga wystapic duplikaty posiadanych dobr, a w rzeczywistosci istnieja one w pojedynczej ilosci. Taki blad moze skutkowac falszywie zawyzona informacja finansowa o posiadanym majatku, a nastepnie prowadzic do problemow prawnych na przyklad zwiazanych z kwestiami podatkowymi. Finalny etap analizy może nastapic za pomocą modelowania oraz algorytmow które zazwyczaj sa matematycznymi równaniami, które pomagają w lepszym zrozumieniu problematyki oraz stawianej tezy w ramach zebranego zbioru danych. Natomiast przy aktualnej technologii najczęściej korzysta się z programów wykorzystujących uczenie maszynowe aby lepiej przeswietlic zgromadzony zbior danych, a następnie przedstawić celne wnioski na podstawie przeprowadzonej analizy.

(Data analysis, 2024)

### Machine learning

Uczenie maszynowe, czesto nazwyane za pomocą skrótu „ML” to gałąź specjalizacji jaka jest artificial intelligence, która również bardzo często jest nazywana przy uzyciu skrótu „AI”. ML ma za zadanie wykonać proces „myslowy” przy pomocy roznych algorytmow oraz AI, który jest wytrenowany na podstawie dostarczonego zbioru danych w postaci tekstowej, numerycznej lub graficznej. W tradycyjnym programowaniu człowiek pisze wszystkie zasady, według których komputer ma działać. W uczeniu maszynowym komputer sam tworzy te zasady, analizując przykłady. Na początku aby wytrenować model, wymagany będzie zbior danych zawierający dane związane z problemem który chcemy rozwiazac. Przykladem może być model przy pomocy którego chcemy rozpoznawać koty, w takim przypadku należy przygotować zestawienie dużej ilości zdjęć kotow oraz innych zwierzat. Kolejnym etapem jest trenowanie modelu który się uczy na podstawie dostarczonych danych. W trakcie trenowania, model analizuje dane i stara się znaleźć wzorce, które pomagają rozpoznać, co jest przedstawine na zdjęciach. Po procesie trenowania, można przystapic do testowania. Testowanie odbywa się na nowych danych których model wcześniej nie widział, aby sprawdzić jak dobrze odnajduje się w rozpoznawaniu zalozonych obiektow przy uzyciu znalezonych wzorcow. Procesz uczenia można przyrownac do nauki dziecka przez pokazywanie mu przedmiotow takich jak banan, jabłko, pomarańcz, podkreslajac nazwe przedstawianej rzeczy. Dziecko powinno znaleźć konkretne wzorce, takiej jak to, ze banany najczęściej sa żółte oraz podłużne lub jabłka sa czerwone i okrągłe. Problem może się pojawić w momencie jeżeli caly czas będziemy przedstawiać czerwone jabłko, a w momencie testu pokazemy jabłko o kolorze zielonym. Dlatego w procesie kompletowania zbioru danych jest bardzo ważne to aby dane były rozne, jeżeli przedstawiamy zdjęcia kotow, to nie może się tam znajdować tylko ten sam kot. W momencie gdy model jest wytrenowany oraz przetestowany, a wyniki sa zadowalające, model jest gotowy do uzycia w praktyce.

(Oxford, 2019)

#### History of Machine Learning

Osoba którą możemy nazwac inicjatorem sztucznej inteligencji to Alan Turing. W roku 1950 Turing zaproponowal test Turinga jako sposób sprawdzenia maszyny czy potrafi myslec jak człowiek. Test polega na pewnego rodzaju grze nazywanej „grze w imitacje”. W eksperymencie biora udział 3 postaci, człowiek, maszyna oraz sedzia. Sedzia to osoba która inicjuje oraz prowadzi test zadając pytania za pomocą interfejsu którym może być przykładowo czat tekstowy. Interfejs jest niezbędny w tym eksperymancie ponieważ sedzia nie wie która postac z która rozmawia jest człowiekiem, a która maszyna. Po zadaniu pytania oraz otrzymaniu odpowiedzi, sedzia analizuje odpowiedzi, a po wykonanej serii pytan nastepuje wynik testu Turinga, czyli moment kiedy sedzia wskazuje jednoznacznie która postac to człowiek, a która to maszyna. Natomiast jeżeli sedzia nie będzie w stanie jednoznacznie wskazać, maszyna zdaje test Turinga. Celem takiego testu była ocena zdolności maszyn do wykazywania się inteligencja. Turing uwazal, ze jeżeli maszyna potrafi prowadzic rozmowe w sposób nie możliwy do odróżniania od rozmowy z człowiekiem to można uznac, ze maszyna jest inteligentna.

(Alan Turing, 2024)

**Picture 8 Left side, Alan Truing Picture 9 Right side, Working Alan Turing**

W roku 1957 Frank Rosenblatt stworzył perceptron, pierwszy algorytm uczenia maszynowego inspirowany neuronami w ludzkim mózgu. Był to wczesny model sieci neuronowej. Praca Rosenblatta nad perceptronem była przełomowa, ponieważ po raz pierwszy pokazała, że maszyny mogą się uczyć i adaptować na podstawie doświadczenia. Mimo że klasyczny perceptron miał swoje ograniczenia, takie jak niemożność rozwiązania problemów nieliniowych, przykladem jest problem XOR, który jest operacja logiczna zwracajaca wynik prawdziwy tylko wtedy kiedy jedno z dwóch wejść jest prawdziwe, niestety Perceptron Rosenblatta, czyli prosty model sieci neuronowej, działa dobrze tylko w przypadku problemów, które można rozwiązać za pomocą jednej prostej linii. Ponieważ problem XOR nie jest liniowo separowalny, perceptron nie potrafi go poprawnie rozwiązać. Mimo to klasyczny perceptron stał się fundamentem dla rozwoju bardziej zaawansowanych architektur sieci neuronowych. Wkład Franka Rosenblatta w rozwój sztucznej inteligencji jest kluczowy, a jego badania nad perceptronem uformowaly fundamenty pod rozwój nowoczesnych metod uczenia maszynowego, które rewolucjonizują nasze podejście do rozwiązywania złożonych problemów w różnych dziedzinach nauki i technologii.



**Picture 10 Frank Rosenblatt**

Jednym z pierwszych odnotowanych programów komputerowych, które potrafią się uczyc na swoich doświadczeniach to program do gry w warcaby autorstwa Arthur Samuel. Arthur to Amerykanski pionier inżynieryjny z dziedziny technologii komputerowej oraz wykorzystywania AI. Wspomniany program jego autorstwa, opracowal w 1959 roku. Program używał techniki zwanej "self-play”, grając tysiące partii przeciwko sobie. Dzięki temu mógł analizować różne strategie i uczyć się, które ruchy są najlepsze w różnych sytuacjach. Samuel wprowadził algorytmy i funkcje oceny, które pozwalały programowi ocenić, jak dobra jest dana pozycja na planszy. Program używał tych ocen do podejmowania lepszych decyzji podczas gry. Jego koncepcje dotyczące samodzielnego uczenia się komputerów były rewolucyjne i zainspirowały wielu przyszłych badaczy. Arthur Samuel jest często nazywany jednym z ojców uczenia maszynowego. Jego innowacyjne podejście do samouczenia się komputerów i gry w warcaby miało ogromny wpływ na rozwój AI. Dziś jego prace są nadal cytowane i studiowane jako podstawowe teksty w dziedzinie sztucznej inteligencji i uczenia maszynowego.

“As a result of these experiments one can say with some certainty that it is now possible to devise learning schemes which will greatly outperform an average person and that such learning schemes may eventually be economically feasible as applied to real-life problems.”

(Samuel, 1959)



**Picture 11 Arthur Samuel**

Wszystkie wyzej wymienione postaci oraz ich dokonania lub odkrycia prowadza nas do czasow w ktorych zyjemy aktualnie. Na codzien uczenie maszynowe jest szeroko stosowane w roznych dziedzinach, od medycyny po marketing. Co chwile wiadomosci obiega informacja o udostepnieniu do uzytku publicznego nowego modelu powszechnie znanego „Chat GPT” autorstwa firmy „OpenAI”. Kazde z osiagniec przybliza ludzkosc do zaawansowanej sztucznej inteligencji.

Historia uczenia maszynowego to historia stopniowego rozwoju, algorytmow oraz co za tym idzie, technologii. Przez lata gromadzone dane, moga byc analizowane automatycznie oraz prowadzic do nowych wnioskow oraz odkryc naukowych. Od wczesnych prac Alana Turing i Franka Rosenblatt, przez rozwoj sieci neuronowych, az po wspolczesne techniki deep learning, uczenie maszynowe nieustannie sie rozwija i zyskuje na znaczeniu w naszym codziennym zyciu.

