Politechnika Warszawska

WYDZIAŁ MECHANICZNY
ENERGETYKI I LOTNICTWA

Instytut Techniki Cieplnej

Praca dyplomowa inżynierska

na kierunku Energetyka
w specjalności Zrównoważona energetyka

Modelowanie zapotrzebowania na energię z wykorzystaniem algorytmów uczenia maszynowego.

Kamil Cisak
Numer albumu 298931

promotor
dr inż. Mateusz Żbikowski

Warszawa, 2022

Streszczenie

Modelowanie zapotrzebowania na energię z wykorzystaniem algorytmów uczenia maszynowego.

Celem pracy inżynierskiej jest przedstawienie metod uczenia maszynowego wykorzystywanych w sektorze energetycznym. Przedstawione zostały niektóre możliwości zastosowania tego narzędzia, które można spotkać w literaturze. Opisane zostały również najważniejsze informacje na temat uczenia maszynowego. Stworzono projekt inżynierski, którego celem jest przewidywanie zapotrzebowania na energię w budynkach. W pracy omówiono cel projektu, użyte narzędzia, wynik analizy danych wykorzystanych do uczenia modeli oraz przedstawiono tabelę ukazującą skuteczność i czas trenowania poszczególnych modeli.

Słowa kluczowe: uczenie maszynowe, modele regresyjne, energetyka, prognozowanie energii

Abstract

Energy forecasting with machine learning algorithms

The purpose of this engineering thesis is to present machine learning methods used in the energy sector. Some possible applications of this tool that can be found in the literature are presented. The most important information about machine learning is also described. An engineering project was created to predict energy demand in buildings. The paper discusses the purpose of the project, the tools used, the result of the data analysis used to teach the models, and presents tables showing the effectiveness and training time of the different models.

Keywords: machine learning, regression models, power engineering, energy forecasting

Oświadczenie autora pracy

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.
Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płycie kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

20.01.2022 ..... Kamil Cisak .....
data podpis autora (autorów) pracy

Oświadczenie

Wyrażam zgodę / ~~nie wyrażam zgody~~*1 na udostępnianie osobom zainteresowanym mojej pracy dyplomowej. Praca może być udostępniana w pomieszczeniach biblioteki wydziałowej. Zgoda na udostępnienie pracy dyplomowej nie oznacza wyrażenia zgody na jej kopiowanie w całości lub w części.
Brak zgody nie oznacza ograniczenia dostępu do pracy dyplomowej osób:
- reprezentujących władze Politechniki Warszawskiej,
- członków Komisji Akredytacyjnych,
- funkcjonariuszy służb państwowych i innych osób uprawnionych, na mocy odpowiednich przepisów prawnych obowiązujących na terenie Rzeczypospolitej Polskiej, do swobodnego dostępu do materiałów chronionych międzynarodowymi przepisami o prawach autorskich.

Brak zgody nie wyklucza także kontroli tekstu pracy dyplomowej w systemie antyplagiatowym.

20.01.2022 ..... Kamil Cisak .....
data podpis autora (autorów) pracy

*1 - niepotrzebne skreślić

Spis treści

1. Wstęp

Dzięki popularyzacji zagadnienia inteligentnych budynków liczba dostępnych danych dotyczących zużycia energii stale rośnie. Dzięki modelowaniu zapotrzebowania na energię jesteśmy w stanie lepiej przygotować system elektroenergetyczny czy ciepłowniczy w celu sprostania zapotrzebowaniu. Prognozowanie zużycia energii również umożliwia porównanie zużycia energii po modernizacji budynku w celu określenia jego efektywności energetycznej. Dostępność dużych ilości danych pozwala wykorzystywać w tej dziedzinie algorytmy uczenia maszynowego.

Celem pracy inżynierskiej jest przyjrzenie się wykorzystaniu uczenia maszynowego w sektorze energetycznym, a w szczególności w celu prognozowania zużycia energii przez budynki. Praca dyplomowa składa się z pięciu rozdziałów. W drugim rozdziale został przeprowadzony przegląd literatury przedstawiający obecne osiągnięcia dotyczące zastosowania modeli uczenia maszynowego w usprawnianiu systemu elektroenergetycznego oraz rozwiązania jakie był stosowane w celach przewidywania zapotrzebowania budynków w energię. Kolejny rozdział przedstawia wiedzę teoretyczną związaną z zagadnieniem uczenia maszynowego. Zostają w nim omówione rodzaje uczenia maszynowego oraz przedstawione są w niej matematyczne opisy modeli wykorzystanych w projekcie inżynierskim. Omówiona zostaje również ewaluacja modeli. Czwarty rozdział został poświęcony omówieniu projektu inżynierskiego. Przedstawiono w nim technologie, jakie zostały użyte do przygotowania danych i szkolenia modeli. W pracy zostały wykorzystane popularne biblioteki dla języka Python używane podczas tworzenia modeli uczenia maszynowego czy w pracy z dużymi zbiorami danych. Do przygotowania modeli zostały wykorzystane gotowe algorytmy zaimplementowane w bibliotece MLlib oraz TensorFlow. W rozdziale zostały również przedstawione wyniki analizy parametrów wykorzystywanych do szkolenia. Przedstawione wyniki zawierają analizę rozkładów parametrów pogodowych oraz proces szukania punktów odstających w zbiorze danych wyjściowych. Została także wspomniana istotność normalizacji zmiennych wyjściowych w celu uzyskania skutecznych modeli. Na koniec zostały przedstawione wyniki skuteczności poszczególnych modeli oraz czas ich trenowania oparte na różnych algorytmach: algorytmie regresji liniowej, drzewa decyzyjnego, lasów losowych, wzmacniających drzew decyzyjnych oraz sieci neuronowych.

2. Zastosowanie uczenia maszynowego w sektorze elektroenergetycznym.

2.1 Zarys historyczny

Dostarczanie energii elektrycznej jest klasycznym tematem w elektrotechnice, natomiast sztuczna inteligencja jest dość nowym działem nauki, który pierwszy raz został zdefiniowany podczas konferencji w Dartmouth w 1956 roku przez Johna McCarthy.[1] Jednak jej rozwój rozpoczął się dopiero w latach 90. XX wieku. Dążąc do rozwoju systemów energetycznych w obecnych czasach dziedziny te zaczęły się ze sobą łączyć. [2]

Aktualnie dostęp do energii elektrycznej jest dobrem, bez którego nie wyobrażamy sobie codziennego życia. W roku 1950 zapotrzebowanie na energię dla Polski wynosiło ok. 10000 GWh, obecnie zużycie roczne kształtuje się na poziomie 165532 GWh.[3] Głównymi czynnikami wzrostu zapotrzebowania była rosnąca liczba ludności oraz wzrost ekonomiczny. Aby zapewnić powszechność dostaw dla obywateli i przedsiębiorstw konieczne było rozwijanie krajowego systemu energetycznego. Dzięki badaniom związanym z transportem energii elektrycznej o wysokich napięciach, dystans na jaki ta energia mogła być przesyłana wzrósł z kilku kilometrów w roku 1890 do tysięcy kilometrów w obecnych czasach.[4] Pozwoliło to na łączenie ze sobą sieci z różnych regionów.

Oprócz zmian związanych z zagadnieniami technicznymi, branża elektroenergetyczna przeszła poważne zmiany w zakresie zarządzania. W latach czterdziestych i pięćdziesiątych system elektroenergetyczny był scentralizowaną strukturą zarządzaną przez państwo. W roku 1997 w Polsce weszła w życie ustawa Prawo Energetyczne, której wprowadzenie przyjmuje się jako rozpoczęcie kształtowania rynku energii. Po tym wydarzeniu następuje demonopolizacja energetyki i rozdzielenie jej na podsektory. Obecnie podział nie uległ zmianie i wygląda następująco: wytwarzanie energii, przesył i dystrybucja energii oraz handel energią. Wprowadzenie konkurencyjności do sektora energetycznego daje nowe możliwości odbiorcom końcowym. Od tego momentu producenci i dostawcy zostali zmuszeni do optymalizacji kosztów i podwyższania wydajności, aby utrzymać przy sobie klienta.[5]

Skutkiem tych przemian było zwiększenie złożoności systemu. Aby utrzymać stabilność pracy i bezpieczeństwo dostaw energii elektrycznej konieczne było wdrożenie nowych systemów automatyzacji i monitoringu takich jak system zarządzania energią (EMS), system nadzoru sieci elektroenergetycznych (WindEx) i wiele innych. Obecnie system energetyczny składa się z wielu jednostek zautomatyzowanych i monitorowanych, dzięki temu jesteśmy w stanie pozyskiwać dane co sekundę. Ponieważ człowiek nie jest zdolny do tak szybkiego analizowania tak dużych zbiorów danych, co za tym idzie optymalnego podejmowania decyzji, wielu badaczy widzi w tym miejscu szanse na wykorzystanie modeli uczenia maszynowego, które mogłyby wspierać człowiek w podejmowaniu decyzji. Dzięki algorytmom uczenia maszynowego będziemy w stanie eksplorować dane i odkrywać informacje, które dla człowieka nie są oczywiste. Ponadto wykorzystując zawansowane technologie sztucznej inteligencji, jesteśmy w stanie przeszkolić modele, aby skutecznie same podejmowały istotne decyzje za człowieka.[2]

2.2 Praktyczne zastosowania uczenia maszynowego

Według raportu IPCC 2021 Ziemia stoi na granicy katastrofy klimatycznej, aby temu zapobiec konieczne jest odchodzenie od paliw kopalnych i zastępowanie ich innymi bez emisyjnymi źródłami energii.[6] Aby sprostać tym wyzwaniom w systemie pojawiają się nowe elementy tak, jak urządzenia zabezpieczające sieć, magazyny energii i nowe jednostki wytwórcze (panele fotowoltaiczne, turbiny wiatrowe). Zmiany te skutkują nowymi problemami związanymi ze stabilnością i bezpieczeństwem sieci energetycznej. Rozwiązanie nowych problemów wymaga zastosowania specjalistycznych systemów wykorzystujących nowoczesne technologie informatyczne dla automatyki i analizy systemu. [2]

Aktualnie algorytmy uczenia maszynowego w energetyce używane są w takich zagadnieniach, jak modelowanie zapotrzebowania na energię, wykrywanie anomalii oraz optymalizacja procesów. Jednak pojawia się coraz więcej artykułów naukowych przedstawiających ich nowe zastosowania.

