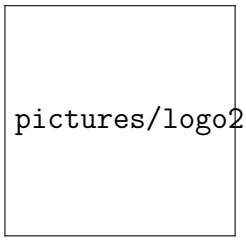


Zachodniopomorski Uniwersytet
Technologiczny w Szczecinie

Praca magisterska

Analiza danych giełdowych przy pomocy narzędzi dostępnych w pakiecie scikit-learn

Łukasz Połoń

Kierunek: Informatyka
Specjalność: Inżynieria oprogramowania

Nr albumu: 24942

Promotor
dr inż. Piotr Błaszyński

Wydział Informatyczny

Szczecin 2018

Oświadczenie autora

Ja, niżej podpisany Łukasz Połoń oświadczam, że praca ta została napisana samodzielnie i wykorzystywała (poza zdobytą na studiach wiedzą) jedynie wyniki prac zamieszczonych w spisie literatury.

.....
(Podpis autora)

Oświadczenie promotora

Oświadczam, że praca spełnia wymogi stawiane pracom magisterskim.

.....
(Podpis promotora)

Spis treści

Streszczenie

Przykładowe streszczenie i test polskich znaków: ąśćżźłóęą

Słowa kluczowe

Python, Scikit-learn, Giełda Papierów Wartościowych, Kivy, Matplotlib, Pandas, Analiza regresji

Abstract

Keywords

Python, Scikit-learn, Stock Market, Kivy, Matplotlib, Pandas, Regression analysis

Wprowadzenie

Giełda Papierów Wartościowych jest to instytucja, która prowadzi działalność w zakresie organizacji obrotu papierami wartościowymi i instrumentami finansowymi[1]. W praktyce spełnia ona rolę pośrednika finansowego pomiędzy kupującym, a sprzedającym papiery wartościowe. Dane generowane przez giełdę poddawane są ciągłym analizom, w szczególności w celu dostarczenia informacji potrzebnych do właściwego zarządzania kapitałem.
Istnieją dwie główne metody analizy danych giełdowych: analiza fundamentalna i analiza techniczna[2]. Pierwsza z nich polega na analizie faktycznej kondycji finansowej podmiotu, podczas gdy druga ma za zadanie prognozowanie przyszłych wartości wskaźników na podstawie zebranych danych.

Temat tej pracy podejmuje opisanie i przeprowadzenie wybranych metod analitycznych dostępnych w pakiecie scikit-learn. Pakiet ten jest biblioteką języka programowania Python umożliwiającą wysokopoziomowe przetwarzanie danych. Udostępnia wiele algorytmów klasyfikacji, regresji oraz uczenia maszynowego, które mogą zostać wykorzystane do przeprowadzania obliczeń między innymi na potrzeby analizy technicznej, której to elementy zostaną tutaj przedstawione.

Rozdział 1

Analiza danych statystycznych

1.1. Rynki finansowe

Giełda jest zbiorem instytucji finansowych, w których odbywa się wymiana papierów wartościowych pomiędzy kupującymi i sprzedającymi[3]. Powinna ona koncentrować popyt i podaż na papiery wartościowe, co prowadzi do kształtowania się powszechnego kursu. Zapewnia ona również bezpieczny i uregulowany przebieg transakcji oraz upowszechnia informacje, które umożliwiają ocenę aktualnej wartości papierów wartościowych.

Dodać informacje o różnych giełdach na świecie!

Papier wartościowy, zgodnie z prawem o obrocie papierami wartościowymi, to *"dokument, mający stwierdzać lub stwierdzający istnienie określonego prawa majątkowego utrwalony w takim brzmieniu i w taki sposób, że może stanowić samodzielny przedmiot obrotu publicznego"*[4] Papierami wartościowymi mogą być między innymi: akcje, obligacje, czeki, bony skarbowe. Akcje dają nabywcy (akcjonariuszowi) prawo do współwłasności w spółce, która je wyemitowała, co przekłada się na bezpośredni udział w wypracowanych zyskach (dywidendy), a także nabycie praw korporacyjnych, umożliwiających decyzyjność w spółce[5].

Obligacje są to papiery dłużne, które poświadczają wierzytelność pomiędzy właścicielem, a dłużnikiem. W ich przypadku dłużnikiem jest emitent, który zobowiązuje się do uregulowania wierzytelności w określonym czasie. Obligacje emituje się w celu pozyskania kapitału, a obligatoriusz, czyli nabywca, otrzymuje prawo do całkowitego zwrotu inwestycji po upłynięciu określonego terminu, a także do otrzymywania stałego dochodu określonego odsetkami[6].

Dodać akapit o danych giełdowych!

1.2. Ekonometria i metody analizy danych finansowych

Podejmowanie decyzji inwestycyjnych na giełdzie papierów wartościowych nieuchronnie wiąże się z ryzykiem straty zainwestowanego kapitału. Zmniejszenie tego ryzyka jest więc kluczowym działaniem inwestorów, którzy oczekują wymiernych zwrotów z prowadzonych inwestycji. Ekonometria jako nauka zajmuje się dostarczeniem metod i narzędzi potrzebnych do przeprowadzenia niezbędnych analiz rynku,

dzięki którym potencjalny inwestor może ocenić ryzyko inwestycyjne oraz je zminimalizować. Rozwój gospodarki i rynków światowych wymusza niejako wzrost zapotrzebowania na coraz to bardziej zaawansowane i dokładne narzędzia, które wspomagają podejmowanie decyzji. Na decyzje z kolei mają wpływ czynniki, które można podzielić na jakościowe i ilościowe[7].