(Foote, 2021)

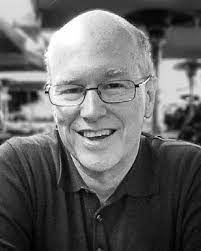
### Refactoring

Refaktoryzacja to technika używana do reorganizacji istniejącego kodu bez zmiany jego funkcjonalności. Programiści często korzystają z tej metody, aby uniknąć powtarzającego się kodu, tworząc uniwersalne metody z użyciem zmiennych. Dzięki temu kod staje się bardziej przejrzysty, czytelny i zwarty, co ułatwia jego dalszy rozwój i utrzymanie. Refaktoryzacja pomaga również w znajdowaniu i eliminowaniu błędów, co prowadzi do bardziej stabilnego i wydajnego oprogramowania. Termin "refaktoryzacja" po raz pierwszy został użyty w 1990 roku przez Williama Opdyke'a i Ralpha Johnsona w ich opublikowanej pracy. Od tego czasu technika ta zyskała ogromną popularność i stała się standardowym narzędziem w arsenale każdego doświadczonego programisty. Martin Fowler, brytyjski deweloper oprogramowania, dzielił się swoją szeroką wiedzą branżową poprzez różne publikacje i książki. Jego prace skupiają się na najlepszych praktykach w programowaniu i projektowaniu oprogramowania. Jedną z godnych uwagi książek jest "Refaktoryzacja: Ulepszanie projektu istniejącego kodu," opublikowana w 2002 roku we współpracy z Williamem Opdyke'iem. Książka ta stała się kluczowym przewodnikiem dla programistów na całym świecie, oferując praktyczne wskazówki i techniki dotyczące refaktoryzacji. Fowler w swoich publikacjach często podkreśla znaczenie refaktoryzacji w kontekście zwinnych metodologii, takich jak Agile, które promują iteracyjny rozwój oprogramowania. Refaktoryzacja jest integralnym elementem tych metodologii, ponieważ umożliwia regularne poprawianie jakości kodu bez konieczności wstrzymywania prac nad nowymi funkcjonalnościami..

(Code refactoring, 2024)

“Help in understanding the code also helps me spot bugs. I admit I'm not terribly good at finding bugs. Some people can read a lump of code and see bugs, I cannot. However, I find that if I refactor code, I work deeply on understanding what the code does, and I put that new understanding right back into the code. By clarifying the structure of the program, I clarify certain assumptions I've made, to the point at which even I can't avoid spotting the bugs. 49 It reminds me of a statement Kent Beck often makes about himself, "I'm not a great programmer; I'm just a good programmer with great habits." Refactoring helps me be much more effective at writing robust code.”

(Flower, Beck, Opdyke, & Brant, 2002)

**Picture 12 Left side, Martin Flower Picture 13 Right side, William Opdyke**

# Description of analysis

## Description introduction

Analiza zbioru danych “Bike sharing” została przeprowadzona w dwóch podejściach. Pierwszym podejściem było przeswietlenie zbioru danych „Bike sharing” manualnie, przy pomocy narzedzi takich jak program „Excel” pochodzący z pakietu „Office” autorstwa firmy „Microsoft”. Następnie za pomocą dostępnych w tym programie formuł analitycznych umożliwiających transformacje, formatowanie, modelowanie danych, prowadzących do ciekawych wyników, zostały wyciągnięte wnioski końcowe. Natomiast drugie podejście to pozostawienie analizy danych zawartych w zbiorze danych „Bike sharing” programowi z dziedziny Machine Learning, który generuje wykresy oraz wypisuje wszystkie interesujące dane wynikowe w zrozumialy dla użytkownika sposób. Te analizy pozwalają lepiej zrozumieć, kiedy i dlaczego rowery są wypożyczane, co może być przydatne dla zarządzających wypożyczalniami rowerów do lepszego planowania i zarządzania zasobami.

## Manual analysis description

Analiza danych z pliku „day.csv” zostanie podzielona na kilka etapów, z krótkim opisem każdej analizy, metodą wizualizacji oraz potencjalnymi wnioskami, jakie można wyciągnąć.

### Seasonal analysis

Na początek porównana zostanie liczba wypożyczeń rowerów w różnych porach roku. Przygotowany wykres słupkowy pokaże, ile razy rowery były wypożyczane wiosną, latem, jesienią i zimą. Będzie można wtedy zauważyć, w których miesiącach wypożyczenia są najczęstsze, co może wynikać z lepszej pogody w pewnych porach roku.

### Annual analysis

Porównane zostaną wypożyczenia w dwóch różnych latach: 2011 i 2012. Wykres słupkowy pokaże całkowitą liczbę wypożyczeń dla każdego roku. Będzie można zobaczyć, czy popularność wypożyczalni rowerów rośnie, czy maleje. Jeśli liczba wypożyczeń w 2012 roku jest większa, można przypuszczać, że usługa zyskuje na popularności.

### Impact of working days and holidays

Zbadane zostanie, jak liczba wypożyczeń różni się między dniami roboczymi a weekendami i świętami. Wykres słupkowy pokaże różnice. Można spodziewać się, że w dni robocze wypożyczeń jest mniej, ponieważ ludzie są w pracy, natomiast w weekendy i święta wypożyczeni prawdopodobnie jest więcej z powodu większej ilości wolnego czasu.

### Weather influence

Porównana zostanie liczba wypożyczeń rowerów przy różnych warunkach pogodowych. Wykres słupkowy pokaże, jak różne typy pogody wpływają na wypożyczenia. Będzie można zauważyć czy w dni słoneczne i bezchmurne jest więcej lub mniej wypożyczeń niż w dni deszczowe, co pozwoli zrozumieć wpływ pogody na zachowania użytkowników.

### Monthly analysis

Zbadane zostanie, jak liczba wypożyczeń zmienia się w poszczególnych miesiącach roku. Wykres liniowy pokaże tę zmianę. Może się okazać, że latem i wiosną wypożyczeń jest więcej niż zimą, co może być związane z lepszą pogodą i wyższymi temperaturami.

### Analysis by days of the week

Przeanalizowane zostanie, jak liczba wypożyczeń zmienia się w zależności od dnia tygodnia. Wykres słupkowy pokaże, ile razy rowery były wypożyczane w każdy dzień tygodnia na przestrzeni dwóch lat. Można będzie zauważyć czy wpływ na ilość wypożyczeń może mieć na przykład fakt dojazdu użytkowników do pracy jeżeli wypozyczen byłoby więcej wciągu pracowniczego tygodnia niż dni weekendowych.

### Comparison of registered and casual users

Porównana zostanie liczba wypożyczeń przez użytkowników zarejestrowanych i okazjonalnych. Wykres słupkowy pokaże różnice między tymi grupami. Można będzie zauważyć czy zarejestrowani użytkownicy wypożyczają rowery częściej, co mogłoby wynikać z niższych kosztów za wynajem lub innych korzyści dla takich użytkowników.

### The impact of holidays on rentals

Pod analizę postawiona zostanie teza, czy liczba wypożyczeń różni się w dni świąteczne. Wykres słupkowy pokaże różnice. Może okazać się, że w święta jest mniej wypożyczeń, bo ludzie spędzają czas z rodziną, albo tych wypożyczeń będzie więcej, bo mają więcej wolnego czasu na rekreację.

## Automated analysis description

Analiza danych z wykorzystaniem Machine Learning to zaprzęgniecie obecnie dostępnej technologii do wykonywania złożonej analizy w stosunkowo krótkim czasie, porównując do analizy wykonanej manualnie przez człowieka. Oczywiście trzeba mieć na względzie poziom trudności oraz zawiłości wykonywanej analizy, ponieważ istnieją sytuacje w których to człowiek dominuje nad maszyną i technologią. Natomiast w przypadku analizowania zbioru danych „Bike sharing”, który zawiera informacje rozciągnięte na przestrzeni dwóch lat, technologia sprawia, ze analiza ta nabiera kolorow, a czasem nawet głębszego znaczenia. Proces takiej zautomatyzowanej analizy przewiduje kilka kluczowych etapow, a każdy etap jest niezbedy do uzyskania finalnego kroku jakim sa wyciagniete wnioski przeprowadzonej analizy.

### Data upload

Dane są wczytywane z pliku o rozszerzeniu „CSV” do programu zapisanego w jezyku Python. Dane dotyczą wynajmu rowerów i zawierają informacje o różnych cechach, takich jak temperatura, wilgotność, dzień tygodnia i tym podobne.

### Data preparation

Dane są przygotowywane do analizy. Skalowane są wartości kolumn, takich jak temperatura, aby były one w odpowiednich jednostkach. Następnie usuwane są kolumny, które nie są potrzebne do analizy, na przykład kolumna z datą.

### Modeling

W tym etapie wykorzystywana jest regresja liniowa. Jest ona stosowana, aby określić, jak różne cechy, takie jak temperatura, wilgotność, dzień tygodnia i tym podobne, wpływają na liczbę wynajętych rowerów. Model jest trenowany na całym zestawie danych, co umozliwia dokładniejsze oszacowanie wpływu poszczególnych cech na liczbe wynajętych rowerow. Co więcej takie podejście przez trenowanie modelu na całym zbiorze danych umozliwia lepsze zrozumienie wszystkich relacji miedzy cechami.

### Model results

Po trenowaniu modelu analizowane są współczynniki regresji. Wartości te wskazują, jak silnie każda cecha wpływa na liczbę wynajętych rowerów. Na przykład wysoki współczynnik dla temperatury oznacza, że temperatura ma duży wpływ na wynajem rowerów. Dotatkowo wyniki prezentowane sa w formie liczb dodatnich, a tazke ujemnych, co pozwala wyznaczyć zależność zmiany wartości cech z ich wpływem progresyjnym lub regresyjnym na liczbe wynajętych rowerow. Przykladowo, wzrost wspolczynnika mowiacego o pogorszeniu się warunków pogodowych prawdopodobnie uzyska ujemny wynik wspolczynnika regresji. Należy te informacje traktować w sposób taki, ze pogorszenie się warunków pogodowych zmniejsza liczbe wynajętych rowerow, a im wieksza wartość bezwzgledna takigo wspolczynnika regresji tym cecha przypisana do tego wspolczynnika, znacznie bardziej wpływa, wręcz dyktuje ilość wynajętych rowerow.