2.2.1 Detekcja uszkodzeń elementów systemu linii przesyłowych

Szybkie rozpoznawanie uszkodzeń związanych z systemem przesyłowym jest ważną czynnością, która gwarantuje poprawność działania sytemu elektroenergetycznego. Zazwyczaj inspekcje przeprowadzane są w sposób tradycyjny, czyli jako patrole piesze, czasami wspierane wykorzystaniem helikopterów. Takie metody z reguły są czasochłonne, kosztowne oraz mogą stanowić zagrożenie. W ostatnich latach badacze z różnych instytucji badawczych pracują nad pełną automatyzacją tego procesu. Nowe metody analizy uszkodzeń opierają się na wykorzystaniu dronów do inspekcji linii przesyłowych, a następnie przetwarzanie obrazów przez nie zebranych, za pomocą algorytmów uczenia maszynowego takich jak: głębokie uczenie, regresja logistyczna, lasy losowe, maszyna wektorów nośnych. [7], [8]

2.2.2 Wykrywanie cyberataków

Technologie inteligentnych sieci bardzo szybko zyskują na popularności w obecnych systemach energetycznych. Z jednej strony technologie te zwiększają niezawodność systemu, umożliwiając szybszą kontrolę oraz ułatwiają przyłączenie rozproszonych źródeł energii. Z drugiej strony są narażone na cyberataki, przez to że są silnie zależne od technologii informacyjnych i komunikacyjnych. Cyberataki związane z pomiarami fazorów możemy podzielić na kilka kategorii: (1) Blokada usług, (2) Atak fizyczny, (3) Atak „Man-in-the-middle”, (4) Analiza pakietów, (5) Wprowadzenie szkodliwego oprogramowania, (6) Fałszowanie danych. Aby walczyć z ewoluującymi metodami cyberataków naukowcy w ostatnich latach pracowali nad różnymi metodami zwalczania tego typu zagrożeń. Jednak Xinan Wang w pracy pod tytułem „Machine learning applications in power systems” przedstawia nowatorskie podejście w tym zakresie. W swojej pracy przedstawił metodę opartą na eksploracji danych, która wykrywa i naprawia ataki związane z manipulacją danymi w systemie WAMS. Podstawowy algorytm opisany w pracy jest oparty na gęstości grupowania aplikacji z szumem.[2]

2.2.3 Prognozowanie prędkości wiatru

Prognozowanie prędkości wiatru jest ważnym zadaniem koniecznym do przewidywania zasobów energii wiatru. Dlatego coraz częściej w literaturze możemy spotkać artykuły wprowadzające nowe sposoby prognozowania szeregów czasowych w celu przewidywania prędkości wiatru. Prognozowanie prędkości wiatru w sposób liniowy jest złożonym zadaniem, ponieważ dane mają zależność stochastyczną i chaotyczną.[9] Dlatego modele przeprowadzają prognozy krótkoterminowe w przedziale czasowym od godzinny do kilku godzin naprzód. Taki sposób prognozy jest głównie wykorzystywany do rozliczeń na rynku energii, operacji sieciowych w czasie rzeczywistym oraz działań regulacyjnych.[10] W pracy przygotowanej przez Mahid Khodayar, Jianhui Wang i Mahammand Manthouri pod tytułem „Interval Deep Generative Neural Network for Wind Speed Forecasting" przedstawili hybrydowy model prognozowania oparty na głębokim uczeniu, teorii zbiorów przybliżonych i teorii zbiorów rozmytych.

2.3 Prognozowanie zapotrzebowania na energie w budynkach

Rozwój kraju i jego coraz większa urbanizacja wymaga budowania coraz większej ilości budynków, a co za tym idzie zwiększenia zapotrzebowania w różne formy energii. W cyklu swojego życia budynki są odpowiedzialne za znaczną produkcję dwutlenku węgla. Według raportu EIA odpowiadają za 67% zużycia energii na świecie i około 40 % w Unii Europejskiej.[11] Dlatego budynki powinny być projektowane z myślą o ograniczeniu zużycia energii, przy jednoczesnym zachowaniu ich komfortu. Z tego powodu temat ten jest popularnym zagadnieniem w rozważaniach akademickich dla sektora energetycznego.

2.3.1 Metody prognozowania zużycia energii

Coraz powszechniejsze staje się budowanie inteligentnych budynków, czyli takich które posiadają sieć czujników i mierników zarządzanych przez jeden system. Dzięki informacjom, które są z nich zbierane, budynek może przystosowywać się do warunków zmieniających się w jego wnętrzu, jak i na zewnątrz. Takie działanie umożliwia minimalizację kosztów potrzebnych do jego eksploatacji oraz ograniczają emisje szkodliwych zanieczyszczeń.[12]

Ważnym aspektem w działaniu takich budynków jest prognozowanie ich zużycia energii. Obecnie powszechnie stosuje się dwie metody. Jedna z nich oparta jest na metodach inżynieryjnych, natomiast druga wykorzystuje zagadnienia sztucznej inteligencji. Zastosowanie wiedzy inżynieryjnej zmusza do wykorzystywania czasochłonnych i specjalistycznych wzorów. Dodatkowo do poprawnego prognozowanie zużycia energii konieczne jest posiadanie szczegółowych informacji na temat obiektu. Metoda oparta na sztucznej inteligencji na swój sposób upraszcza nam zadanie. Wykorzystując dane historyczne do szkolenia modelu jesteśmy w stanie opracować model zapotrzebowania w energię.[13]

Pośród dużej liczby dostępnych modeli w uczeniu maszynowym popularnością cieszą się takie algorytmy jak sztuczne sieci neuronowe, regresja wektorów nośnych czy regresja liniowa. Dla przykładu w roku 2020 doktor Shashwat Ganguly przygotował model prognozujący zużycie energii w galerii sztuki historycznej. Do predykcji wykorzystał sztuczne sieci neuronowe.[14] Innym przykładem jest praca doktora Saleha Seyedzadeha,

który w swojej pracy oceniał skuteczność poszczególnych modeli wykorzystywanych do przewidywania energii potrzebnej do ogrzewania i chłodzenia budynków. W swoich badaniach zbudował modele oparte o takie algorytmy, jak sztuczne sieci neuronowe, regresje wektorów nośnych, lasy losowe oraz wzmocnione drzewa regresyjne. Porównując wyniki poszczególnych modeli za pomocą metryki błędu średniokwadratowego ustalił, że wzmocnione drzewa regresyjne wykazują najlepszą skuteczności przewidywania zapotrzebowania.[15]

2.3.2 Stan pracy w dziedzinie

Zagadnienie uczenia maszynowego w zagadnieniach związanych z analizą budynku po raz pierwszy zostało zastosowane w 1997 roku przez profesora Kalogirou w pracy „Building heating load estimation using artificial neural networks". Został w niej przedstawiony model oparty o sztuczne sieci neuronowe, którego zadaniem było przewidywanie zapotrzebowanie w energię cieplną.[16] Cztery lata później przedstawił kolejną pracę, w której przedstawił model przewidujący dobowe zapotrzebowanie w energię cieplną. Danymi wejściowymi do algorytmu były warunki pogodowe dla Cypru oraz różne typy ścian i dachów. W pracy tej profesor otrzymał wartości metryki ewaluacyjnej na poziomie 0.9885 i 0.9905 dla maksymalnego i minimalnego zapotrzebowania. Wykazał w ten sposób, że uczenie maszynowe może być wykorzystywane w modelowaniu zapotrzebowania w różnych typach konstrukcji.[17]

W 2005 roku pierwszy raz została zaproponowana metoda prognozowania zużycia energii z wykorzystaniem algorytmu maszyn wektorów nośnych. Profesor Dong w swojej pracy przedstawił model przewidujący zapotrzebowanie w budynkach, które znajdują się w regionie tropikalnym. Wykorzystał w tym celu dane wygenerowane przez cztery budynki z Singapuru. Ostatecznie udało mu się osiągnąć współczynnik wariancji poniżej 3 % oraz błąd procentowy na poziomie 4%.[18]