Czynniki jakościowe, ze względu na swoją naturę i duży wpływ osób dokonujących analizy, nie mogą podlegać bezpośredniej mierzalności, przez co uważane są za subiektywne[7]. Czynniki ilościowe natomiast umożliwiają ocenę na podstawie danych liczbowych, ich wzajemnych powiązań i relacji. Są więc uważane za bardziej obiektywne, lecz nie mniej ważne.

1.2.1. Model ekonometryczny

Ekonometrię finansową można więc dogłębniej zdefiniować jako zastosowanie metod ilościowych do analizy zjawisk na rynku finansowym[7].
Podstawowym krokiem, który należy postawić podczas analizy ekonometrycznej jest określenie modelu. Model jest uproszczoną reprezentacją rzeczywistego procesu, który podlega naszym badaniom[9]. Powinien więc możliwie dokładnie opisywać badane procesy, z uwzględnieniem właściwych zmiennych. Proste modele z reguły upraszczają rzeczywistość, co w połączeniu z przyjętymi założeniami niekoniecznie będącymi zgodne ze stanem faktycznym pozwalają domniemywać wadliwość modelu. Jednakże dokładność modelu często zależy od dostępności danych. Jedna z koncepcji konstruowania modelu zakłada, że warto rozpoczynać badania od modelu prostego i w miarę poznawania dodatkowych danych, sukcesywnie zwiększać jego złożoność.

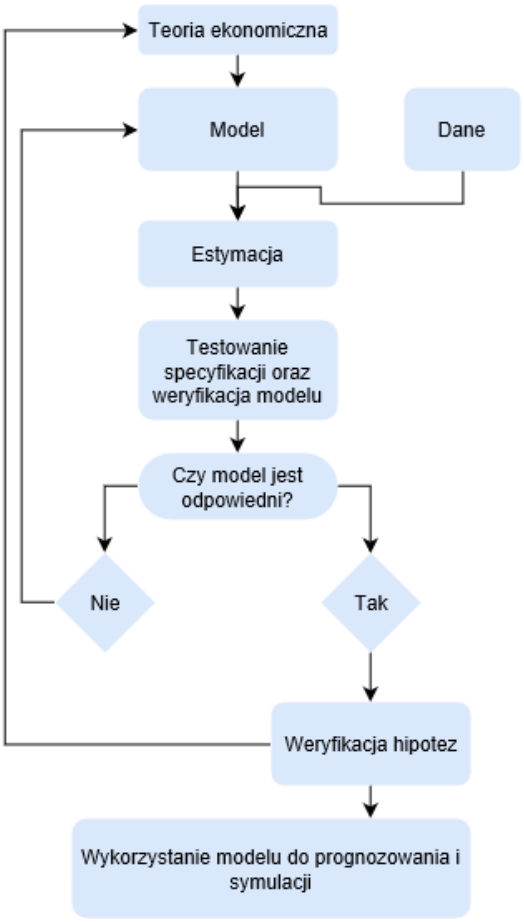
Model ekonometryczny składa się z[9]:
- Zbioru równań behawioralnych, czyli opisujących zachowanie
- Opisu możliwych błędów pomiaru lub obserwacji
- Specyfikacji rozkładu prawdopodobieństwa zakłóceń i błędów pomiaru

Model jest częścią algorytmu analizy ekonometrycznej, który został przedstawiony na diagramie 1.1.

Pierwszymi krokami jakie należy wykonać podczas analizy eknometrycznej są opracowanie teorii, modelu, oraz skompletowanie niezbędnych danych. Następnie model podlega estymacji i weryfikacji, co pozwala na ocenę jego poprawności. Kolejne iteracje tego algorytmu prowadzą do opracowania dokłądnego modelu, który w jak najlepszym stopniu opisuje zjawisko, które jest aktualnie badane.

1.2.2. Analiza techniczna

Analiza techniczna skupia się głównie na bezpośrednich, mierzalnych sposobach oceny aktualnej kondycji rynku. Podejmuje analizę jego aktywności, w tym cen, ilości transakcji w danym okresie czasu, poprzez badanie wzajemnych zależności oraz wzor-



Rysunek 1.1. Schemat kroków w ekonometrycznej analizie modeli ekonomicznych

ców.
Oparta jest na trzech zasadach:[8]

- Wszystkie czynniki, które wpływają na rynkową cenę instrumentu finansowego znajdują swoje odzwieciedlenie w cenie tego instrumentu

- Ceny zawsze podlegają określonym trendom

- Historia się powtarza - założenie zakładające, że ceny na rynku zmieniają się i powtarzają zgodnie z określonymi wzorami, które wydarzyły się w przeszłości

Klasyczna analiza techniczna opiera się na teorii Dowa, która jest pierwszą teorią stanowiącą fundament dla analizy tego typu. Założenia teorii Dowa to[10]:

- Średnie giełdowe dyskontują wszystko

- Istnieją trzy kategorie trendu rynkowego: trendy główne, wtórne i mniejsze

- Wolumen potwierdza trend

Główny trend rynkowy jest trendem o największym znaczeniu, trwającym zwykle od roku do kilkunastu lat. Trendy mniejsze, trwające do trzech tygodni, są marginalizowane i utożsamiane z czynnikami losowymi, nie mającymi realnego wpływu na trend główny. Trendy wtórne natomiast korygują trendy główne i trwają od kilku tygodni do trzech miesięcy. Są one szczególnie ważne i utożsamiane z korektą techniczną trendu głównego.