### Visualization of results

Tworzone są wykresy w celu ułatwienia interpretacji wyników. Macierz korelacji pokazuje, jak różne cechy są ze sobą powiązane. „Heatmap” wizualizuje te zależności, co pozwala zobaczyć, które cechy są ze sobą silnie powiązane, a które mniej. Dodatkowe wykresy takie jak słupkowe lub punktowe, najciesciej sa wykorzystywane do wizualizacji korelacji jednej z cech z cecha interpretowana jaka jest liczba wynajętych rowerow.

### Chart analysis

Ostatni etap, dojście do którego nie byłoby możliwe bez wykonania poprzednich, to analiza wygenerowanych wykresow. Wnioski z takiej analizy to wazny moment każdej analizy, ponieważ bledne zrozumienie wartości wynikowych lub popełnienie bledow na etapach przygotowywania lub modelowania danych, może nieść ze sobą poważne konsekwencje. Przykladowo gdyby wynik takiej analizy miał wplywac na restrukturyzacje strategii lub nawet przebranżowienie firmy zlecającej analizę, niosłoby to ze sobą ogromne, zbędnie generowane koszty. W celu zestawieniea wynikow obrazujących szersze spojrzenie na relacje miedzy cechami znajdującymi się w zbiorze danych, dobrze sprawdza się macierz korelacji. Pozwala zobaczyć, jak różne czynniki wpływają na siebie nawzajem. Na przykład można jednocześnie zauważyć, że liczba wynajętych rowerów jest silnie powiązana z temperaturą, a temperatura jest silnie powiązana z porą roku. Natomiast wykresy punktowe Pokazują relację między konkretną cechą, a liczbą wynajętych rowerów. Przy ich pomocy można wykazać czy w cieplejsze dni rowery są częściej wynajmowane.

## Dataset

Zbiór danych dotyczy systemu wypożyczania rowerów „Capital Bikeshare” w Waszyngtonie D.C., USA, i obejmuje dane historyczne z lat 2011-2012. Systemy wypożyczania rowerów umożliwiają automatyczne wypożyczanie i zwracanie rowerów, a dane z tych systemów mogą być używane do analizy mobilności w mieście. Dodatkowo, dane te mogą być pomocne w badaniach nad wpływem różnych czynników środowiskowych na wypożyczanie rowerów.



**Picture 14 Capital Bikeshare rental bicycle**

Dane dotyczace wynajmowanych rowerow firmy „Capital Bikeshare” z lat 2011-2012 znajduja się w pliku korzystającego z rozszerzenia „csv” pod nazwa „day.csv”. Każdy wpis w tym zestawie danych reprezentuje kolejny dzień z zakresu dwuletniego. W efekcie korzystamy ze zbioru danych posiadającego 731 wpisow, a każdy wpis posiada 16 wartosci. Wartosci te zostały nazwane w pierwszym wierszu.

- *instant*: record index  
- *dteday*: date  
- *season*: season (1: spring, 2: summer, 3: fall, 4: winter)  
- *yr*: year (0: 2011, 1: 2012)  
- *mnth*: month (1 to 12)  
- *holiday*: whether the day is a holiday or not (1: yes, 0: no)  
- *weekday*: day of the week (0: Sunday, 1: Monday, ..., 6: Saturday)  
- *workingday*: whether the day is a working day (1: yes, 0: no, meaning it's a weekend or holiday)  
- *weathersit*: weather situation:  
1: Clear, Few clouds, Partly cloudy  
2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds  
3: Light Snow, Light Rain + Thunderstorm + Scattered clouds  
4: Heavy Rain + Ice Pellets + Thunderstorm + Mist, Snow + Fog  
- *temp*: normalized temperature in Celsius (values divided by 41)  
- *atemp*: normalized feeling temperature in Celsius (values divided by 50)  
- *hum*: normalized humidity (values divided by 100)  
- *windspeed*: normalized wind speed (values divided by 67)  
- *casual*: count of casual users  
- *registered*: count of registered users  
- *cnt*: count of total rental bikes including both casual and registered users

**Figure 1 List of attributes in dataset “Bike sharing”**

Autorem zbioru danych jest Hadi Fanaee-T, natomiast właścicielem jest “Laboratory of Artificial Intelligence and Decision Support (LIAAD), University of Porto, INESC Porto”. Jest to jednostka badawcza powiązana z Uniwersytetem w Porto, jednym z największych i najbardziej prestiżowych uniwersytetów w Portugalii. „LIAAD” jest znane ze swoich badań w dziedzinie sztucznej inteligencji oraz systemów wspomagania decyzji. Systemy wspomagania decyzji to miedzy innymi analiza decyzji i zarządzanie ryzykiem, inteligencja biznesowa, techniki optymalizacyjne oraz symulacja i modelowanie. Celem „LIAAD” jest dążenie do zaawansowania społecznej wiedzy oraz technologii w dziedzinach sztucznej inteligencji oraz systemow wspomagiania decyzji.

(LIAAD, 2024)



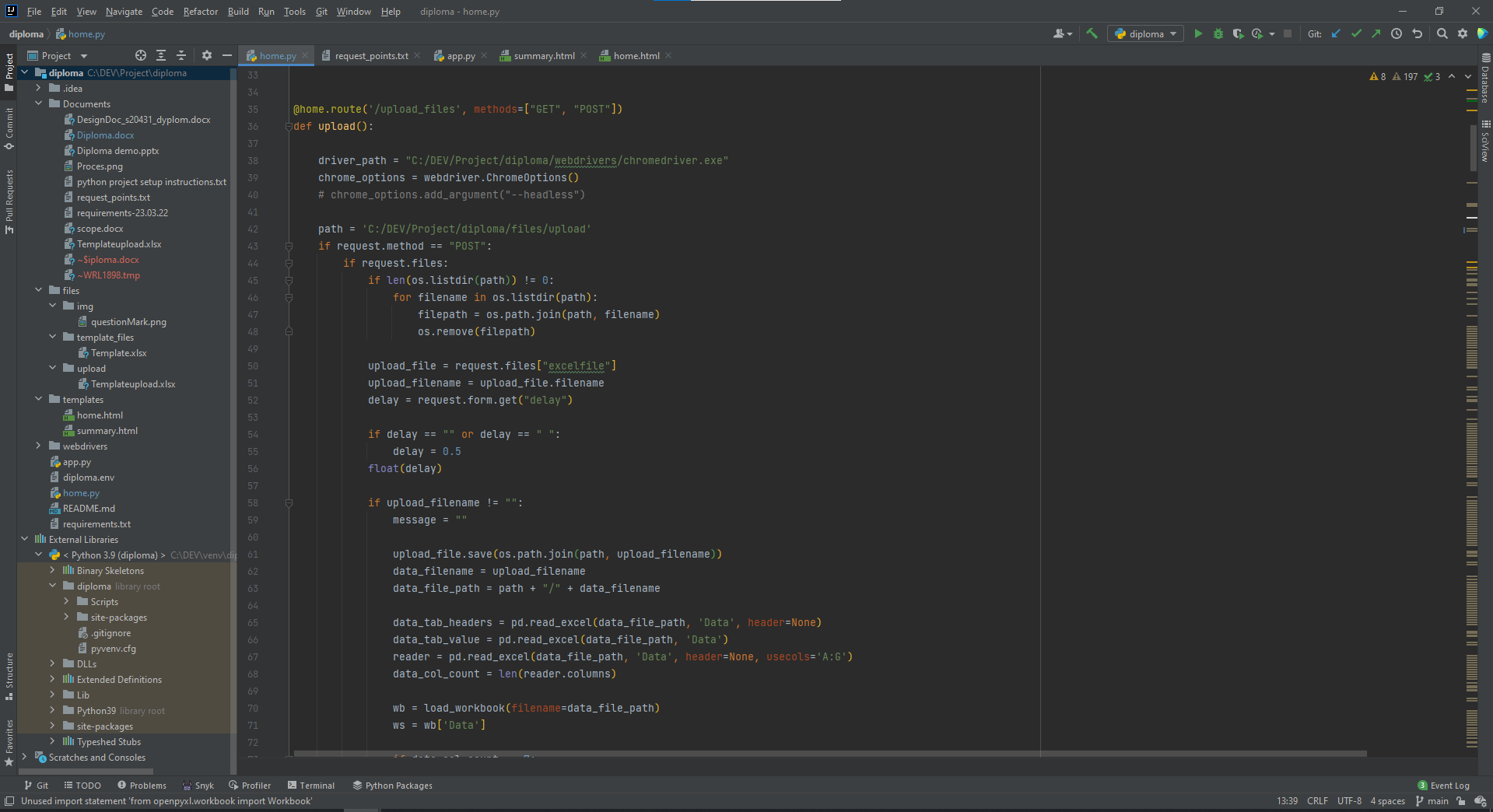
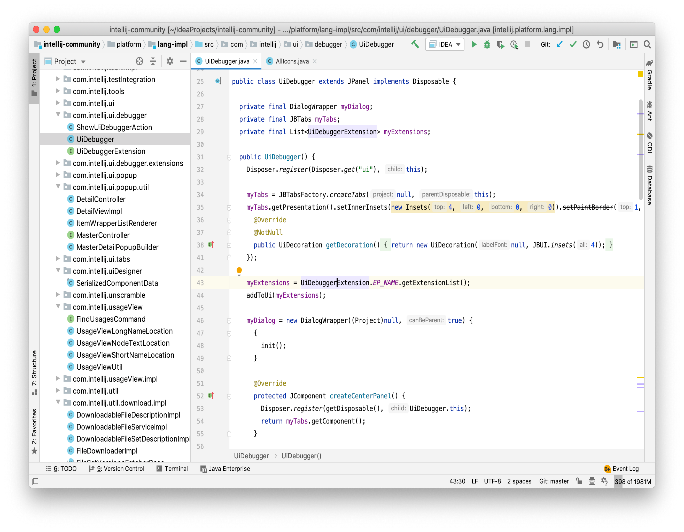
**Picture 15 Hadi Fanaee-T**

Omawiany zbior danych posiada wytyczne aby w razie wykorzystywania go w publikacjach musi zostać cytowane zgodnie z poniżej prezentowana publikacją.  
„Fanaee-T, Hadi, and Gama, Joao, "Event labeling combining ensemble detectors and background knowledge", Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg, doi:10.1007/s13748-013-0040-3.”