Modele oparte o algorytm lasów losowych i wzmocnionych drzew decyzyjnych są znanym zagadnieniem w dziedzinie uczenia maszynowego już od dłuższego czasu. Jednak swoje pierwsze zastosowanie w analizę energii budynków znalazło dopiero w 2012 roku. Badacze Tsansa i Xifara wykorzystali lasy losowe w celu predykcji zapotrzebowania na energię potrzebną do ogrzewania i chłodzenia budynku. Badali w swojej pracy w wpływ ośmiu zmiennych na dwie zmienne wyjściowe. W efekcie ustalili, że narzędzie jakim jest uczenie maszynowe jest wygodnym i dokładnym sposobem w analizie energetycznej budynków.[19]

Podsumowując, na przestrzeni XXI wieku powstało wiele przykładów prac naukowych, w których badacze wykorzystują modele uczenia maszynowego do rozwiązania problemu jakim jest przewidywanie zapotrzebowania na energię przez budynki. Jednak przykłady przedstawione powyżej, jak i wiele innych prac w swoich analizach przedstawiają wykorzystanie jednego algorytmu lub zestawienie ze sobą dwóch różnych. Jednak brakuje artykułów, w których naukowcy przedstawiają metody optymalizacji modeli w celu osiągnięcia najlepszej skuteczności. [15] Powstała luka w ostatnich latach zaczęła się zapełniać i przykładem jest praca wcześniej już wspomnianego doktora Seyedzadeha. W najbliżej przyszłości można przepuszczać, że nowe prace pojawiające się w tej dziedzinie będą skupiały się wokół poprawy skuteczności modeli.

3. Uczenie maszynowe

3.1 Wprowadzenie

Uczenie maszynowe jest elementem sztucznej inteligencji, który daje komputerom możliwość uczenia się. Algorytmy uczenia maszynowego wykorzystują statystykę do eksploracji i rozpoznawania wzorów w dużych ilościach danych. Informacje, które możemy wprowadzić do algorytmów, obejmują wiele kategorii takich jak liczby, słowa, obrazy oraz inne dane, które można przechowywać na dyskach. Obecnie z tej dziedziny nauki, korzysta wiele dużych firm takich jak Netflix, Google, Facebook czy Twitter.[20] Wykorzystują algorytmy do ulepszania swoich produktów. Tak jak zostało to przedstawione w poprzednich rozdziałach uczenie maszynowe zyskuje również na popularności w sektorze energetycznym. W projekcie inżynierskim algorytmy uczenia maszynowego zostały wykorzystane do przewidywania zapotrzebowania w różne formy energii przez budynki.

Pierwszym krokiem koniecznym do wytrenowania modelu jest wprowadzenie danych wejściowych oraz prawdopodobnych wyników. W ten sposób umożliwiamy algorytmowi poznanie wzorców, które powodują powstawanie poszczególnych danych wejściowych. Tak przeszkolony model jest gotowy do wyciągnięcia wniosków. Wprowadzając do niego dane, których jeszcze nie widział tzw. Dane testowe możemy sprawdzić efektywność działania naszego modelu.

3.2 Rodzaje uczenia

Badacze wykorzystują różne algorytmy w celu uzyskania satysfakcjonujących wyników, ale możemy je przydzielić do jednej z trzech kategorii bazujących na typie uczenia: (1). Uczenie nadzorowane, (2). Uczenie nienadzorowane, (3). Uczenie przez wzmacnianie. W praktyce występuje jeszcze jeden typ – uczenie częściowo nadzorowane. W rzeczywistych danych mogą pojawiać się szumy czy braki informacji, co tworzy lukę pomiędzy uczeniem nadzorowanym, a nienadzorowanym.[21]

3.2.1 Uczenie nadzorowane

Zadaniem uczenia nadzorowanego jest wyznaczenie funkcji *h*, która przybliża rzeczywistą funkcję *f*. Aby wyszkolić model dostarczamy mu treningową bazę danych składający się z *N* elementowego zbioru wejść-wyjść:

$$(x_1, y_1), (x_2, y_2), \dots (x_N, y_N), \tag{2.1}$$

gdzie każda y_j wartość została wygenerowana przez nieznaną funkcje $y = f(x)$. Uczenie nadzorowane najczęściej jest wykorzystywane przy zagadnieniach związanych z klasyfikacją lub regresją. W przypadku klasyfikacji algorytm przewiduje dyskretne odpowiedzi, natomiast dla regresji przewiduje liczby rzeczywiste.[21]

3.2.2 Uczenie nienadzorowane

W uczeniu nienadzorowanym nie wskazujemy modelowi czego powinien się nauczyć. W tym przypadku algorytm sam ma znaleźć wzorce na podstawie dostarczonych danych. Baza danych w tym przypadku składa się tylko z *N* wejść. Uczenie nienadzorowane

najczęściej wykorzystywane do grupowania podobnych przykładów czy wykrywanie anomalii.[22]

3.2.3 Uczenie przez wzmacnianie

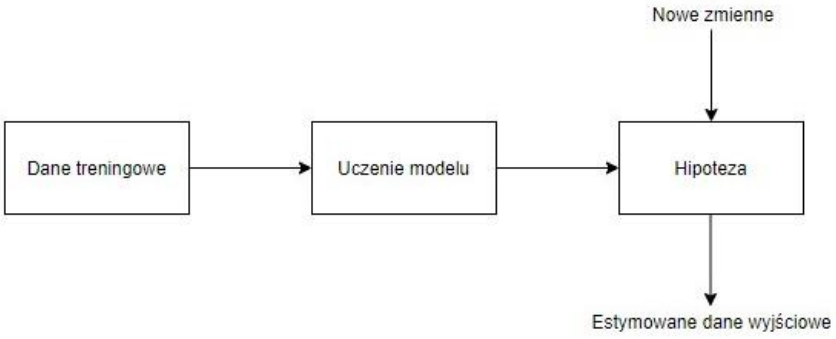
Dla wielu problemów, w których musimy podejmować decyzje lub rozwiązać problem ze sterowaniem, ciężko jest zapewnić jawny nadzór, jak dla algorytmów uczenia nadzorowanego. Z tego powodu w tego typu modelach stosujemy funkcję nagrody, która ma wskazywać czy czynności wykonywane przez algorytm są prawidłowe. Dla przykładu można podać uczenie czworonożnego robota chodzenia. Kiedy robot przemieszcza się w przód funkcja nagrody daje mu pozytywną nagrodę, gdy czwórnóg przewraca się lub cofa funkcja daje negatywną nagrodę. Uczenie to znalazło swoje zastosowanie w autonomicznych lotach helikopterem, autonomicznych robotach, kontroli procesów w fabrykach i wielu innych. [23]

3.2.4 Uczenie częściowo nadzorowane

Ten typ uczenia występuje, gdy posiadamy dużą ilość danych wejściowych, ale tylko niektóre z nich posiadają dane wyjściowe. Przykładem jest archiwum zdjęć, gdzie tylko niektóre z nich są podpisane (np. pies, kot, człowiek), a pozostałe z nich są nieoznaczone. Większość problemów obecnie mieści się w tym zakresie. Wynika to z niższego kosztu danych nieoznaczonych oraz większej dostępności.[24]

3.3 Algorytmy uczenia maszynowego

Predykcja energii zużywanej w budynkach wykorzystuje dane wejściowe oraz wyjściowe do wyszkolenia modelu. W tej sytuacji mamy do czynienia z algorytmami uczenia nadzorowanego. Na Rys. 2.1 został przedstawiony uproszczony diagram działania takich algorytmów.[25] W podrozdziale zostały omówione popularne algorytmy wykorzystywane w tego typu uczeniu.



Rys. 3.1. Uproszczony diagram działania algorytmów uczenia nadzorowanego.

3.3.1 Regresja liniowa

Regresja liniowa jest najstarszym algorytmem znanym od ponad 200 lat. W literaturze możemy spotkać się z różnym nazewnictwem tego modelu, jednak niezależnie od nazwy zmienna wyjściowa *Y* może być obliczona jako liniowa kombinacja zmiennych wejściowych

X. Gdy mamy do czynienia z jedną zmienną, model nazywamy prostą regresją liniową. W sytuacji wielu zmiennych metoda nazywa się wielokrotną regresją liniową.[24]

Regresja jest popularnym algorytmem ze względu na swoją prostotę. Możemy ją przedstawić jako równanie liniowe o określonej ilości danych wejściowych, którego rozwiązaniem jest wartość wyjściowa. Hipotezę tego modelu możemy przedstawić następująco[25]:

$$h(x) = \sum_{j=0}^{n} \theta_j x_j, \qquad gdzie\ x_0 = 1 \tag{2.2}$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \dots \\ \theta_n \end{bmatrix} \tag{2.3}$$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \tag{2.4}$$

$$\theta - macierz\ parametrów \qquad x - macierz\ zmiennych \qquad x_0 - wyraz\ wolny$$

Następnie konieczne jest zdefiniowanie funkcji straty, którą maszyna będzie wykorzystywać do optymalizacji algorytmu. Jednym z powszechnych sposobów jest wykorzystanie funkcji błędu średniokwadratowego[25]:

$$\min_{\theta} J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (h_\theta(x^{(i)} - y^{(i)})^2 \tag{2.5}$$

Optymalizacja odbywa się poprzez iteracyjne minimalizowanie błędu modelu. W tym celu wykorzystujemy metodę gradientu prostego. Proces jest powtarzany aż do osiągnięcia minimalnego błędu lub do momentu, gdy poprawa wartości będzie niemożliwa[25].