Trend główny dzieli się na trzy fazy. Pierwsza faza, zwana fazą akumulacji[10] oznacza, iż podczas hossy inwestorzy skupują akcje. Faza druga charakteryzuje się przetrzymaniem akcji i niepodejmowaniem działań przez inwestorów, w oczekiwaniu na dalsze informacje. Trzecia faza natomiast związana jest ze sprzedażą akcji, a także z większym zainteresowaniem inwestorów indywidualnych, którzy w tym momencie zaczynają skupować akcje.
W przypadku bessy, która również dzieli się na trzy fazy, faza pierwsza oznacza wyprzedaż akcji przez wtajemniczonych inwestorów, faza druga jest nasileniem tej tendencji, a faza trzecia charakteryzuje się stagnacją.

Z opisu przebiegów trendu głównego podczas hossy i bessy wynika, że związany jest on bezpośrednio z wartościami wolumenu obrotów. Wolumen jest to łączna liczba transakcji przeprowadzonych dla danego papieru wartościowego. Tak więc, jeśli wraz ze wzrostem ceny, wzrasta wolumen obrotów, można potwierdzić występowanie hossy[10].

Narzędzia analizy technicznej, ponieważ bazują między innymi na teorii Dowa, są najczęściej wykorzystywane do analizy trendów, ich identyfikacji oraz wychwytywania oznak ich odwrócenia.

1.2.3. Analiza fundamentalna

Analiza fundamentalna, w odróżnieniu od analizy technicznej, ma na celu opis danej spółki oraz jej otoczenia. Czynniki brane pod uwagę w tego typu analizie to najczęściej sytuacja gospodarcza kraju, czyli między innymi poziom PKB, inflacja, czy wielkość rynku[10], oraz sytuacja gospodarcza samego przedsiębiorstwa.

Jednym z najstarszych lecz wciąż skutecznych sposobów mierzenia pozycji danego przedsiębiorstwa na rynku jest macierz BCG (Boston Consulting Group). Pozwala ona na przypisanie przedsiębiorstwa do jednej z czterech kategorii:
- Gwiazdy
- Obiecujące
- Dojne krowy
- Psy

Kategoria pierwsza, czyli gwiazdy, to przedsiębiorstwa z największym udziałem sprzedaży w odniesieniu do całego rynku, a także o najwyższej pozycji konkurencyjnej. Przedsiębiorstwa kategorii drugiej charakteryzują się szybkim wzrostem sprzedaży, lecz równolegle intensywnie inwestujące, przez co są pozbawione zapasów kapitału. Dojne krowy są stosunkowo podobne do przedsiębiorstw kategorii drugiej, lecz w ich przypadku intensywność inwestycyjna jest minimalna. Są zazwyczaj stabilne, a ich udział w rynku pozostaje na stałym poziomie. Kategoria czwarta określa podmioty nieobiecujące, które posiadają niewielki udział w rynku, wraz z brakiem perspektyw do rozwoju.

1.3. Statystyczne metody analizy danych

1.3.1. Klasyfikacja

Klasyfikacja jest jednym z najbardziej podstawowych metod analizy danych statystycznych. Jej głównym zadaniem jest przyporządkowanie klas do obiektów z danego zbioru danych. Zagadnienia uczenia maszynowego dzielą klasyfikację na nadzorowaną i klasyfikację bez nadzoru[13].

Algorytmy klasyfikacji nadzorowanej charakteryzują się dostępnością testowego zbioru danych z przypisanymi klasami, który spełnia rolę wzorca dla pozostałych danych. Algorytm klasyfikacji bez nadzoru pozbawiony jest tego typu informacji, często również brak jest informacji jakie klasy ma tworzyć dany zbiór. Zadaniem takiego algortmu jest więc wydzielenie i powiązanie ze sobą danych w taki sposób, aby stworzyć klasy i przyporządkować do nich odpowiednie dane.

Istnieje wiele algorytmów kalsyfikacji, których zastosowanie praktyczne uzależnione jest od czynników takich jak szybkość działania, zużycie pamięci, łatwość interpretacji, oraz oczywiście trafność predykcji[12].
Wśród nich możemy wyróżnić:
- Naiwny klasyfikator Bayes'a
- Drzewa decyzyjne
- Algorytm najbliższego sąsiada
- Maszyna wektorów nośnych (SVM)

Rysunek 1.2. Klasyfikacja nienadzorowana

- Liniowa analiza dyskryminacyjna

1.3.1.1. Naiwny kalsyfikator Bayes'a

W 1763 roku Thomas Bayes przestawił w swojej pracy twierdzenie teorii prawdopodobieństwa, które warunkuje prawdopodobieństwa dwóch zdarzeń warunkujących się na wzajem.
Brzmi ono [14]: *Dla dowolnej hipotezy* $h \in H$ *oraz zbioru danych* D *zachodzi równość*

$$Pr(h \mid D) = \frac{Pr(h)Pr(D \mid h)}{Pr(D)} \tag{1.1}$$

Prawdopodobieństwo *Pr(h)* jest prawdopodobieństwem *a priori* co oznacza, że przy jego określaniu nie były brane pod uwagę dane, które mogły mieć wpływ na jego wartość. Poprzez ich uwzględnienie określone zostaje prawdopodobieństwo *a posteriori* $Pr(h \mid D)$. Wzór Bayesa wyraża zatem związek między tymi dwoma prawdopodobieństwami, natomiast do jego wyrażenia używane są jeszcze prawdodopodbieństwo zaobserwowania danych *Pr(D)*, oraz prawdopodobieńswto zaobserwowania tych danych przy założeniu poprawności hipotezy $Pr(D \mid h)$.