## Tools

### IntelliJ IDEA

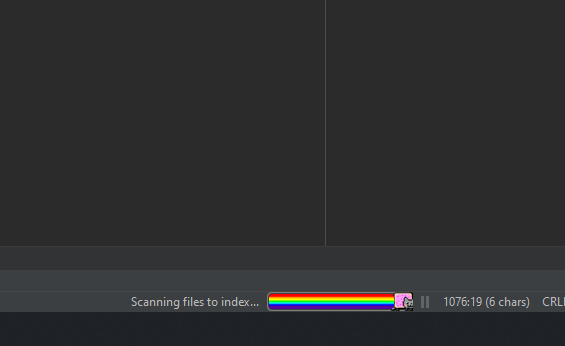
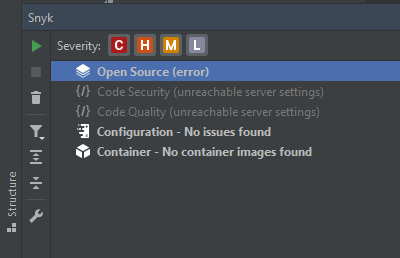
Aby stworzyć zaawansowane lub proste programy IT, które są potrzebne programiście lub organizacji, konieczny jest edytor kodu. Jednym z najlepszych edytorów jest IntelliJ IDEA, komercyjne środowisko programistyczne, które może kompilować napisany kod. Ten edytor został stworzony przez firmę "JetBrains" w języku Java. Pierwsza wersja została wydana na początku 2001 roku i zawierała narzędzia do refaktoryzacji kodu. Teraz IntelliJ IDEA oferuje wiele przydatnych wskazówek i skrótów klawiszowych, które ułatwiają pracę programisty. IntelliJ IDEA obsługuje 19 języków programowania i automatyzacji, takich jak "Python", "Java", "Scala" oraz języki do tworzenia stron internetowych, jak "HTML" i "CSS". Dzięki temu programiści mogą pracować w różnych językach w jednym, spójnym środowisku. IntelliJ IDEA współpracuje również z wieloma innymi narzędziami i środowiskami open source, takimi jak "GIT", "SVN", "CVS", "Apache Maven", "Apache Ant" i "JUnit", co znacznie ułatwia zarządzanie projektami i integrację z innymi systemami. Dla wielu fanów IT interesujący jest także ciemny motyw interfejsu tego edytora, który zmniejsza zmęczenie oczu podczas długotrwałej pracy. IntelliJ IDEA jest nie tylko narzędziem do pisania kodu, ale także wsparciem dla programisty dzięki funkcjom takim jak inteligentne podpowiedzi, automatyczne uzupełnianie kodu, analiza błędów w czasie rzeczywistym i wszechstronne opcje debugowania. To wszystko sprawia, że jest to narzędzie cenione zarówno przez początkujących, jak i doświadczonych programistów. Edytor ten jest również regularnie aktualizowany, co zapewnia dostęp do najnowszych technologii i najlepszych praktyk w programowaniu. Dzięki aktywnej społeczności użytkowników i szerokiej bazie wiedzy dostępnej online, wsparcie i rozwój umiejętności są na wyciągnięcie ręki. IntelliJ IDEA to inwestycja, która przynosi korzyści w postaci zwiększonej produktywności i jakości kodu.

**Picture 16 Left side, IntelliJ dark theme Picture 17 Right side, IntelliJ default light theme**

Aby rozszerzyć funkcjonalność podstawowej wersji IntelliJ IDEA, dostępnych jest wiele wtyczek. Przykładem jest "Rainbow Brackets" autorstwa "izhangzhihao", która poprawia czytelność kodu poprzez kolorowanie nakładających się nawiasów okrągłych, kwadratowych i klamrowych. Inna wtyczka, "Snyk Security - Code, Open Source, Container, IaC Configurations", analizuje kod napisany przez programistę i generuje raport na temat bezpieczeństwa aplikacji. Raporty są bardzo czytelne, nawet dla początkujących programistów, co ułatwia zrozumienie potencjalnych problemów. "Snyk" oferuje również możliwe rozwiązania dla wykrytych luk bezpieczeństwa. Istnieją również wtyczki rekreacyjne, które mają na celu poprawić humor użytkownikom IntelliJ IDEA. Jedną z nich jest "Nyan Progress Bar" stworzona przez "Dimitry Batkovich", która zamienia pasek ładowania kompilatora na tęczowy ogon kultowego bohatera społeczności programistycznej, jakim jest "Nyan Cat". Dzięki takim dodatkom, praca z edytorem staje się nie tylko efektywna, ale również przyjemna. Rozbudowa możliwości IntelliJ IDEA za pomocą takich wtyczek sprawia, że edytor ten jest jeszcze bardziej wszechstronny i dostosowany do indywidualnych potrzeb użytkowników. Niezależnie od tego, czy chodzi o poprawę czytelności kodu, zwiększenie bezpieczeństwa aplikacji czy wprowadzenie elementu zabawy, dostępne wtyczki mogą znacznie ułatwić i uprzyjemnić codzienną pracę programistów.

(Jetbrains, 2024)

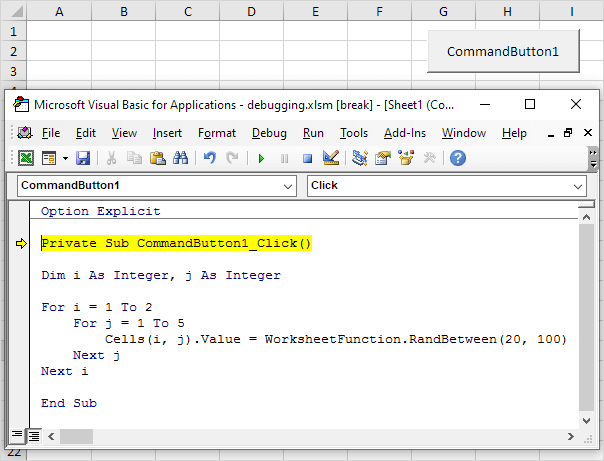
 

**Picture 18 Left side, Nyan Progress Bar Picture 19 Right side, Snyk add-in**

### Excel

Microsoft Excel to popularny arkusz kalkulacyjny dostępny na systemy Windows, macOS, Android i iOS. Umożliwia użytkownikom wykonywanie różnych obliczeń matematycznych oraz prezentowanie danych za pomocą wykresów i tabel przestawnych. Dzięki funkcji Makr, można także automatyzować powtarzalne zadania. Makra tworzy się za pomocą języka Visual Basic for Applications (VBA). Każda nowa wersja Excel wprowadza szereg nowych funkcji i usprawnień, które czynią program bardziej przyjaznym dla użytkownika. Oprócz standardowych funkcji, dostępne są również dodatki, które rozszerzają możliwości programu. Te dodatki, często opracowane przez niezależnych programistów, oferują dodatkowe narzędzia i funkcje, które nie są zawarte w oficjalnej wersji programu. Dzięki tym rozszerzeniom, Microsoft Excel staje się jeszcze bardziej elastycznym i potężnym narzędziem, idealnym do zaawansowanej analizy danych i automatyzacji różnych procesów, co znacząco zwiększa jego wartość zarówno w zastosowaniach biznesowych, jak i codziennych.

(Microsoft, 2024)

**Picture 20 Left side, Excel Worksheet Picture 21 Right side, Visual Basic for Applications**

### Python

Python to język programowania, który oferuje wiele możliwości dzięki licznym narzędziom i bibliotekom. Jego główną zaletą jest przejrzystość i prostota kodu, co sprawia, że jest łatwy do nauki i użycia. Python pozwala na pisanie kodu w sposób jasny i zwięzły. W Pythonie nie trzeba deklarować typów zmiennych, takich jak integer czy string, co ułatwia pisanie kodu. Dzięki swojej popularności, Python działa na wielu systemach operacyjnych. Jego rozwój jest prowadzony przez "Python Software Foundation", organizację non-profit. Standardowa wersja Pythona, "CPython", jest napisana w języku C. Istnieją też inne wersje, takie jak JPython, napisany w Javie, i IronPython dla .NET. Python powstał na początku lat 90., a jego twórcą jest Guido van Rossum. Język został stworzony jako następca języka ABC w Centrum Matematyki i Informatyki w Amsterdamie (CWI - Centrum voor Wiskunde en Informatica). Ciekawostka jest to, że nazwa Python, którą autor nazwal ten jezyk programowania, nie pochodzi od nazwy gatunku węża. Autor wybierając nazwe dla tego jezyka programistycznego był wielkim fanem serialu komediowego emitowanego przez brytyjska telewizje na antenie „BBC” pod tytułem "Monty Python's Flying Circus". Python jest używany w wielu dziedzinach, takich jak analiza danych, sztuczna inteligencja, tworzenie stron internetowych, automatyzacja zadań i tworzenie gier. Frameworki takie jak Django i Flask pomagają w tworzeniu stron internetowych, a biblioteki takie jak NumPy, pandas i TensorFlow są używane w analizie danych i uczeniu maszynowym. Dzięki dużej społeczności i wielu dostępnych narzędziom, Python jest idealnym wyborem zarówno dla początkujących, jak i doświadczonych programistów. Jest to język, który można łatwo dostosować do różnych projektów i zadań, co czyni go bardzo wszechstronnym narzędziem w programowaniu.