3.3.2 Drzewa decyzyjne

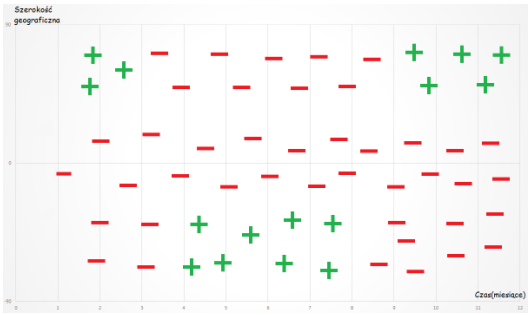
Drzewa decyzyjne są jednym z ważniejszych algorytmów predykcyjnych w uczeniu maszynowym. Klasyczne modele drzew istnieją od dziesięcioleci, a obecnie są rozwijane w nowoczesnych odmianach takich jak lasy losowe, które zaliczamy do jednych z lepszych algorytmów uczenia maszynowego.[24]

Drzewa decyzyjne charakteryzują się nieliniowością w porównaniu do algorytmu maszyny wektorów nośnych czy ogólnych modeli liniowych. Oznacza to, że funkcja nie może zostać sprowadzona do postaci jak poniżej:

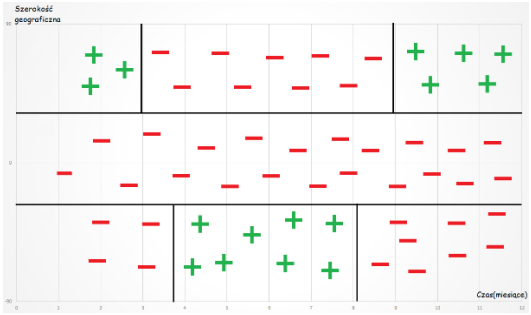
$$h(x) = \theta^T x, \qquad gdzie\ \theta \in R^n \tag{2.6}$$

Dodatkowo, jeśli metoda może generować nieliniowe funkcje hipotez, również nazywamy ją nieliniową. Jedną z metod osiągnięcia nieliniowej funkcji hipotezy jest kernelizacja metody liniowej poprzez mapowanie cech $\phi(x)$. Jednak drzewa decyzyjne są w stanie generować nieliniowe funkcje hipotez bez konieczności odpowiedniego mapowania cech.[26]

Dla przykładu chcemy zbudować model klasyfikacyjny, który na podstawie dwóch danych wejściowych: lokacji oraz czasu będzie przewidywał, czy można w danym miejscu jeździć na nartach. Poniżej na Rys. 2.2 został przedstawiony zestaw danych (lokalizacja jest reprezentowana jako szerokość geograficzna, a czas jest reprezentowany jako miesiące).



Rys. 3.2. Zbiór danych



Rys. 3.3 Przykładowy podział wykresu

Jak widać nie ma linii, która mogłaby prawidłowo podzielić przedstawione dane. Jednak możemy rozpoznać kilka miejsc, w których występują tylko wartości pozytywne lub negatywne. Przykładowy podział został przedstawiony na Rys. 2.3. Osiągamy to dzieląc przestrzeń X na regiony R_i [26]:

$$X = \bigcup_{i=0}^{n} R_i, \quad takie\ że\ R_i \cap R_j\ dla\ i \neq j \quad gdzie\ n \in \mathrm{Z}^+ \tag{2.7}$$

W przypadku regresji przewidywana wartość przez model jest średnią wszystkich wartości znajdujących się w regionie R[26]:

$$\hat{y} = \frac{\sum_{i \in R} y_i}{|R|} \tag{2.8}$$

W celu optymalnego dobrania podziału wykorzystujemy stratę kwadratową[26]:

$$L(R) = \frac{\sum_{i \in R} (y_i - \hat{y})^2}{|R|} \tag{2.9}$$

Istotną kwestią podczas szkolenia modelu jest również określenie kryteriów zatrzymujących dzielenie w naszym algorytmie. Najprostszym kryterium jest tzw. kryterium w pełni rosnącego drzewa. Kontynuujemy dzielenie regionów do momentu, w którym każdy region będzie zawierał dokładnie jeden punkt treningowy. Jednak taka technika charakteryzuje się wysoką wariancją oraz niskim obciążeniem modelu, dlatego taki model często może być przetrenowany. Dlatego w praktyce wykorzystujemy inne techniki regulacyjne[26]:

- **Minimalny rozmiar regionu** – nie dzielimy *R,* jeśli jego rozmiar spadł poniżej pewnej ustalonej wartości
- **Maksymalna głębokość** – nie dzielimy *R,* jeśli osiągnięto już pewną ustaloną ilość podziałów
- **Maksymalna liczba węzłów** – zatrzymujemy algorytm, jeśli drzew ma więcej liści niż ustalony próg

3.3.3 Lasy losowe

W poprzednim podrozdziale została poruszona istotna kwestia związana z drzewami decyzyjnymi. Mianowice chodzi o ich podatności na przeuczenie wynikająca z wysokiej wariancji oraz niskiego obciążenia. Jednym ze sposobów radzenia sobie z takim niebezpieczeństwem jest budowanie zespołu drzew. W roku 2001 Leo Breiman wprowadził jeden z takich algorytmów pod nazwą lasy losowe.[27]

Algorytm treningowy dla lasów losowych opiera się na technice agregacji bootstrapowej, inaczej nazywanej workowaniem. Jest to tradycyjna metoda w statystyce, która umożliwia redukcje wariancji.[26] Aby objaśnić działanie tej techniki załóżmy, że mamy pewną populację P. Z danej populacji wyodrębniamy zestaw treningowy S. Następnie tworzymy podzbiory Z_1, Z_2, …, Z_M wybierając losowo dane z zestawu treningowego. W tym momencie pojawia się pewna różnica, która występuje w algorytmie lasów losowych. Podczas szkolenia algorytmów G_m nie będziemy wykorzystywali wszystkich parametrów. Dla każdego drzewa wybieramy losowo określoną ilość parametrów i tylko je wykorzystujemy do szkolenia modeli. Aby otrzymać predykcję na podstawie nowych danych wejściowych, wprowadzamy do każdego modelu nowe dane i w ten sposób otrzymujemy M predykcji $G_m(x)$. Obliczając średnią z otrzymanych wyników uzyskujemy predykcje zagregowaną G(X).[26]

Powyższe zabiegi mają na celu zmniejszenie wariancji modelu, której wzór można przedstawić w poniższy sposób:

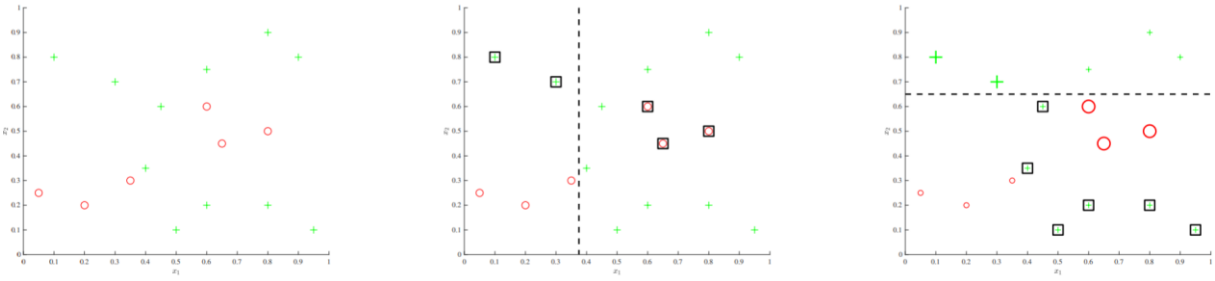
$$Var(\bar{X}) = \rho\sigma^2 + \frac{1-\rho}{M}\sigma^2 \tag{2.10}$$

Dzięki zastosowaniu metody lasów losowych otrzymujemy mniej skorelowane przewidywania, co za tym idzie obniżamy współczynnik korelacji ρ. Dodatkowo współczynnik korelacji nie jest zależny od liczby modeli M, co oznacza, że przy wzroście ilości modeli wariancja może tylko maleć. Metoda ta również niesie ze sobą wady, a mianowicie brak dostępności pełnego zestawu danych dla algorytmów powoduje zwiększenie obciążenie modelu. Jednak naukowcy stwierdzili, że spadek wariancji jest znaczniej większy niż wzrost obciążenia.