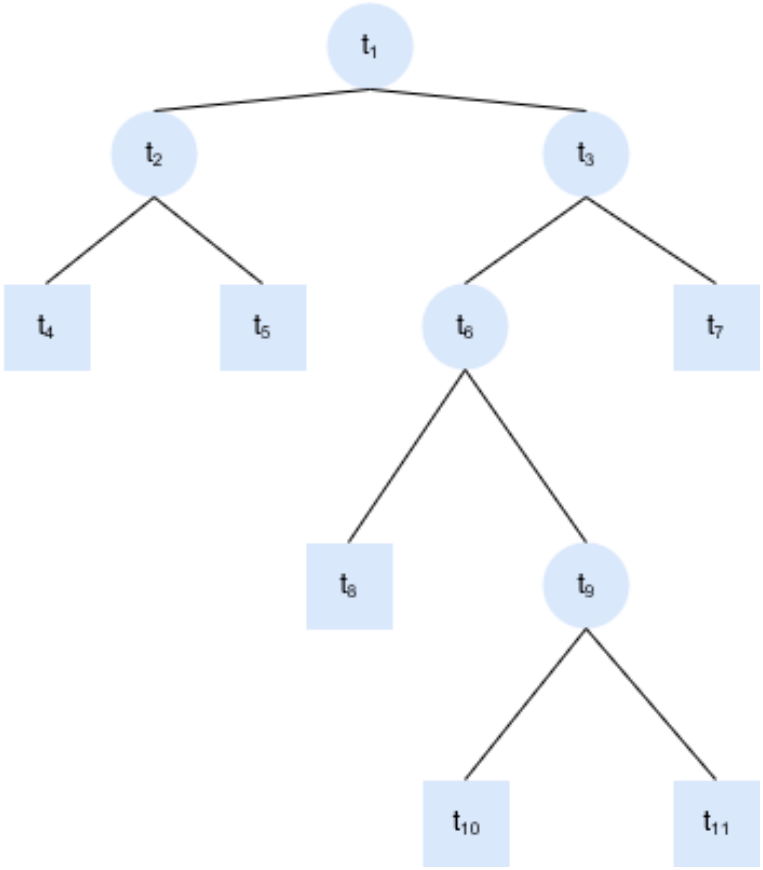
W klasyfikatorze Bayes'a wybór decyzja o wyborze odpowiedniej hipotezy h ze zbioru hipotez H nie jest podejmowana na podstawie dokładności czy złożoności, lecz prawdopodobieństwa. Wybór dokonywany jest na podstawie dwóch sposobów, dzięki którym można uznać hipotezę za najlepszą[14]:
- Zasada maksymalnej zgodności
- Zasada maksynalnego prawdopodobieństwa *a posteriori*

Pierwsza z nich jest nazywana zasadą ML (*Maximum Likelihood*) i mówi, że najlepszą jest hipoteza $h_{ML} \in H$, która maksymalizuje warunkowe prawdopodobieństwo danych treningowych

$$h_{ML} = arg \overset{h \in H}{max}\ Pr(T \mid h) \tag{1.2}$$

Kolejna, nazywana zasadą MAP (*Maximum a posteriori*) wyjaśnia, że należy wybrać taką hipotezę $h_{MAP} \in H$, która posiada maksymalne prawdopodobieństwo *a posteriori*:

$$h_{MAP} = arg \overset{h \in H}{max}\ Pr(h \mid T) \tag{1.3}$$



Rysunek 1.3. Drzewo decyzyjne

1.3.1.2. Drzewa decyzyjne

Algorytmy drzew decyzyjnych są jednymi z najszerzej stosowanych metod analizy danych za pomocą uczenia maszynowego. Polegają na sekwencyjnym podziale danego zbioru danych na dwa rozłączne podzbiory w taki sposób, aby oba podzbiory były możliwie jednorodne[15]. Działanie drzewa decyzyjnego ilustruje Rysunek 1.3.

Wierzchołki drzewa oznaczone literą t są nazywane węzłami i stanowią podzbiory zbioru danych. Węzły oznaczone okręgami są węzłami wewnętrznymi, natomiast kwadratami - węzłami zewnętrznymi. Dla każdego węzła określona jest funkcja podziału, która każdemu elementowi do niego należącymi przypisuje jedną z wartości : *Prawdę* lub *Fałsz*.
Klasyfikatory zbudowane na podstawie drzewa decyzyjnego mają więc postać[15]:

$$d_T(x) = \sum_{t \in T} ind(t) I(x \in t) \tag{1.4}$$

Zaletą algorytmów drzew decyzyjnych jest niewątpliwie możliwość bezproblemowego wykorzystania zarówno jakościowych jak i ilościowych do klasyfikacji. Algorytmy te wykazują również odporność na sytuacje braku części zmiennych, a także na obserwacje odstające[15].