(Python Software Foundation, 2024)

**Picture 22 Left Side, CWI building Picture 23 Right side, Guido van Rossum**

### Pandas

Pandas to biblioteka do jezyka programistycznego Python, stworzona do obsługi i analizy różnorodnych danych, zwłaszcza tych numerycznych używanych w zadaniach analizy danych. Została stworzona i wydana na licencji "BSD" w 2008 roku przez Wes McKinney, który wtedy pracował w "AQR Capital Management". Motywacja do stworzenia tej biblioteki wynikała z potrzeby posiadania potężnego narzędzia do efektywnej analizy danych finansowych. Nazwa "Pandas" pochodzi od terminu "panel data", który odnosi się do zbiorów danych zawierających obserwacje z różnych okresów czasowych w ekonometrii.

(Pandas, 2024)



**Picture 24 Wes McKinney**

### NumPy

NumPy to bardzo popularna biblioteka, używana przez programistow piszących swój kod w jezyku programistycznym Python. Biblioteka ta jest stosowana do obliczen numerycznych, jest nieoceniona gdy wymagane sa skomplikowane obliczenia matematyczne operujące na dużych zbiorach danych. Znaczaco szybciej wykonuje obliczenia niż standardowe metody dostępne w surowym jezyku programistycznym Python. Wynika to z tego, ze biblioteka NumPy jest napisana w jezyku C. Cała biblioteka NumPy jest rozwijana jako projekt typu „open-source”, co w praktyce oznacza, ze każdy użytkownik może wnieść swój wkład w rozwój tego potężnego narzędzia programistycznego. Niestety nie jest to jednoznaczne z tym, ze autorski pomysl oraz zapisana fukcjonalosc zostanie dodana do oficjalnej wersji tej biblioteki, ponieważ rozwojem NumPy zarządza grupa glownych deweloperow, którzy rzetelnie oraz starannie podejmują kluczowe decyzje dotyczące kierunku dalszego rozwoju projektu. Od strony technicznej, biblioteka NumPy pozwala na prace z wielowymiarowymi tablicami. Takie tablice noszą nazwę „ndarray”. Po rozwinięciu tej nazwy otrzymamy wyrażenie „n-dimensional array”. Co więcej, korzystając z biblioteki NumPy możliwe jest przeprowadzanie operacji matematycznych na calych tablicach jednocześnie, co znacząco skraca czas wykonywania operacji matematycznych niż w trakcie przeprowadzania ich na pojedynczych elementach znajdujących się w tablicy. Historia tej biblioteki ma swój początek już w latach 90. Natomiast funkcjonowało jako inna biblioteka o nazwie „Numeric”. Dopiero w roku 2005, amerykański naukowiec komputerowy oraz biznesmen, Travis Oliphant postanowił połączyć funkcjonalności biblioteki „Numeric” z biblioteką „Numarray”, aby uzyskać bibliotekę „NumPy”, którą znamy dzisiaj. NumPy to biblioteka, która jest podstawą dla wielu innych bibliotek stworzonych oraz przeznaczonych do analizy danych, a także uczenia maszynowego.

(NumPy, 2024)



**Picture 25 Travis Oliphant**

### Matplotlib

Matplotlib to kolejna biblioteka kojarzona z uczeniem maszynowym, przeznaczona do użytku dla programistów korzystających do pisania swojego kodu w języku programistycznym Python. Biblioteka Matplotlib służy do tworzenia wykresów, a także szeroko pojętej wizualizacji danych. Umożliwia generowanie projekcji graficznych pochodzących z szerokiego zakresu grupy grafik służących do wizualizacji danych. Począwszy od prostych wykresów liniowych, a kończąc na skomplikowanych oraz zaawansowanych diagramach reprezentujących wyznaczony zakres danych. Warto tez dodać, ze Matplotlib umozliwia tworzenie wykresow interaktywnych, co pozwala na dynamiczne modyfikowanie wykresow. Natomiast wykresy w bardzo latwy sposób mogą być dostosowywane względem potrzeb użytkownika w zakresie kolorow, czcionek oraz innych elementow znajdujących się na wygenerowanych projekcjach graficznych. Bardzo często wystepuje w projektach programistycznych w parze z takimi bibliotekami jak „pandas” czy „NumPy”. Popularność tej biblioteki towarzyszy jej właściwie od poczatkow jej istnienia gdy autor, John D. Hunter w roku 2003 postanowil stworzyć zestaw narzedzi dostępnych dla uzytkownikow jezyka programistycznego Python, podobnych funkcjami do narzedzi do tworzenia wykresow dostępnych w srodowisku „MATLAB”. John D. Hunter był neurobiologiem, który potrzebowal narzędzia spełniającego jego wymogi aby móc wizualizować dane w trakcie konstruowania swojej pracy badawczej. Matplotlib, podobnie jak większość bibliotek dostępnych do użytku w jezyku programistycznym Python, należy do projektów „open-source”, gdzie użytkownicy mogą zglaszac bledy, proponować nowe funkcje oraz brac udział w dyskusjach na temat przyszłości projektu jakim jest biblioteka „Matplotlib”. Po śmierci Johna D. Huntera w 2012 roku, projekt jest zarządzany przez grupę głównych deweloperów, którzy kontynuują jego pracę i dbają o rozwój biblioteki.

(Matplotlib, 2024)



**Picture 26 John D. Hunter**

### Seaborn

Seaborn to biblioteka dostepna do uzytku w języku programowania Python, która służy do tworzenia statystycznych wizualizacji danych. Jest zbudowana na podstawie Matplotlib i zapewnia wyższego poziomu interfejs do tworzenia atrakcyjnych i informacyjnych wykresów. Biblioteka Seaborn jest używana do tworzenia wizualizacji, które ułatwiają zrozumienie statystycznych relacji pomiędzy danymi. Seaborn oferuje proste sposoby na generowanie skomplikowanych wykresów, takich jak wykresy korelacji, rozkładu, regresji, wykresy par i wiele innych. W środowisku programistycznym, ta biblioteka znana jest ze swojej łatwości użycia, oferuje ona intuicyjny interfejs, który sprawia, ze tworzenie zaawansowanych i skomplikowanych wykresów jest proste i szybkie. Co więcej, biblioteka bezproblemowo współpracuje z biblioteka „pandas” oraz typowym dla tej biblioteki „Pandas DataFrame”. Jest to kluczowe, ponieważ ta integralność pozwala na latwe tworzenie wykresow bezpośrednio z danych strukturalnych, przeprocesowanych przez narzędzia dostępne w bibliotece „pandas”. Przez to, Seaborn zyskuje na swojej popularności oraz częstotliwości wykorzystania w projektach programistycznych. Z pozostałych funkcji które zasługują na wyróżnienie tej biblioteki to wbudowane narzędzia do wykorzystywania statystyk w wizualizacjach, takich jak wykresy pudełkowe lub mapy cieplne. Autorem tego zbioru narzędzi jest neurobiolog, statystyk Michael Waskom. Stworzył bibliotekę Seaborn w roku 2012 aby ułatwić tworzenie statystycznych wykresow w jezyku programistycznym Python. Po dziś dzień Michael Waskom odgrywa kluczową role w rozwoju projektu „Seaborn”, ale również inni deweloperzy i użytkownicy przyczyniają się do jego rozwoju poprzez zgłaszanie błędów, proponowanie poprawek i nowych funkcji na zasadzie projektu typu „open-source”.

(Seaborn, 2024)



**Picture 27 Michael Waskom**

### Scikit-learn

Scikit-learn, często stylizowane jako „sklearn”, to popularna biblioteka w języku programistycznym Python, używana do uczenia maszynowego. Jest łatwa w użyciu i oferuje szeroki zestaw narzędzi do analizy i modelowania danych. Biblioteka ta jest bardzo często wybierana jako narzędzie do budowania i oceny modeli z zakresu uczenia maszynowego. Scikit-learn wspiera różne techniki powiazane z uczeniem maszynowym, warto natomiast wyróżnić klasyfikację, regresję, klasteryzację, redukcję wymiarowości, a także narzędzia do wstępnego przetwarzania danych. Kompleksowość to zdecydowanie określenie które można użyć omawiając bibliotekę Scikit-learn, ponieważ oferuje ona bardzo szeroki wachlarz algorytmów uczenia maszynowego. Co więcej idealnie współgra ze środowiskiem jezyka programistycznego Python, oraz bibliotek dedykowanych dla tego jezyka, takich jak NumPy, SciPy, Pandas i Matplotlib. Autorem tej biblioteki jest David Cournapeu, powstala ona jako projekt „Google Summer of Code” w roku 2007.