3.3.4 Wzmocnione drzewa decyzyjne

Jak zostało to wcześniej omówione workowanie jest techniką umożliwiając redukcję wariancji, natomiast podbijanie (z ang. boosting) jest znanym sposobem obniżania obciążenia. W metodzie tej modele o wysokim obciążeniu i niskiej wariancji nazywamy słabymi uczniami. Ogólną zasadę działania przedstawia Rys. 2.4. Na początku mamy oryginalny zestaw danych na rysunku przedstawiony z lewej strony. Następnie dokonujemy

jednego podziału, jak na środkowym wykresie oraz określamy, które obserwację są ciężkie do sklasyfikowania i podnosimy ich wagę, obniżając również wagę obserwacji, które były prawidłowo sklasyfikowane. Kolejnym krokiem jest zbudowanie nowego drzewa opartego na ważonych danych, co przedstawia prawy wykres przedstawiony na Rys. 2.4. Taki algorytm powtarzamy określoną ilość razy, a na końcu otrzymujemy kombinację słabych uczniów sklasyfikowaną na podstawie ich wag.[26]



Rys. 3.4. Przykład działania metody podbijania
Źródło: https://www.youtube.com/watch?v=wr9gUr-eWdA

Jednym z popularniejszych metod podbijania jest algorytm Adaboost. W tym podejściu wykonujemy iteracje podczas, których następuje trenowanie, a następnie pomiar błędu ważonego err_m słabych uczniów oraz obliczenie wagi α_m według poniższych wzorów[26]:

$$err_m = \frac{\sum_i w_i \, 1(y_i \neq G_m(x_i))}{\sum w_i} \tag{2.11}$$

$$\alpha_m = \log(\frac{1 - err_m}{err_m}) \tag{2.12}$$

Algorytm wykonujący kolejne iteracje zwiększa wagę słabych uczniów, aby model kładł większy nacisk na źle sklasyfikowane dane[26].

W rzeczywistości nie jest tak łatwo zaimplementować powyższy algorytm. Z pomocą przychodzi algorytm XGboost oparty o optymalizacje numeryczne. W podbijaniu gradientowym podczas każdego treningu obliczamy gradient w odniesieniu do obecnego przewidywania:

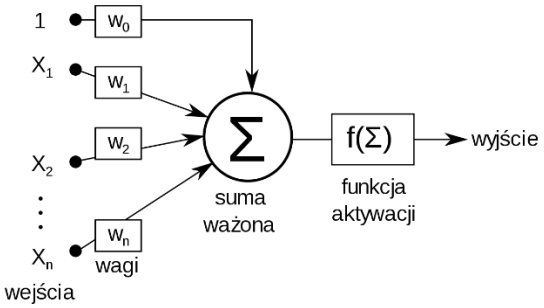
$$g_i = \frac{\partial L(y, f(x_i))}{\partial f(x_i)} \tag{2.13}$$

Kolejnym krokiem jest wytrenowanie nowego modelu, tak aby dopasować do niego gradient i wykorzystać go jako krok gradientu.[26] Podsumowując zastosowanie metody podbijania obniża obciążenie modelu oraz podnosi jego skuteczności. Jednak sposób ten zwiększa wariancje naszego modelu, co naraża na zbytnie dopasowanie do danych uczących.

3.3.5 Sztuczne sieci neuronowe

Sztuczne sieci neuronowe są metodą uczenia maszynowego opartą o działanie sieci neuronowych w mózgu. W uproszczeniu mózg złożony jest z dużej liczby neuronów połączonych między sobą w sieciach komunikacyjnych. W ten sposób mózg może wykonywać złożone operacje. Sztuczne sieci neuronowe zbudowane są zgodnie z tą metodyką.[27]

Najprostsza sieć neuronowa, którą można skonstruować składa się tylko z pojedynczego neuronu, który można nazywać perceptronem. Graficznie możemy przedstawić model jak na Rys. 2.5.



Rys. 3.5. Schemat neuronu McCullocha-Pittsa
Źródło: https://pl.wikipedia.org/w/index.php?title=Neuron_McCullocha-Pittsa&oldid=55296103

Dane wejściowe są przekazywane do neuronu jako suma ważona, gdzie wagi są ustalane w porównaniu do innych danych wejściowych. Aby uprościć zapis do danych wejściowych dodajemy jeden dodatkowy wymiar o stałej wartości, który reprezentuje wyraz wolny. Obliczenie sumy można przedstawić następującym wzorem:

$$z = \sum_i w_i x_i \tag{2.14}$$

Po obliczeniu sumy jesteśmy w stanie określi wartość wyjściową z neuronu jako wartość funkcji aktywacji f(z) zastosowanej w modelu. W literaturze zostało zdefiniowanych wiele funkcji, które mają różne przeznaczenia. Jednak do powszechnie używanych zaliczamy trzy funkcje[28]:

- **Funkcja sigmoidalna**

$$f(z) = \frac{1}{1 + e^{-z}} \tag{2.15}$$

- **Funkcja liniowa tzw. ReLu**

$$f(z) = \max(z, 0) \tag{2.16}$$

- **Funkcja tangensoidalna**

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{2.17}$$

Następnym krokiem podczas szkolenia modelu jest poprawianie parametrów. Po iteracji sieci neuronowej wyjście jest predykcją $\hat{y}$. W tym celu definiujemy funkcje kosztu J [28]:

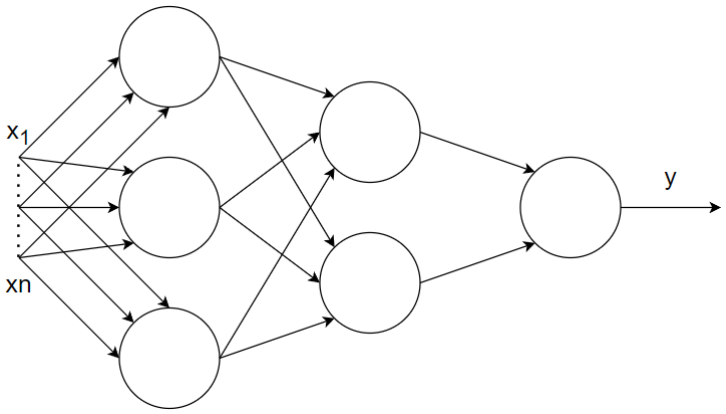
$$J(\hat{y}, y) = \frac{1}{m} \sum_{i=1}^{m} Z(\hat{y}, y) \tag{2.18}$$

$$\text{Gdzie } Z(\hat{y}, y) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y}) \tag{2.19}$$

Po obliczeniu funkcji kosztu, kolejnym krokiem jest aktualizacja parametrów we wszystkich warstwach. Dla dowolnej warstwy *l* nowe parametry możemy obliczyć poniższym wzorem[28]:

$$w^{[l]} = w^{[l]} - \alpha \frac{\partial J}{\partial w^{[l]}} \tag{2.20}$$

Przykładową sieć neuronową posiadającą więcej neuronów przedstawiono na Rys. 2.6. Pierwsza warstwa w sieci nazywa się warstwą wejściową, warstwa środkowa nazywana jest warstwą ukrytą ze względu na brak bezpośredniego połączenia z danymi wejściowymi i wyjściowymi modelu, ostatnia skrajna warstwa to warstwa wyjściowa odpowiedzialna za dane wyjściowe z modelu.[29]



Rys. 3.6. Przykładowa dwuwarstwowa sieć neuronowa

3.4 Ewaluacja modelu

3.4.1 Sposoby walidacji modelu

Ważną rzeczą podczas szkolenia modelu jest nietrenowanie go na całym zbiorze danych. Koniecznym jest podzielenie naszego zbioru danych na dane treningowe oraz dane testowe. Podczas tworzenia modelu chcemy, aby działał on równie dobrze z nowymi danymi, co oznacza, że chcemy uniknąć nadmiernego dopasowania do zestawu uczącego. Typowo w projektach wykorzystuje się 70% danych do trenowania modelu oraz 30% do jego testowania.[30]

Częstym zabiegiem podczas tworzenia modeli uczenia maszynowego, jest również podział danych w trzy podzbiory: treningowy, walidacyjny oraz testowy. Zbiór treningowy jest wykorzystywany do szkolenia modelu, zbiór walidacyjny umożliwia wybór najlepszej architektury modelu, ale w ten sposób narażamy się na zbytnie dopasowanie danych. Dlatego wykorzystujemy zbiór testowy, żeby ocenić jak najlepszy model dział na nowych danych.[30]

W zagadnieniach, w których posiadamy ograniczoną ilość danych, wskazane jest, aby wykorzystać jak największą ilość danych do uczenia. Z drugiej strony nadal potrzebujemy określonej ilości danych do oceny modelu. W tym celu możemy wykorzystać walidację krzyżową. W tym sposobie walidacji dzielimy nasz zbiór danych na K podzbiorów (w praktyce dzielimy główny zbiór na 10 podzbiorów). Następnie szkolimy K różnych modeli, gdzie jeden podzbiór jest wykorzystywany jako testowy, a pozostałe są wykorzystywane do trenowania algorytmu. Aby otrzymać ostateczny wynik, uśredniamy wartości z wszystkich szklonych modeli. Sposób ten wymaga więcej czasu do osiągnięcia wyniku końcowego, ale znacznie poprawia dokładność algorytmu.[30]