1.3.2. Analiza regresji

Jedną z metod umożliwiających predykcję wartości na podstawie szeregu innych jest analiza regresji. Polega ona na odszukiwaniu i opisie związków między zmiennymi, co prowadzi do stworzenia modelu, czyli równania regresji, które pozwala na analizę struktury zależności i umożliwia prognozowanie.

Model regresji liniowej jest to funkcja liniowa zmiennych objaśniających i składnika losowego[16]. Ma postać:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k + \xi \tag{1.5}$$

Symbole β_k odpowiadają współczynnikom regresji cząstkowej, które są najważniejszą częścią równania. Informują o zmianie zmiennej Y podczas zmiany zmiennej mu odpowiadającej przy założeniu, że pozostałe zmienne nie uległy zmianie. Wyraz wolny regresji jest wyrażony symbolem β_0 i zazwyczaj nie podlega interpretacji[16]. Składnik losowy oznaczany jest przez ξ, a zmienne Y i X nazywane są odpowiednio zmienną objaśnianą (zależną) i zmiennymi objaśniającymi (niezależnymi).

Zasadniczą rolę podczas analizy regresji odgrywa estymacja parametrów modelu. Ma ona na celu znalezienie wartości ocen parametrów na podstawie danych z próby[16]. Tak więc dzięki estymacji można uzyskać takie wartości ocen, które sprawiają iż model regresji jak najlepiej pasuje do danych. W przypadku regresji liniowej z jedną zmienną objaśniającą, gdzie równanie regresji przyjmuje postać prostej na wykresie, uzyskujemy jak najlepsze dopasowanie tej prostej do punktów na wykresie rozrzutu danych.

Do przeprowadzenia estymacji parametrów modelu regresji liniowej najczęściej stosowana jest Metoda Najmniejszych Kwadratów. Pozwala ona na znalezienie ocen parametrów o najmniejszej wartości kwadratów odchyleń pomiędzy rzeczywistymi a teoretycznymi wartościami zmiennej objaśnianej[16]. Oznaczanie wartości dopasowanej $\hat{y}_i$ za pomocą wzoru regresji liniowej jest przewidywaniem tej zmiennej na podstawie modelu. Zmienna ta jest różna od rzeczywistej wartości y_i, a różnica pomiędzy nimi nazywana jest *resztą*, którą można zdefiniować jako[17]:

$$e_i = y_i - x_{1i} b_1 - x_{2i} b_2 - ... - x_{ki} b_k = y_i - \hat{y}_i \tag{1.6}$$

Dopasowanie modelu jest tym lepsze, im mniejsza jest różnica pomiędzy wartościami dopasowanymi i rzeczywistymi. Na wykresach jest to przedstawione jako wartość bezwzględna odległości punktu od prostej regresji.

Rozdział 2

Zastosowanie języka programowania Python w obliczeniach analitycznych

Python jest językiem programowania wysokiego poziomu, charakteryzujący się przede wszystkim wysoką klarownością i czytelnością kodu. Jest to język interpretowany, co w odróżnieniu od języków kompilowanych pozwala na bardzo szybkie tworzenie i testowanie kodu. Wadą tego rozwiązania jest niestety spadek wydajności oraz zwiększone zużycie pamięci i procesora, jednak zastosowania praktyczne Pythona zazwyczaj pozwalają na poniesienie tego typu kosztów.

Python został stworzony w 1989 roku przez Guido van Rossum, a do dzisiaj rozwijany jest jako projekt Open Source i zarządzany przez organizację non-profit Python Software Foundation. Jego specyficzna struktura oraz cechy takie jak dynamiczne typowanie, automatyczna zarządzanie pamięcią, przenośność, czy duża czytelność i prostota kodu, umożliwiają bardzo szybkie wytwarzanie i utrzymywanie aplikacji.

Biblioteka standardowa języka Python zawiera wiele użytecznych modułów i gotowych rozwiązań, które wspomagają szybką i efektywną implementację kodu. Ponadto dostępny jest *Python Package Index* (PyPI) - zbiór paczek zewnętrznych, tworzonych przez niezależnych programistów, dystrybuowanych na licencjach Open Source. Dzięki takiej mnogości pakietów i modułów język Python może być wykorzystywany w wielu projektach, łącząc różne technologie i dziedziny informatyki. Jednym z przykładów wykorzystania tego języka jest tworzenie aplikacji internetowych za pomocą frameworku Django. Łączy on ze sobą różne technologie wykorzystywane przy tworzeniu serwisów internetowych, zapewniając bardzo dobry mechanizm back-endowy oraz wygodne środowisko.

Ze względu na wyżej wymienione cechy Python znalazł również zastosowanie w analityce i analizie danych, włączając w to analizę danych statystycznych i giełdowych, a także we wspomaganiu obliczeń matematycznych.