(Scikit-learn, 2024)

# Analysis process assumptions

Pierwszym etapem przeprowadzenia analizy było znalezienie odpowiedniego zbioru danych, który będzie zawieral duza ilość danych oraz będzie dotyczyl tematu na podstawie którego będzie można przeprowadzić analizę prowadzaca do bardzo ciekawych wnioskow. Duzo czasu poswiecilem aby taki zbior danych znaleźć, jednym z glownych aspektow decydującym o moim wyborze było to, w jakim stopniu problematyka w formie danych będzie dla mnie interesujaca. Przejrzalem kilkadziesiąt zbiorow danych dostępnych na portalu „UC Irvine”. Ostatecznie moja uwagę skupil zbior danych przedstawiający informacje o wynajmie rowerow miejskich. Wielokrotnie zastanawiałem się nad tym w jaki sposób funkcjonują takie systemy wynajmu rowerow miejskich, miedzy innymi czy to opłacalny biznes dla firm prowadzących takie usługi z perspektywy zainteresowania korzystaniem z takiego sposobu wynajmu rowerow przez użytkowników. To wlasnie te pytania sklonily mnie do podjęcia decyzji o analizie zbioru danych „Bike Sharing”. Nastepnie uważnie się wczytałem w opisy atrybutow znajdujących się w zbiorze aby podjąć wstępne założenia. Założenia obejmowaly podjecie decyzji, które ze znajdujących się atrybutow w zbiorze danych „Bike Sharing” w momencie zestawienia ze sobą, będą prowadzic do ciekawych wnioskow aby maksymalnie wykorzystać potencjal statystyczny znajdujący się w tym zbiorze danych. Po wyznaczeniu celi jakie chciałbym osiagnac analizując dane, przystapilem do doboru odpowiednich dla konkretnych zadań narzedzi, wspomagających mnie w przeprowadzeniu analizy. Wybor padl na program „Excel” autorstwa firmy „Microsoft” z pakietu „Office”, ponieważ oferuje on wiele formul umożliwiających przetwarzanie danych oraz prezentacje ich w formie graficznej, co więcej program ten obsluguje możliwość odczytu wielu rozszerzen plikow. W moim przypadku zbior danych „Bike Sharing” jest zapisany przy uzyciu rozszerzenia plikow „CSV”, z którym program „Excel” radzi sobie bez problemu. Uznalem, ze poza analiza zestawienia konkretnych atrybutow ze sobą, ciekawym elementem analizy będzie zestawienie informacji, jakie znaczenie maja konkretne atrybuty pod kątem wpływu na ilość wynajętych rowerow na przestrzeni całego zakresu czasu przedstawionego w zbiorze danych „Bike Sharig”. Do wykonania tego zadania postawiłem na trening modelu polegający na uczeniu maszynowym. Aby odpowiednio wytrenować model uczenia maszynowego oraz otrzymać interesujące mnie wyniki, bazując na strukturze danych znajdujących się w zbiorze danych „Bike Sharing”, zdecydowałem się skorzystać z regresji liniowej. Regresja liniowa jest jedną z najprostszych metod modelowania relacji między zmiennymi. Współczynniki regresji bezpośrednio pokazują, jak zmiana jednej jednostki w danej cechy wpływa na wynik. Dzięki temu łatwo jest zrozumieć wpływ poszczególnych cech na liczbę wynajętych rowerów. W wielu przypadkach relacje między zmiennymi a wynikiem mogą być dobrze przybliżone przez funkcję liniową. W kontekście wynajmu rowerów, cechy takie jak temperatura, dzień tygodnia czy wiatr mogą mieć liniowy wpływ na liczbę wynajętych rowerów. Przy uzyciu regresji liniowej można łatwo ocenić istotność poszczególnych cech. Współczynniki regresji oraz ich wartości pozwalają na identyfikację, które zmienne mają największy wpływ na wynik. Regresja liniowa jest dobrym wyborem do trenowania modelu analizy wynajmu rowerów, ponieważ jest prosta, łatwa do interpretacji, efektywna obliczeniowo i dobrze sprawdza się w przypadku dużej liczby obserwacji. Dodatkowo, umożliwia łatwą ocenę istotności cech i daje solidne podstawy do ewentualnego rozszerzenia na bardziej zaawansowane modele, jeśli byłoby to konieczne. Przechodząc dalej, przed rozpoczęciem wyciągania wnioskow, należy oczyscic zbior danych z informacji które nie koniecznie będą przydatne w trakcie analizy lub takie które mogą mieć nawet negatywny wpływ na wyniki końcowe przeprowadzanej analizy. W momencie gdy analizy przeprowadzamy przy uzyciu programu „Excel” to zbędne atrybuty nie stanowa problemu, jedynie wartości puste w nich zawarte, tak zwane „NULLS”, ponieważ w trakcie zapisywania formul statystycznych wystarczy nie uwzgledniac tych atrybutow w obliczeniach które nie maja większego znaczenia lub negatywnie wplywaja na przeprowadzana analizę. Natomiast w momencie trenowania modelu uczenia maszynowego, który przeswietla caly zestaw danych, należy wyciac z obszaru analizowanego zbioru danych atrybuty negatywnie wplywajace na wynik treningu, a w kolejnym etapie analizy. Takimi atrybutami są „instant”, „dteday”, „casual” oraz „registered”. Atrybut „instant”, nie został zakwalifikowany ze względu na to, ze tylko reprezentuje iteracyjny numer wpisu w zbiorze danych „Bike Sharing”. Pozycja „dteday” została wykluczona ponieważ do przeprowadzania analizy nie posiada większego znaczenia, a tylko sluzy do przedstawienia innych atrybutow na jej postawie, przykładowo atrybut „season” reprezentujący porę roku. Podobnie sytuacja wygląda z atrybutami „casual” oraz „registered” ponieważ suma ich wartości to atrybut docelowy w zautomatyzowanej analizie, mowa o „cnt” reprezentującym dokladna ilość wynajętych rowerow konkretnego dnia. Natomiast atrybuty „casual” oraz „registered” posiadają swoje zastosowanie w manualnej analizie.

# Implementation

## Implementation of manual analysis

Proces analizy wykonywanej ręcznie prowadzi przez osiem scenariuszy. Każde z przeprowadzonych zestawień przedstawia inny czynnik wpływający na ilość wypożyczonych rowerów w założonym przez scenariusz, okresie czasowym.

### Seasonal analysis implementation

Analiza sezonowa to zestawienie czterech danych wynikowych. Każda z wartości reprezentuje konkretny sezon spośród wiosny, lata, jesieni i zimy. Dane są wyliczane przy użyciu dostępnej w programie „Excel” formuły statystycznej nazywającej się „SUMIF”. Jak sama nazwa wskazuje, formuła „SUMIF” pozwala sumować konkretne wartości w momencie jeżeli zostanie spełniony konkretny, zdefiniowany oraz zapisany w formule warunek. Struktura całej formuły prezentuje się w następujący sposób, przedstawiony poniżej.

=SUMIF( range; criteria; [sum\_range] )

**Figure 2 SUMIF formula structure**

Ta struktura przedstawia trzy-elementową formułę. Pierwszym elementem jest zakres komórek znajdujących się w konkretnym sheet poddanych do analizy przeprowadzanej przez funkcjonalność formuły „SUMIF”. Drugi element to kryteria jaka powinna zostać spełniona w zdefiniowanym zakresie danych w elemencie pierwszym tej formuły. Ostatni element to zakres komórek które powinny zostać zsumowane gdy zostanie spełniony warunek zdefiniowany na drugiej pozycji formuly, znajdujący się pod określeniem „criteria”.

W Przypadku analizy sezonowej, konkretne wartości znajdujące się w formule „SUMIF” to wartości przedstawione poniżej.

=SUMIF( C2:C732; 2; P2:P732 )

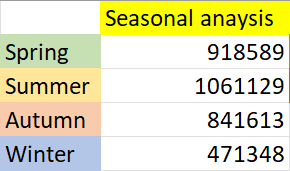
=SUMIF( C2:C732; 3; P2:P732 )

=SUMIF( C2:C732; 4; P2:P732 )

=SUMIF( C2:C732; 1; P2:P732 )

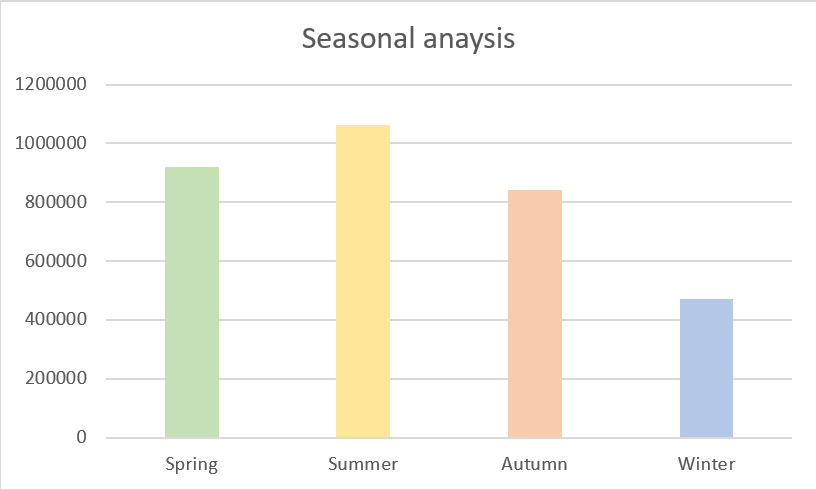
**Figure 3 Seasonal analysis formulas**

Pierwszy element to zakres komórek zapisany przy użyciu cell reference reprezentujących litery kolumn oraz liczby wierszy atrybutu „season”. Element na pozycji drugiej to kryteria warunkujące dobór odpowiednich wartości w kolumnie zawierającej dane dotyczące sezonów. Numer 1 reprezentuje zimę, pod numerem 2 opisane są recordy zebrane wiosną, 3 to lato, a numer 4 przypisany jest do wpisów jesiennych. Ostatni element odnoszący się do sumowanych wartości w momencie spełnienia zadeklarowanego warunku znajdującego się pod elementem drugim formuły „SUMIF” to zakres komórek oznaczonych atrybutem „cnt”, który odpowiada liczbom wynajętych rowerów konkretnego dnia.