3.4.2 Regresyjne metryki ewaluacyjne

Regresja jest rozwiązywaniem problemów predykcyjnych, gdzie przewidujemy wartości liczbowe. W przeciwieństwie do metryk klasyfikacyjnych nie możemy użyć metod oceny skuteczności. Zamiast tego używamy metryk błędu, które oceniają jak blisko jest naszej predykcja do wartości rzeczywistej. W praktyce najczęściej wykorzystywane są trzy metryki błędu[31]:

- **Błąd średniokwadratowy –** jest powszechnie wykorzystywany, ponieważ jest niewrażliwy na to czy wartość przewidywana była za wysoka lub za niska.

$$MSE(y, \hat{y}) = \frac{1}{n}\sum(y - \hat{y})^2 \tag{2.21}$$

- **Pierwiastek błędu średniokwadratowego –** często wykorzystywany jest w zagadnieniach uczenia głębokiego. Zaletą metryki jest otrzymanie wartości wyjściowej w tej samej jednostce co zmienne. Umożliwia to łatwiejszą interpretację straty.

$$RMSE(y, \hat{y}) = \sqrt{\sum\frac{(y - \hat{y})^2}{n}} \tag{2.22}$$

- **Średni błąd bezwzględny –** metryka ta jest bardzo podobna do błędu średniokwadratowego, z taką różnicą, że wartość błędu nie jest podnoszona do kwadratu. Dzięki temu również otrzymujemy wartość błędu w takiej samej jednostce jak zmienne.

$$MAE(y, \hat{y}) = \frac{1}{n}\sum|y - \hat{y}| \tag{2.23}$$

4. Projekt inżynierski

4.1 Cel projektu

Celem projektu inżynierskiego było przygotowanie i porównanie skuteczności pięciu modeli uczenia maszynowego opartych o algorytmy opisane w poprzednim rozdziale. Prognozowanie dotyczyło czterech form energii w postaci wody lodowej, energii elektrycznej, ciepłej wody oraz pary.

4.2 Użyte technologie

4.2.1 Pandas

Pandas jest otwartą biblioteką przygotowaną dla języka Python, która umożliwia przetwarzanie i analizowanie danych. Pakiet ten umożliwia wczytywanie różnych różnego typu plików takich jak CSV, JSON, zapytań baz danych SQL czy XLSX.[32] W projekcie inżynierskim została użyta funkcja *read_csv()*, która umożliwia wczytanie danych do ramki danych(z ang. DataFrame). Na rysunku Rys. 4.1 została przedstawiona ramka danych dla treningowych danych pogodowych. Biblioteka dostarcza również szereg funkcji matematycznych i statystycznych umożliwiających lepsze zrozumienie wczytanych danych. Przykładowo pakiet dostarcza funkcje *.median()*, która umożliwia obliczenie mediany z analizowanych danych.

	site_id	timestamp	air_temperature	cloud_coverage	dew_temperature	precip_depth_1_hr	sea_level_pressure	wind_direction	wind_speed
0	0	2016-01-01 00:00:00	25.0	6.0	20.0	NaN	1019.7	0.0	0.0
1	0	2016-01-01 01:00:00	24.4	NaN	21.1	-1.0	1020.2	70.0	1.5
2	0	2016-01-01 02:00:00	22.8	2.0	21.1	0.0	1020.2	0.0	0.0
3	0	2016-01-01 03:00:00	21.1	2.0	20.6	0.0	1020.1	0.0	0.0
4	0	2016-01-01 04:00:00	20.0	2.0	20.0	-1.0	1020.0	250.0	2.6

Rys. 4.1 Ramka danych dla treningowych danych pogodowych
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/ML%20models.ipynb

4.2.2 NumPy

NumPy jest kolejną biblioteką opracowaną dla języka Python wykorzystaną w projekcie inżynierskim. Umożliwia przetwarzanie wielowymiarowych tablic i macierzy. Ze względu na swoją wysoką wydajność i wygodę zastępuje standardową strukturę danych Pythona jakim jest tablica.[33] Pakiet ten zapewnia również szereg funkcji umożliwiających operacje na tablicach, w tym matematyczne, logiczne, sortowania, wybierania i wiele innych.

4.2.3 Matplotlib

Matplotlib jest popularną biblioteką języka Python wykorzystywaną do wizualizacji danych. Pakiet umożliwia tworzenie wykresów słupkowych, punktowych, histogramów oraz wiele innych. Na Rys. 4.2 został przedstawiony przykładowy wykres kołowy wygenerowany za pomocą pakietu.

Rys. 4.2 Przykładowy wykres kołowy
Źródło: https://www.w3schools.com/python/matplotlib_pie_charts.asp (dostęp sty. 05, 2022).

4.2.4 Apache Spark

Apache Spark jest narzędziem umożliwiającym szybkie przetwarzanie dużych zestawów danych, daje również możliwość rozdzielenia procesu przetwarzania danych pomiędzy kilkoma komputerami. Te dwie właściwości są istotne w branży big data i uczenia maszynowego, które wymagają dużych mocy obliczeniowych. W projekcie inżynierskim wykorzystywane jest API przygotowane specjalnie dla języka Python o nazwie PySpark. Biblioteka PySpark zawiera w sobie również narzędzia uczenia maszynowego, które umożliwiają tworzenie modeli dla zagadnień regresyjnych. W projekcie inżynierskim została wykorzystana do tworzenia modeli.

4.3 Źródło danych

Dane wykorzystane w projekcie inżynierskim zostały udostępnione przez Amerykańskie Stowarzyszenie Inżynierów Ogrzewnictwa, Chłodnictwa i Klimatyzacji w konkursie organizowanym na stronie Kaggle. Zbiór danych można odnaleźć pod adresem https://www.kaggle.com/c/ashrae-energy-prediction/overview. Organizacja udostępniła zainteresowanym pięć plików, w których zostały zebrane informacje o ponad 1000 budynków z okresu trzech lat.

4.3.1 Dane o budynkach

Pierwszy z udostępnionych plików o nazwie building_meta.csv zawiera podstawowe informację o budynkach, w których prowadzone były pomiary. Plik zawiera informację o 1449 budynkach opisanych sześcioma zmiennymi:

- **site_id –** identyfikator umożliwiający zestawienie między sobą danych o budynkach z danymi pogodowymi
- **building_id –** identyfikator umożliwiający zestawienie danych o budynkach z danymi odczytanymi z liczników
- **primary_use –** parametr określający kategorie użyteczności danego budynku np. edukacyjna, przemysłowa, biurowa
- **square_feet –** powierzchnia budynku
- **year_built –** rok oddania budynku do użytku
- **floor_count –** liczba pięter budynku

4.3.2 Dane pomiarowe mierników

Dane odczytane z mierników zostały zebrane i podzielone na dwa pliki: train.csv i test.csv. Pierwszy plik zawiera 20 216 100 rekordów, które w projekcie są wykorzystywane do szkolenia modeli. Drugi plik zawierający 41 697 600 danych został wykorzystany w celu walidacji modeli i porównania ich między sobą. Obydwa pliki zawierają po cztery kolumny:

- **building_id –** identyfikator umożliwiający zestawienie danych z odczytów z danymi o budynkach
- **meter –** parametr określający rodzaj pomiaru, budynki mogą mieć zamontowane cztery typy mierników (miernik elektryczny, wody lodowej, pary, ciepłej wody)
- **timestamp –** okres, w jakim był prowadzony pomiar
- **meter_reading –** odczytana wartość podczas pomiaru (tylko dla danych treningowych)
- **row_id –** identyfikator umożliwiają ustawienie predykcji w właściwej kolejności (tylko dane testowe)

4.3.3 Dane pogodowe

Ostatnimi dwoma plikami, które znalazły się w udostępnionej bazie danych są informacje na temat warunków pogodowych. Informacje te również zostały podzielone na dwa pliki: weather_train.csv oraz weather_test.csv. Dane treningowe zawierają 139 773 odczyty, a dane testowe 277 243. Parametry opisujące pogodę prezentują się następująco:

- **site_id –** identyfikator umożliwiający zestawienie między sobą danych pogodowych z danymi o budynkach
- **air_temperature –** temperatura zewnętrzna powietrza
- **cloud_coverage –** poziom zachmurzenia w skali okta
- **dew_temperature –** temperatura punktu rosy
- **precip_depth_1_hr –** poziom opadów atmosferycznych
- **sea_level_pressure –** ciśnienie powietrza
- **wind_direction –** kierunek wiatru
- **wind_speed –** prędkość wiatru