2.1. Cechy charakterystyczne języka Python

Podstawową charakterystyczną cechą języka Python jest fakt, iż nie jest on kompilowany lecz interpretowany, czyli tłumaczony do wykonywalnego kodu maszynowego

lub kodu pośredniego. Dzięki użyciu interpretera w konsoli systemowej można bezpośrednio wykonywać kod Pythona w czasie rzeczywistym.

```
Python 2.7.12 (default, Nov 20 2017, 18:23:56)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from sklearn import datasets
>>>
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
>>> digits.data
array([[  0.,   0.,   5., ...,   0.,   0.,   0.],
       [  0.,   0.,   0., ...,  10.,   0.,   0.],
       [  0.,   0.,   0., ...,  16.,   9.,   0.],
       ...,
       [  0.,   0.,   1., ...,   6.,   0.,   0.],
       [  0.,   0.,   2., ...,  12.,   0.,   0.],
       [  0.,   0.,  10., ...,  12.,   1.,   0.]])
>>>
>>>
```

Pozwala to na bardzo szybkie testowanie niewielkich fragmentów kodu, użycia bibliotek, a także przeprowadzanie testowych obliczeń. Jest to również doskonałe narzędzie do sprawdzania i dostosowywania środowiska, w szczególności gdy użyte zostaje symulowane środowisko - program *virtualenv*, który instaluje wybraną wersję interpretera we wskazanym katalogu i umożliwia instalowanie bibliotek niezależnie od tych, które zainstalowane są w systemie.

Kolejną wartą uwagi cechą języka Python jest jego składnia. W odróżnieniu od języków takich jak na przykład Java czy C++, w Pythonie zastosowano tak zwane dynamiczne typowanie. Oznacza to, że podczas definiowania zmiennych nie określa się ich typu. Jest to możliwe, ponieważ w języku Python każdy element, na przykład funkcja, klasa czy też struktura danych jest obiektem. Obiekt ten ma z góry zdefiniowany typ, więc przypisanie jego referencji do konkretnej zmiennej pomaga go w ten sposób określić.

```
  Python 2.7.12 (default, Nov 20 2017, 18:23:56)
  [GCC 5.4.0 20160609] on linux2
  Type "help", "copyright", "credits" or "license" for more information.
  >>> variable_one = 44
  >>> type(variable_one)
  <type 'int'>
  >>>
  >>> variable_one = 'text'
  >>> type(variable_one)
  <type 'str'>
  >>>
```

Jedną z najbardziej użytecznych cech Pythona jest zastosowanie elementów programowania funcyjnego. Elementami takimi są przykładowo wyrażenia *lambda*, oraz *list comprehension* i *dict comprehension*. Wyrażenia *lambda* pozwalają na stworzenie i przypisanie do zmiennej ktrótkiej funkcji, która jest w stanie przyjmować argumenty oraz zwracać wartości. Znajduje to zastosowanie w przypadkach, które wymagają wielokrotnego wykorzystania danego fragmentu kodu, a użycie ich skraca znacznie ilość wypisanych poleceń. Pozwala to uniknąć tworzenia wielu krótkich funkcji lub metod poza obecnie wykorzystywaną przestrzenią, co często wpływa pozytywnie przede wszsytkim na czytelność kodu.

Wyrażenia *list comprehension* oraz *dict comprehension* wykorzystywane są do szybkiego tworzenia odpowiednio list oraz słowników. W swojej konstrukcji zawierają pętlę *For*, która iteruje po wskazanej strukturze, na przykład liście, zwracając w każdym kroku jeden jej element. Element ten może być sprawdzony warunkiem wbudowanym w strukturę, oraz następnie zmieniony i wbudowany w nową listę lub słownik.

```
1  Python 2.7.12 (default, Nov 20 2017, 18:23:56)
2  [GCC 5.4.0 20160609] on linux2
3  Type "help", "copyright", "credits" or "license" for more information.
4  >>> test_lambda = lambda x: x+5
5  >>> test_lambda(10)
6  15
7  >>>
8  >>> test_list = [1, 2, 3, 4, 5]
9  >>>
10 >>> list_comprehension = [x+5 for x in test_list]
11 >>> list_comprehension
12 [6, 7, 8, 9, 10]
13 >>>
14 >>> dict_comprehension = {x: x+1 for x in test_list}
15 >>> dict_comprehension
16 {1: 2, 2: 3, 3: 4, 4: 5, 5: 6}
17 >>>
```

Naturalnie, natura i składnia języka Python jest o wiele bardziej różnorodna, a przedstawione przykłady odzwierciedlają jedynie namiastkę jego możliwości. Należałoby wspomnieć tutaj między innymi o posługiwaniu się choćby wbudowanymi strukturami danych, wykorzystaniu programowania orientowanego obiektowo oraz typowych dla niego elementach. Niemniej jednak, biorąc pod uwagę temat niniejszej pracy, którym jest przedstawienie możliwości biblioteki *Scikit-learn*, wyżej wymienione podstawy uzupełnione późniejszymi wyjaśnieniami powinny wystarczyć aby w pełni zrozumieć naturę problemu.

2.2. Python w obliczeniach analitycznych

Język programowania Python jest bardzo dobrym narzędziem wspomagającym obliczenia analityczne. Cechy tego języka zapewniają skoncentrowanie się na bezpośrednim podejściu do problemu tworzenia algorytmów i modeli, minimalizując czas

projektowania od podstaw skomplikowanych algorytmów pomocniczych. Jednak największą zaletą tego języka jest dostępność wielu bibliotek z gotowymi rozwiązaniami, które mogą zostać wykorzystane do sprawnej implementacji modeli analitycznych. Podstawowymi bibliotekami wspomagającymi przeprowadzanie obliczeń matematycznych są *NumPy* i *SciPy*.

Pierwsza z nich dostarcza przede wszystkim obiekty wielowymiarowych list oraz szereg metod i funkcji umożliwiających szybką manipulację, przetwarzanie i stortowanie. Zawiera także zestaw metod pozwalających na przeprowadzanie podstawowych działań statystycznych i matematycznych[19]. Stosowana jest w wielu innych bibliotekach analitycznych, na przykład w pakiecie *Scikit-learn*.