**Figure 4 Output values of seasonal analysis**

Wyniki liczbowe przeprowadzonej analizy wskazują sezon letni jako sezon w którym wynajmuje się najwięcej rowerów miejskich na przestrzeni dwóch lat. Natomiast sezonem kiedy wynajmowanych rowerów jest najmniej, jest sezon zimowy.



**Figure 5 Seasonal analysis chart**

Przedstawiony wykres to łatwiejsza w zobrazowaniu dla osoby analizującej forma przedstawienia wyników analizy. Dzięki temu wykresowi słupkowemu, możemy zauważyć wręcz jednostajną tendencje do falowania wyników pomiędzy czterema sezonami na podstawie danych zgromadzonych na przestrzeni dwóch lat. Poza wartościami maksymalnymi oraz minimalnymi należącymi do sezonów letnich oraz zimowych, zauważyć możemy zwiększoną liczbę wynajętych rowerów dla okresy wiosennego w odniesieniu do sezonu jesiennego. Analizując następnie cały wykres, biorąc pod uwagę wszystkie pozycje znajdujące się na osi horyzontalnej, możemy dojść do wniosku, ze wpływ na ilość wynajętych rowerów może mieć inna temperatura towarzysząca każdemu z prezentowanych sezonów. Ponieważ z reguły sezon wiosenny jest sezonem cieplejszym niż sezon jesienny. Natomiast skrajnymi przeciwieństwami temperaturowymi są sezony letnie oraz zimowe, stanowiące pierwsze oraz ostatnie miejsce w rankingu sezonów pod kątem ilości wynajętych rowerów.

### Annual analysis implementation

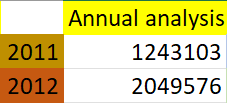
Podobnie jak w poprzednim scenariuszu, do przeprowadzenia analizy została wykorzystana formuła należąca do standardowego oraz podstawowego pakietu formuł w programie „Excel”.

=SUMIF( D2:D732; 0; P2:P732 )

=SUMIF( D2:D732; 1; P2:P732 )

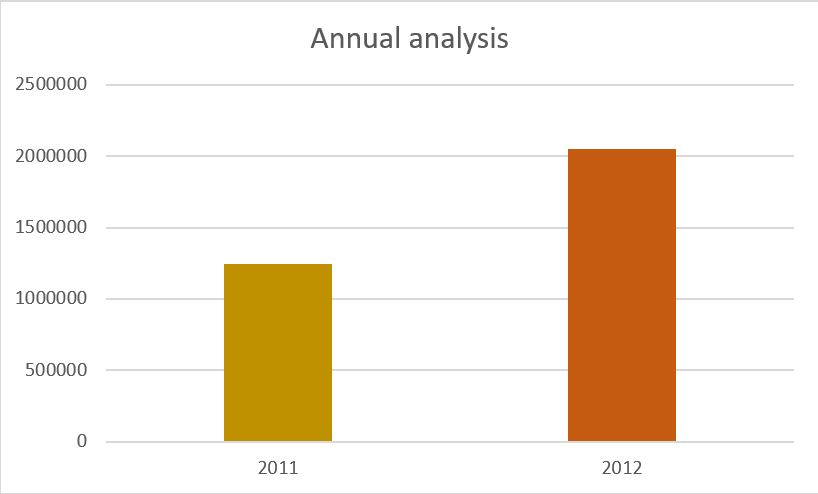
**Figure 6 Annual analysis formulas**

Formuła „SUMIF” w tym przypadku umożliwiła wyznaczenie ilości wynajętych rowerów miejskich z podziałem na lata w których dokonano wynajmów. Pierwszy element formuły „SUMIF” to zakres komórek które zostały poddane weryfikacji. Zakres ten reprezentuje atrybut „yr” w zbiorze danych „Bike sharing”. Drugim elementem formuły jest kryteria która została zdefiniowana w celu segregacji wyników na podstawie wartości znajdujących się w zbiorze komórek zadeklarowanych w pierwszym elemencie formuły „SUMIF”. Wartości zadeklarowanych kryteriów reprezentują lata obowiązujące w zbiorze danych oraz dotyczące roku w których zostały wynajęte rowery. Liczba 0 reprezentuje rok 2011, natomiast liczba 1 odnosi się do roku 2012.



**Figure 7 Output values of annual analysis**

Powyżej przedstawione formuly “SUMIF” wykazaly zestawienie liczbowe rocznej analizy. Rok 2012 zdecydowanie przewyższa liczbę wynajętych rowerow miejskich względem roku poprzedzającego, mianowicie względem roku 2011.



**Figure 8 Annual analysis chart**

Przedstawiony wykres w znacznie lepszy sposób obrazuje jak bardzo wyniki wynajętych rowerów w roku 2012 przewyższają liczbę wynajmów z roku 2011. Wzrost liczby wynajmowanych rowerów na przestrzeni tych dwóch lat wzrósł o 54 punkty procentowe. Taki wynik może sugerować zwiększone zainteresowanie programem wynajmu rowerów, a nawet obserwowalnym wzrostem przyzwyczajenia się społeczeństwa do tego sposobu komunikacji. Wiele firm stosuje nawet formy wynagradzania użytkowników za polecanie ich produktu kolejnym użytkownikom, którzy nie korzystali wcześniej z usług tej firmy.

### Impact of working days and holidays analysis implementation

Analiza tego scenariusza opiera się na dwóch wartościach wyjściowych, natomiast proces ich wyliczenia jest bardziej złożony, ponieważ analiza uwzględnia fakt, że dni powszednich jest znacznie więcej w ciągu roku niż dni wolnych od pracy uwzględniając w to również dni stanowiące święta obowiązujące w kraju pochodzenia danych znajdujących się w zbiorze danych „Bike Sharing”. Aby uwzględnić tę dysproporcję, posłużyłem się formułą „COUNTIF”, która również znajduje się w pakiecie formuł standardowych oraz podstawowych w programie kalkulacyjnym „Excel”.

=COUNTIF( range; criteria)

**Figure 9 COUNTIF formula structure**

Formuła „COUNTIF” składa się z dwóch parametrów. Pierwszym z nich jest zakres komórek zawierających dane, których ilość jest następnie zliczana na podstawie zdefiniowanego warunku. Drugim parametrem jest kryteria, która równocześnie stanowi warunek wymagany aby konkretny wpis mógł zostać doliczony do puli wpisów spełniających wymagania względem deklarowanego w pierwszym parametrze zakresu komórek.

=COUNTIF( H2:H732; 1)

=COUNTIF( H2:H732; 0)

=SUMIF( H2:H732; 1; P2:P732 )

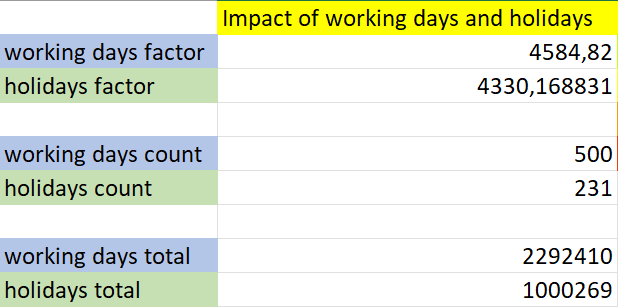
=SUMIF( H2:H732; 0; P2:P732 )

=W8/W5

=W9/W6

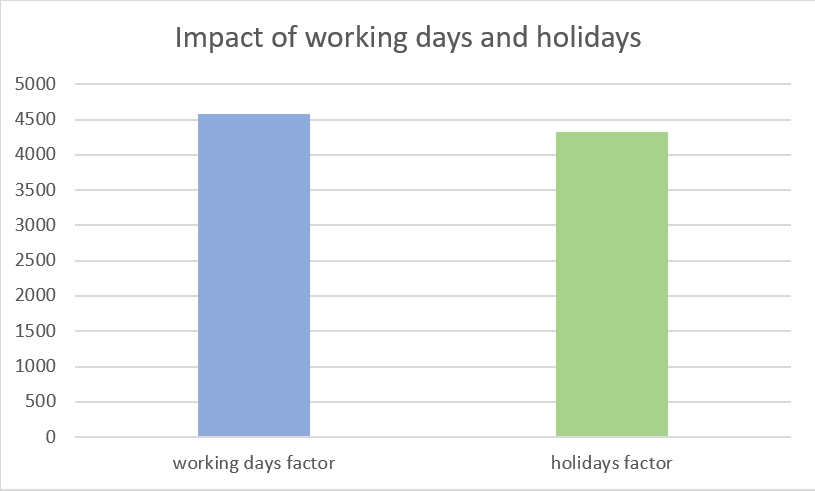
**Figure 10 Impact of working days and holidays analysis formulas**

Formuły statystyczne przedstawione powyżej zostały podzielone na trzy segmenty, każdy z nich wylicza inna, lecz równie kluczowa wartość w kontekście wyniku końcowego który został poddany analizie. Pierwszy segment zapisany przy użyciu formuły „COUNTIF” to segment odpowiadający za wyliczenie dokładnej liczby występujących dni konkretnego rodzaju, spośród dni pracujących oraz dni wolnych od pracy. Pierwszy segment tej formuły to zakres komórek nazwanych atrybutem „workingday” zawierającym informacje dotyczące czy konkretny wpis w zbiorze danych reprezentuje dzień pracowniczy oznaczony liczba 1 czy dzień wolny od pracy oznaczony w zbiorze danych liczba 0. Kolejny segment ter formuly to kryteria, przy pomocy której możliwe jest rozdzielenie ilości dni pracowniczych od dni wolnych od pracy. Drugim segmentem formuł przedstawionych powyżej to segment zawierający formuły „SUMIF”. Ten segment pozwala zliczyć ilość wynajętych rowerów w zbiorze komórek zawierających informacje o rodzaju dnia spośród dni pracowniczych oraz dni wolnych od pracy, zdefiniowanych w elemencie pierwszym formuł znajdujących się w segmencie „SUMIF”. Podobnie jak w poprzednim segmencie, kryteriami są wartości 0 lub 1, które reprezentują rodzaj dnia spośród dni pracowniczych oraz dni wolnych od pracy. Elementem który zawiera zakres komórek których wartości są sumowane to komórki oznaczone atrybutem „cnt”, przedstawiające dokładną liczbę wynajętych rowerów miejskich dla każdego z wpisów znajdujących się w zbiorze danych. Ostatni segment formuł to formuły wykazujące wynik końcowy. Jest to podstawowy typ formuły wykorzystujący zwykle działanie matematyczne, którym w tym przypadku jest dzielenie. Zastosowane dzielenie wylicza ile średnio dziennie jest wypożyczanych rowerów miejskich dla dni pracowniczych i ile jest średnio wypożyczonych rowerów miejskich dla dni wolnych od pracy.