4.4 Analiza danych

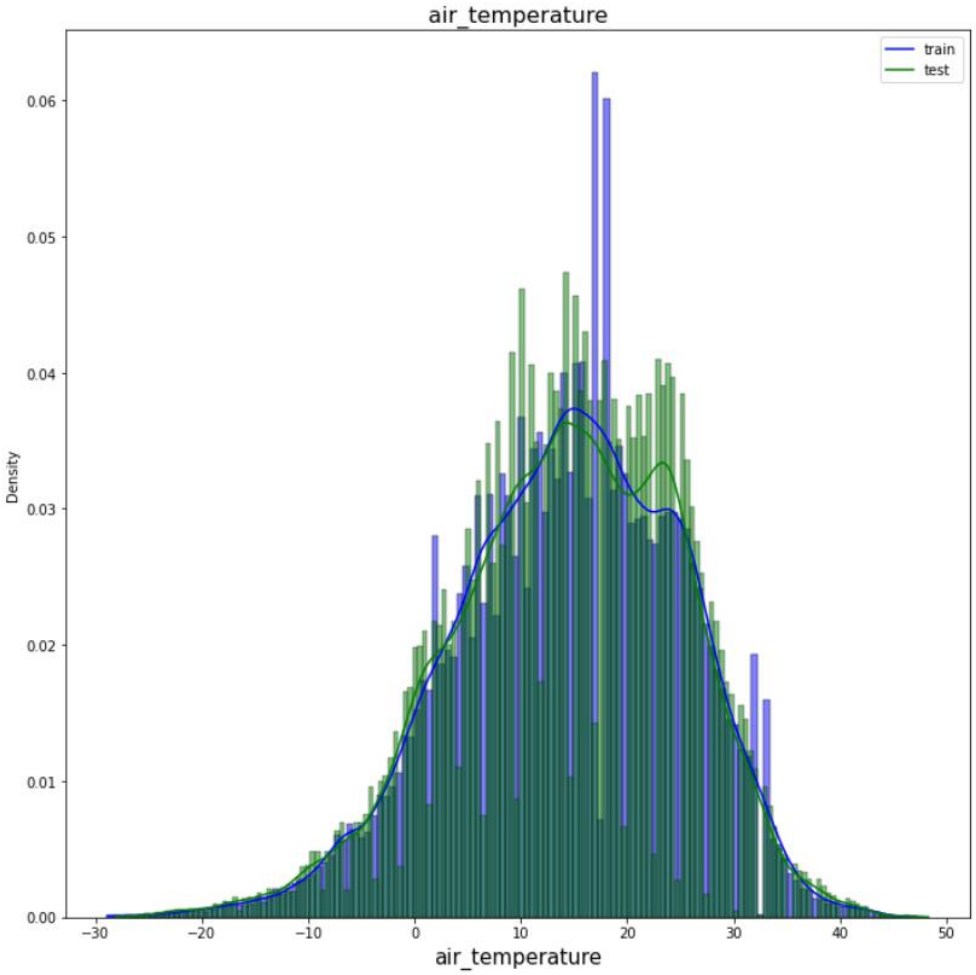
Analiza danych jest jednym z najważniejszych i najbardziej czasochłonnych elementów w projektach uczenia maszynowego. Często to, co widzimy gołym okiem nie daje nam pełnego spojrzenia na sytuację. Podczas pracy nad projektem uczenia maszynowego konieczne jest wizualizowanie i podsumowanie wykorzystywanych danych. Takie podejście pozwala nam uzyskać pewne spostrzeżenia, które umożliwią, jak najlepsze dobranie i dopracowanie zmiennych, które będą wykorzystywane podczas szkolenia modelu.

Na początku analizy stawiamy pewne hipotezy, które będziemy chcieli sprawdzić przed przystąpieniem do trenowania modeli. Podczas eksploracji staramy się potwierdzić lub obalić daną hipotezę. W projekcie inżynierskim analiza danych odegrała najważniejszą rolę w celu otrzymania jak najlepszy wyników przez modele. W poniższym podrozdziale zostały przedstawione wynik tej analizy i wnioski do jakich udało się dzięki niej dojść.

4.4.1 Analiza danych pogodowych

W pierwszej kolejności analizie zostały podane dane pogodowe. Głównym celem analizy był sprawdzenie jak prezentują się poszczególne rozkłady parametrów. W ten sposób można było sprawdzić czy któryś z parametrów nie będzie zaburzał działania modelu. Dane pogodowe składają się z 7 parametrów. Dla danych treningowych mamy do dyspozycji 139 773 odczyty, natomiast dla danych testowych jest ich 277 243. W dalszej części podrozdziału zostały przedstawione wykresy rozkładu gęstości dla wszystkich parametrów pogodowych.

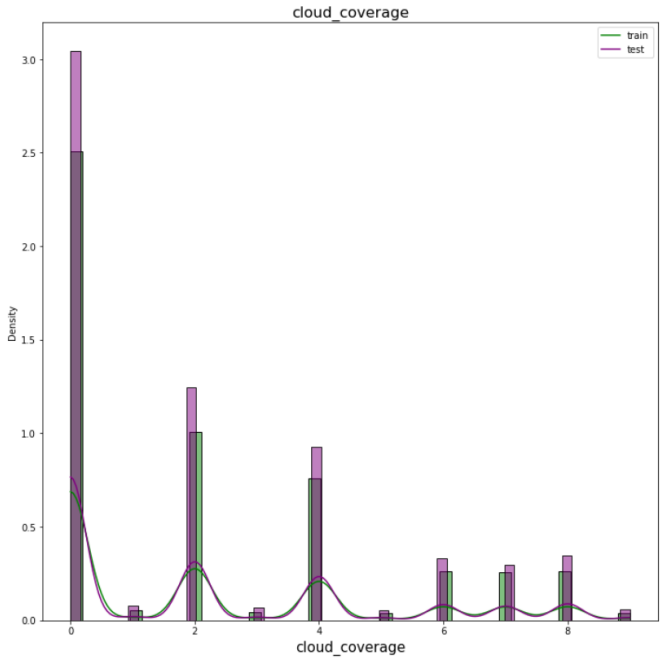
Rys 4.3 przedstawia wykres rozkładu dla temperatury powietrza. Jak widać dane temperaturowe mają rozkład normalny dla danych treningowych, jak i dla danych testowych. Dodatkowo można zaobserwować ze największa ilość odczytów zawiera się w przedziale -10 °C do 25 °C.



Rys. 4.3 Rozkład temperatury powietrza
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/thessis.ipynb

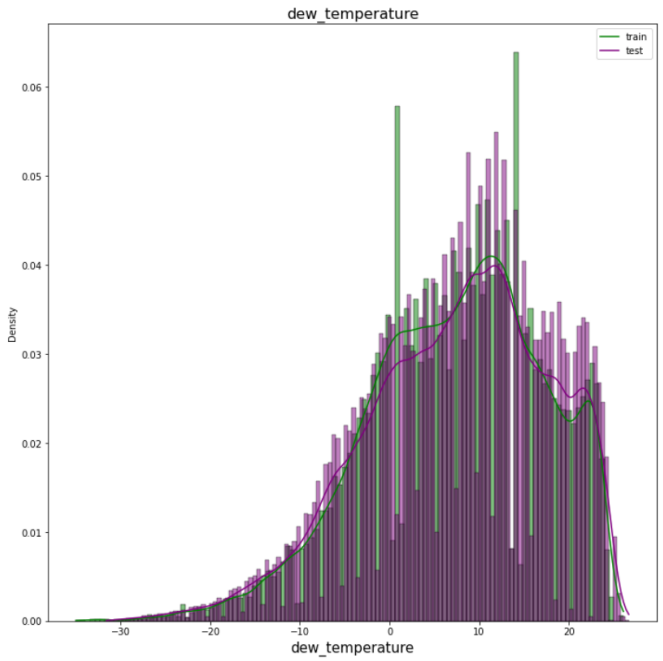
Następnym parametrem, który został przeanalizowany było zachmurzenie nieba. Pomiary ten był wyrażone w skali okta, w której to zero reprezentuje niebo całkowicie czyste, cztery niebo w połowie zachmurzone, osiem to niebo całkowicie zachmurzone, a dziewięć to niebo, którego nie dało się obserwować. Na podstawie Rys. 4.4 wiemy, że

najczęściej niebo było całkowicie czyste oraz rzadko występowała sytuacja, gdy ponad połowa nieba była przesłonięta.



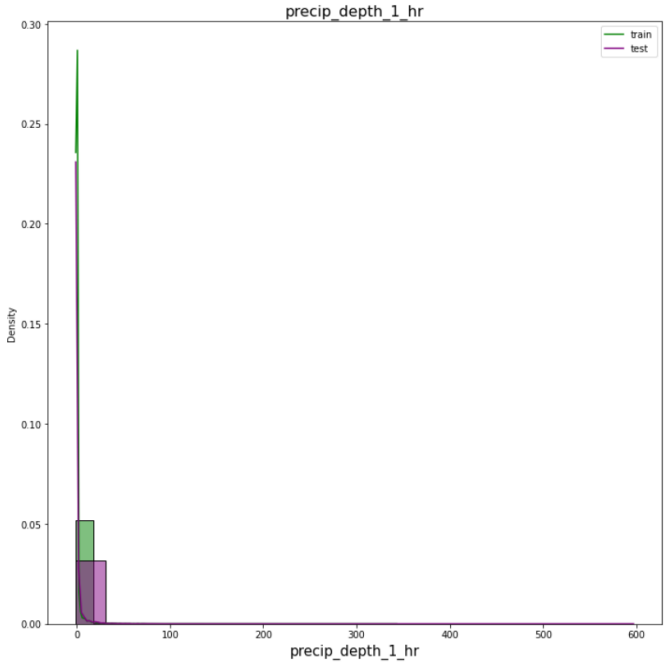
Rys. 4.4 Rozkład zachmurzenia
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/thessis.ipynb

Temperatura punktu rosy w danych pogodowych jest miarą wilgotności powietrza. Jest on temperaturą, do której powietrze musi zostać schłodzone, aby jego wilgotność względna osiągnęła 100 %. Rozkład temperatury rosy również ma rozkład normalny, a największa ilość pomiarów pochodzi z przedziału od -13 °C do 25 °C. Na Rys. 4.5 został przedstawiony wykres rozkładu.