Podstawową różnicą pomiędzy obiektami *array* z pakietu *NumPy*, a wbudowanymi listami języka Python jest fakt, iż podczas tworzenia obiektu ustala się stały rozmiar struktury, a każde zwiększenie tego rozmiaru powoduje powstanie nowego obiektu i usunięcie poprzedniego.

```
>>> python_list = [1, 2, 3, 4]
>>> before_append = id(python_list)
>>> python_list.append(5)
>>> python_list
[1, 2, 3, 4, 5]
>>> after_append = id(python_list)
>>> print(before_append, after_append)
(140635439293360, 140635439293360)
>>> print(before_append == after_append)
True
>>>
>>>
>>> import numpy as np
>>> np_array = np.zeros(shape=(1, 4))
>>> np_array
array([[ 0.,  0.,  0.,  0.]])
>>> before_resize = id(np_array)
>>> np_array = np.resize(np_array, (1, 5))
>>> np_array
array([[ 0.,  0.,  0.,  0.,  0.]])
>>> after_resize = id(np_array)
>>> print(before_resize, after_resize)
(139910114654128, 139910001813664)
>>> print(before_resize == after_resize)
False
>>>
```

Powyższa cecha obiektów biblioteki *NumPy* oznacza, że wykonywanie operacji na takich obiektach powinno być bardziej skuteczne pod względem czasu ich przeprowadzania. Jednakże wielokrotne przebudowywanie struktury obiektu wiąże się z bardzo dużym zużyciem pamięci, dlatego polecane jest stosowanie konwersji i tworzenie obiek-

tów dopiero w momencie, kiedy dane są skompletowane i gotowe do przetwarzania[19].

Biblioteka *SciPy* zbudowana jest na podstawie biblioteki *NumPy* i rozszerza ją o wiele algorytmów analizy danych. Elementami składowymi tej biblioteki są między innymi[19]:
- **cluster** - algorytmy klastrowania
- **linalg** - algebra liniowa
- **signal** - przetwarzanie sygnałów
- **stats** - funkcje i algorytmy statystyczne

Funkcjonalność biblioteki jest bardzo szeroka, dzięki czemu znajduje ona zastosowanie w wielu projektach, a także jest ona częścią składową inych bibliotek analitycznych języka Python. Przykładową metodą należącą do biblioteki *stats* jest *linregress*, która umożliwia przeprowadzenie regresji liniowej dla wskazanych danych.

```
>>> from scipy import stats
>>> import numpy as np
>>>
>>> data_x = np.random.random_integers(1, 99, 10)
>>> data_y = np.random.random_integers(1, 99, 10)
>>> data_x
array([72, 45, 69, 52, 93, 14, 80, 14, 13,  5])
>>> data_y
array([37, 90, 19,  7, 95, 89, 88, 94, 81, 19])
>>>
>>> slope, intercept, r_value,
    p_value, std_err = stats.linregress(data_x, data_y)
>>> print(slope, intercept, r_value, p_value, std_err)
(-0.033350664784966184, 63.424125380672955,
 -0.029541780200591877, 0.93543372355182242, 0.39896357644710517)
>>>
```

2.3. Pakiet Scikit-learn

2.3.1. Cel i przeznaczenie pakietu

Biblioteka *Scikit-learn* zawiera zestaw zaawansowanych narzędzi stosujących uczenie maszynowe do analizy danych w języku Python. Dystrybuowana jest na licencji BSD, która pozwala na modyfikowanie i rozprowadzanie kodu źródłowego, a nawet na włączanie go do produktów komercyjnych pod warunkiem unieszczenia w dokumentacji odpowiednich adnotacji dotyczących autorów. Dzięki temu zaliczana jest do wolnego oprogramowania, które rozwijane jest przez społeczność kontrybutorów. Większa część kodu stworzona jest bezpośrednio w języku Python, lecz niektóre elementy takie jak na przykład implementacje SVM oraz modeli liniowych oparte są na bibliotekach języka C++, odpowiednio LibSVM oraz LibLinear[21].

Podstawowym założeniem twórców biblioteki jest priorytetyzacja utrzymywania jakości i czytelności kodu, ponad implemenację bardzo wielu funkcji[21]. Dodatkowo

rozwijana jest wysokeij jakości kompleksowa dokumentacja, co razem stanowi bardzą dobrą bazę do rozwijania całego projektu przez wielu niezależnych deweloperów i wydawania stabilnych wersji produktu. Scikit-learn bazuje na trzech bibliotekach języka Python: *NumPy*, *SciPy* i *Matplotlib*, i stanowią one wymagania systemowe, niezbędne do poprawnego działania pakietu.

W pakiecie *Scikit-learn* algorytmy podzielone są na algorytmy uczenia z nadzorem oraz algorytmy uczenia bez nadzoru[22]. Pierwsze z nich operają się o podział danych na uczące i testowe, gdzie dane uczące zawierają przykładowe oczekiwane wartości na podstawie których budowany jest model. W zestawie algorytmów uczenia nadzorowanego znaleźć można między innymi[22]:
- Ogólne modele liniowe
- Liniową i kwadratową analizę dyskryminacyjną
- Regresję grzbietową (KRR)
- Maszynę wektorów nośnych (SVM)
- Algorytm k najbliższych sąsiadów
- Proces Gaussa
- Naiwny klasyfikator Bayesa
- Drzewa decyzyjne

Algorytmy uczenia bez nadzoru w pakiecie *Scikit-learn*, dla których dane uczące nie posiadają żadnych wartości odniesienia, możemy natomiast podzielić między innymi na:
- Klasteryzację
- Estymację kowariancji
- Nieliniową redukcję przestrzenną

W niniejszej pracy zastoswane zostały algorytmy uczenia z nadzorem, należące do grup: *Ogólne modele liniowe*, Uzupełnić!