**Figure 11 Output values of impact of working days and holidays analysis**

Przedstawione powyżej wartości liczbowe pokazują jak duże znaczenie ma podział sumy wszystkich wypożyczonych rowerów spośród dni pracowniczych oraz dni wolnych od pracy przez liczbę występujących dni dla konkretnej analizowanej kategorii, ponieważ dla pól opisanych jako „total”, wartości liczbowe różnią się ponad dwukrotnie, natomiast przy wyliczeniu wartości opisanych jako „factor”, różnica ta znacząco się zmniejsza.



**Figure 12 Impact of working days and holidays analysis chart**

Na przedstawionym wykresie, można zauważyć dosłownie niewielka różnicę względem uśrednionej ilości wypożyczonych rowerów miejskich w odniesieniu do dwóch analizowanych przypadków spośród dni pracowniczych oraz dni wolnych od pracy. Wnioski wynikające z tej analizy są takie, ze rodzaj typu dnia spośród dni pracowniczych oraz dni wolnych od pracy ma nieznaczny wpływ na ilość wypożyczonych rowerów miejskich. Jednakże ta niewielka różnica może wynikać z tego, że pewna grupa użytkowników stanowiących mniejszość może spędzać dni świąteczne z rodziną niż na korzystaniu ze środka komunikacji jakim są rowery miejskie.

### Weather influence analysis implementation

## Back-end code

### Initialization classes

# Summary

## Application outcome

The application "GeoFinderUI" is a program that returns to the user suggestive data about a given location, which were obtained from the minimum input data provided by the user. All search results are subject to the risk of incorrect data, this is due to the program finding the first, most matching search result to the input data entered. The idea behind this application is to reduce as much as possible the time needed to obtain geolocation data required in a defined business environment, which requires obtaining specific output data. The application was successfully formulated and written, thus achieving a fully functional program. Thanks to this project, I have achieved a much broader perspective on IT topics, mostly related to programming. My knowledge has been enriched with a lot of new information, skills and experience.

## Acknowledgments

The most interesting and rewarding part of the application development process was the achievement of the set goals. Thanks are due to Maksymilian Kubica, M.Sc. for his professional supervision of the project and for providing many valuable tips during the process of writing the thesis. The knowledge and experience he imparted are invaluable, and his management of the project allowed it to develop properly. For providing a place to work, help during brainstorming, and support, thanks are due to all those close to me - parents and friends.

# Bibliography

* *Aplikacja internetowa*. (2022, December). From Wikipedia: https://pl.wikipedia.org/wiki/Aplikacja\_internetowa
* *Bot program*. (2022, May). From Wikipedia: https://pl.wikipedia.org/wiki/Bot\_(program)
* *Code Refactoring*. (2023, January). From Refactoring: https://refactoring.com/
* *Discover Intellij Idea*. (2023, January). From jetbrains: https://www.jetbrains.com/help/idea/discover-intellij-idea.html
* Flower, M., Beck, K., Opdyke, W., & Brant, J. (2002). Refactoring Helps You Find Bugs. In M. Flower, *Refactoring: Improving the design of Existing Code* (pp. 48-49). Addison-Wesley.
* *Framework*. (2022, October). From Wikipedia: https://pl.wikipedia.org/wiki/Framework
* *Google Maps*. (2022, May). From https://www.google.com/maps/about/#!/
* *HTML*. (2023, January). From https://html.spec.whatwg.org/
* *Markup language*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Markup\_language
* *Materialize*. (2023, January). From https://materializecss.com/
* *Microsoft Excel*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Microsoft\_Excel
* *Openpyxl*. (2022, May). From https://openpyxl.readthedocs.io/en/stable/
* *Pandas software*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Pandas\_(software)
* *Python*. (2023, January). From https://docs.python.org/pl/3/
* *Selenium*. (2022, December). From https://www.selenium.dev/
* *Uniform Resource Locator*. (2022, December). From Wikipedia: https://pl.wikipedia.org/wiki/Uniform\_Resource\_Locator
* *Web scraping*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Web\_scraping

# Attachment

## Table of Figures

[**Figure 1 Presentation markup example** 8](#_Toc127208177)

[**Figure 2 Procedural markup example** 9](#_Toc127208178)

[**Figure 3 Descriptive markup example** 10](#_Toc127208179)

[**Figure 4 File Template.xlsx** 13](#_Toc127208180)

[**Figure 5 Input file with test data** 13](#_Toc127208181)

[**Figure 6 Output file with test data** 14](#_Toc127208182)

[**Figure 7 Google Maps search result** 23](#_Toc127208183)

[**Figure 8 GeoFinderUI application home page with Materialized styling packages**.......................................................................................................................24](#_Toc127208184)

[**Figure 9 the original diagrammatic process of the operation of the GeofinderUI application** 26](#_Toc127208185)

[**Figure 10 sample visualization of the GeoFinderUI start page** 27](#_Toc127208186)

[**Figure 11 data acquisition points by GeoFinderUI application** 27](#_Toc127208187)

[**Figure 12 Manual geo finder use case diagram** 28](#_Toc127208188)

[**Figure 13 GeoFinderUI use case diagram** 29](#_Toc127208189)

[**Figure 14 Information about the authors of materializecss.com** 34](#_Toc127208190)

## Table of Pictures

[**Picture 1 Left side, Martin Flower Picture 2 Right side, William Opdyke** 7](#_Toc126793112)

[**Picture 3 Left side, IntelliJ dark theme Picture 4 Right side, IntelliJ default light theme** 13](#_Toc126793113)

[**Picture 5 Left side, Nyan Progress Bar Picture 6 Right side, Snyk add-in** 14](#_Toc126793114)

[**Picture 7 Left side, Excel Worksheet Picture 8 Right side, Visual Basic for Applications** 15](#_Toc126793115)

[**Picture 9 Left Side, CWI building Picture 10 Right side, Guido van Rossum** 16](#_Toc126793116)

[**Picture 11 Jason Huggins** 17](#_Toc126793117)

[**Picture 12 Wes McKinney** 18](#_Toc126793118)

[**Picture 13 Left side, Eric Gazoni Picture 14 Right side, Charlie Clark** 19](#_Toc126793119)

[**Picture 15 Left side, Timothy Berners-Lee Picture 16 Right side, Robert Cailliau**.........................................................................................................................20](#_Toc126793120)

## Table of Code

[**Code 1 The <Head> section with the style part needed for positioning the content of the GeoFinderUI start page** 26](#_Toc126793275)

[**Code 2 Part of the code that allows the user to download the official version of the data template acceptable by the application** 27](#_Toc126793276)

[**Code 3 The form section needed for receiving and sending files and information to the back-end code** 28](#_Toc126793277)

[**Code 4 The part of the code responsible for the footer visible on each page of the "GeoFinderUI" web application** 29](#_Toc126793278)

[**Code 5 The part of the code that contains the information and its layout on the summary page** 30](#_Toc126793279)

[**Code 6 Part of the code containing the import of the necessary tools used in the "GeoFinderUI" application** 31](#_Toc126793280)

[**Code 7 Side methods in "home.py"** 32](#_Toc126793281)

[**Code 8 The beginning of the main method responsible for the functionality of the application** 32](#_Toc126793282)

[**Code 9 Part of the code responsible for preparing and cleaning the location for storing data files** 33](#_Toc126793283)

[**Code 10 Part of the code responsible for preparing the data file and launching the automated browser** 34](#_Toc126793284)

[**Code 11 A loop procedure that goes through all the records entered in the data file** 34](#_Toc126793285)

[**Code 12 Part of code with nested "if" statements that returns final data values** 36](#_Toc126793286)

[**Code 13 Last part of main class code** 37](#_Toc126793287)

[**Code 14 Class dedicated to accept cookies on website** 38](#_Toc126793288)

[**Code 15 A class that returns address value from name value** 39](#_Toc126793289)

[**Code 16 A class that returns city value from zip-code value** 39](#_Toc126793290)

[**Code 17 A class that returns name value from address value and city value** 41](#_Toc126793291)

[**Code 18 A class that returns city value from name value** 43](#_Toc126793292)

[**Code 19 A class that returns zip-code value from name value** 44](#_Toc126793293)

[**Code 20 A class that returns city value from address value** 45](#_Toc126793294)

[**Code 21 A class that returns zip-code value from address value** 47](#_Toc126793295)

## What CD contains

* **\Szymon\_Wujec\_s20431\_DiplomaThesis\_EPG**
  + Electronic version of engineering thesis
* **\GeoFinderUI**
  + All files necessary for the functioning of the web application