2.3.2. Ogólne modele liniowe

W pakiecie *Scikit-learn* przedstawione zostały algorytmy regresji, w których oczekiwane wartości docelowe są liniową kombinacją wartości wejściowych. Podstawę stanowi równanie regresji:

$$\hat{y} = \omega_0 + \omega_1 X_1 + \omega_2 X_2 + ... + \omega_p X_k \tag{2.1}$$

Wektor $\omega = (\omega_1, ..., \omega_p)$ jest utożsamiany z parametrem *coef_*, a wyraz wolny ω_0 z parametrem *intercept_* [22].

2.3.2.1. Regresja liniowa

Regresja liniowa w pakiecie *Sciki-learn* dostępna jest poprzez obiekt **LinearRegression**. Parametry, jakie przyjmuje ten obiekt w momencie inicjalizacji to[22]:
- *fit_intercept*: boolean, optional, default True
- *normalize*: boolean, optional, default False
- *copy_X*: boolean, optional, default True
- *n_jobs*: int, optional, default 1

Obiekt udostępnia następujące metody:

- *fit*: dopasowanie modelu liniowego
- *get_params*: pobranie parametrów estymacji
- *predict*: predykcja na podstawie modelu liniowego
- *score*: współczynnik determinacji R^2
- *set_params*: ustawienie parametrów estymacji

2.3.3. Wybrane modele klasyfikacji i regresji

Rozdział 3

Przedstawienie aplikacji

3.1. Podstawowe założenia

3.2. Zastosowane pakiety języka Python

3.2.1. Kivy

3.2.2. Pandas i Matplotlib

3.2.3. Scikit-learn

3.3. Opis funkcjonalności aplikacji

3.3.1. Okno opcji

3.3.2. Okno analizy regresji

3.4. diagramy UML

Rozdział 4

Wyniki testów aplikacji

Rozdział 5

Wnioski

Bibliografia

[1] Tobiasz Maliński, *Giełda Papierów Wartościowych Dla Bystrzaków*, Helion 2016.

[2] Justin Kuepper, *Basics of Technical Analysis*, 19 Kwiecień 2017. https://www.investopedia.com/university/technical/

[3] Investopedia, *Stock Market*, 20 Listopad 2017. https://www.investopedia.com/terms/s/stockmarket.asp

[4] Prawo o publicznym obrocie papierami wartościowymi i funduszach powierniczych *Ustawa z dnia 22 marca 1991r.*, Art 2.

[5] Roman Ciepiela, Piotr Pytlik, Magda Wiernusz, *Encyklopedia Zarządzania* 25 Październik 2016. https://mfiles.pl/pl/index.php/Akcje

[6] Roman Ciepiela, Szymon Kułakowski, Sabina Blok, *Encyklopedia Zarządzania*, 13 Lipiec 2017 https://mfiles.pl/pl/index.php/Obligacje

[7] Małgorzata Łuniewska *Ekonometria Finansowa: Analiza rynku kapitałowego*, Warszawa 2008 Wydawnictwo Naukowe PWN

[8] Stockopedia *Technical Analysis (Part 1): History, Theory and Philosophy* Kwiecień 2017 https://www.stockopedia.com/content/technical-analysis-part-1-history-theory-and-philosophy-17959

[9] G.S. Maddala *Ekonometria* Warszawa 2006 Wydawnictwo Naukowe PWN

[10] Mariusz Czekała *Analiza fundamentalna i techniczna* Wrocław 1997 Wydawnictwo Akademii Ekonomicznej im. Oskara Langego we Wrocławiu

[11] Jacek Koronacki, Jan Ćwik *Statystyczne Systemy Uczące Się* Warszawa 2005 Wydawnictwa Naukowo-Techniczne

[12] Emil Lundkvist *Decision Tree Classication and Forecasting of Pricing Time Series Data* Stockholm 2014 Master's Degree Project

[13] Alex Smola *Introduction to Machine Learning* 2008 Cambridge University Press

[14] Paweł Cichosz *Systemy Uczące Się* Warszawa 2000 Wydawnictwo Naukowo-Techniczne Warszawa

[15] Mirosław Krzyśko *Systemy uczące się* Warszawa 2008 Wydawnicatwa Naukowo-Techniczne

[16] Stanisław M. Kot *Statystyka* Warszawa 2011 Difin SA

[17] Siegmund Brandt *Analiza Danych* Warszawa 1998 Wydawnictwo Naukowe PWN

[18] Fabrizio Romano *Learning Python* 2015 Packt
[19] NumPy Community *NumPy User Guide* Release 1.13.0
[20] SciPy.org *SciPy Documentation* https://docs.scipy.org/doc/scipy/reference/
[21] Fabian Pedregosa *Scikit-learn: Machine Learning in Python* Journal of Machine Learning Research 12 (2011
[22] scikit-learn.org *Sciki-learn Documentation* http://scikit-learn.org/stable/documentation.html

Spis rysunków

Spis tabel