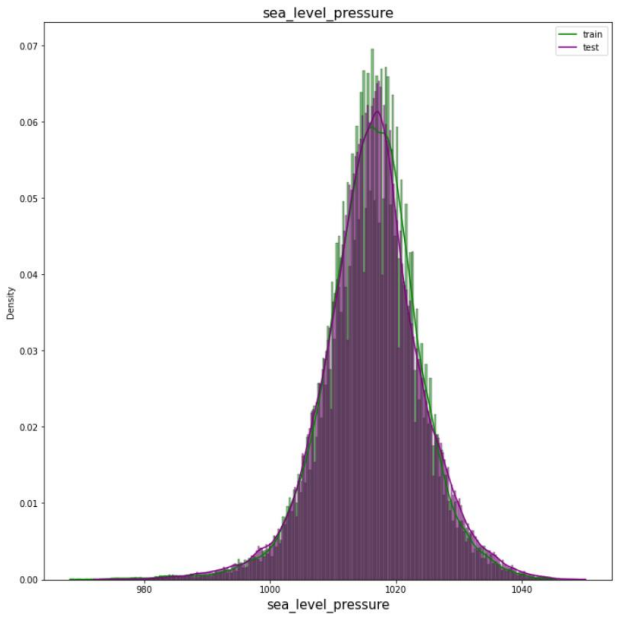
Rys. 4.5 Rozkład temperatury punktu rosy
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/thessis.ipynb

Głębokość opadów wskazuje, jak duże musiały by być opady, aby pokryć poziomą powierzchnie w okresie obserwacji, zakładając, że ciecz nie może spłynąć, odparować lub wsiąknąć. Parametr ten podawany jest w milimetrach (głębokość opadów równy 1 milimetrowi odpowiada 1 litrowi cieczy na 1 m^2 gruntu). Na Rys. 4.6 został przedstawiony rozkład tego parametru.



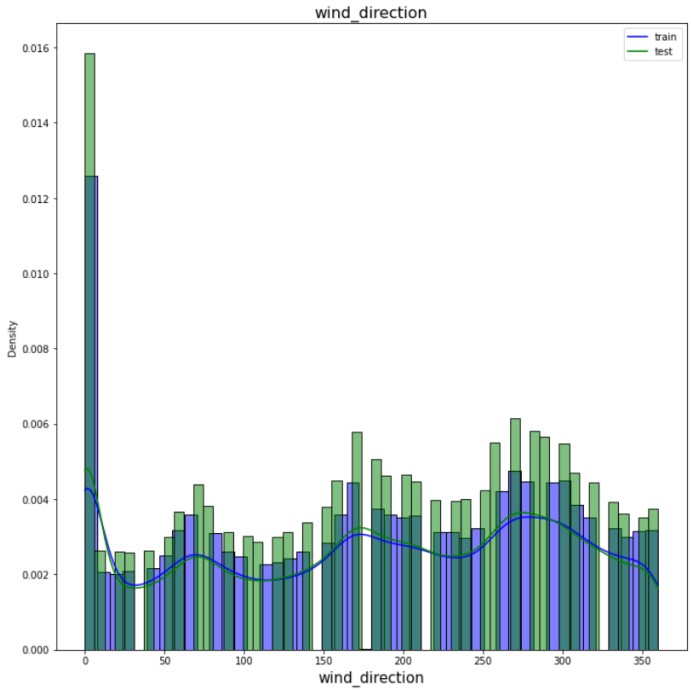
Rys. 4.6 Rozkład głębokości opadów
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/thessis.ipynb

Na Rys. 4.7 został przedstawiony rozkład ciśnienia atmosferycznego. Największa ilość odczytów zawiera się w przedziale 1010 do 1030 hPa, a rozkład jest rozkładem normalnym.

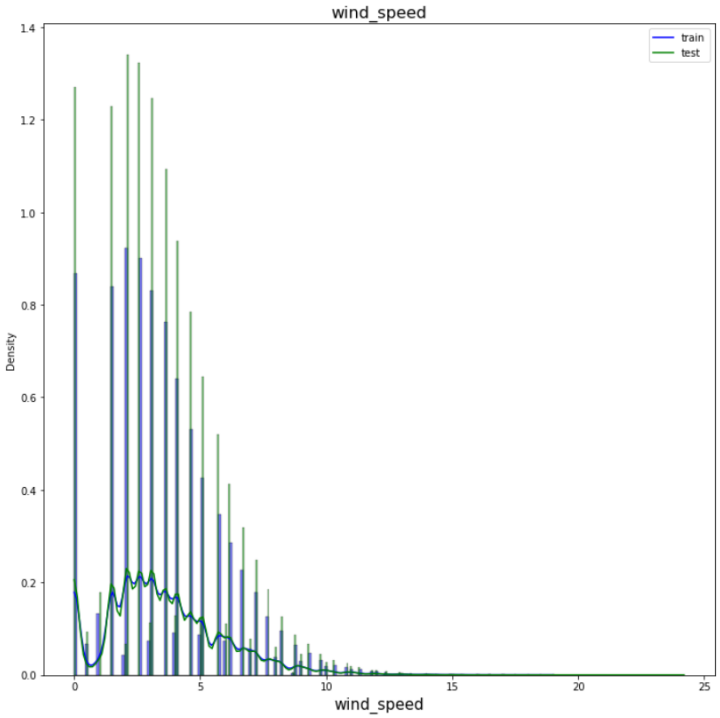


Rys. 4.7 Rozkład ciśnienia atmosferycznego
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/thessis.ipynb

Ostatnie dwa parametry odnoszą się do tego samego czynnika pogodowego to znaczy wiatru. Na Rys. 4.8 został przedstawiony rozkład kierunku wiatru, natomiast na Rys. 4.9 prędkość wiatru. Prędkość, która najczęściej była mierzona zawiera się w przedziale od $0\,\frac{m}{s}$ do $5\,\frac{m}{s}$, natomiast kierunek wiatru rozkłada się mniej więcej podobnie dla każdego kąta. Jedynie dla kąta równego $0\,^{\circ}$, możemy zaobserwować bardzo duże odchylenie.



Rys. 4.8 Rozkład kierunku wiatru
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/thessis.ipynb

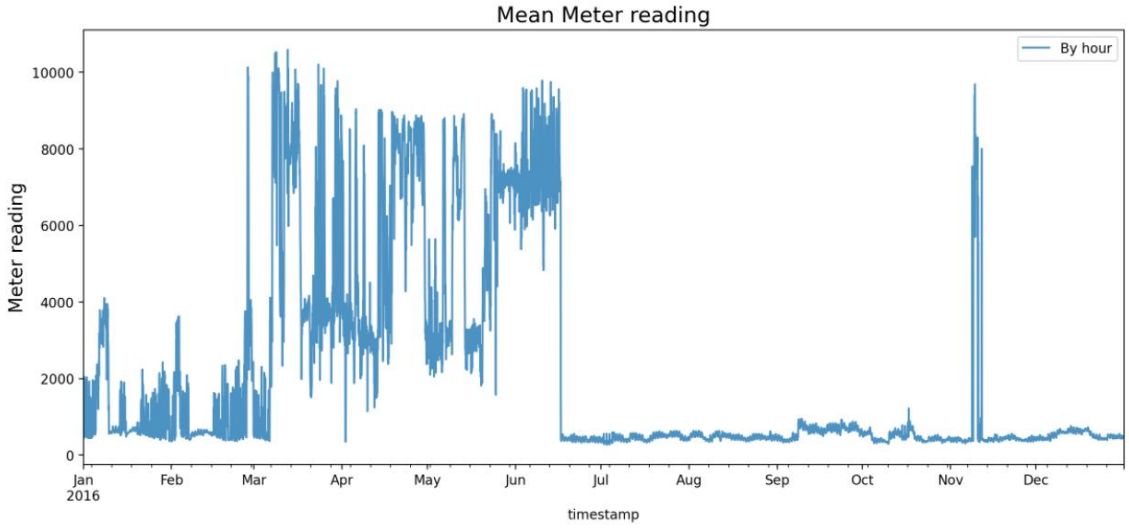


Rys. 4.9 Rozkład prędkości wiatru
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/thessis.ipynb

Podsumowując przeprowadzoną analizę, żaden z parametrów pogodowych nie wykazywał nietypowych rozkładów, nie zgodnych z oczekiwaniami. Skutkowało to brakiem konieczności pracy nad poprawieniem tych parametrów.

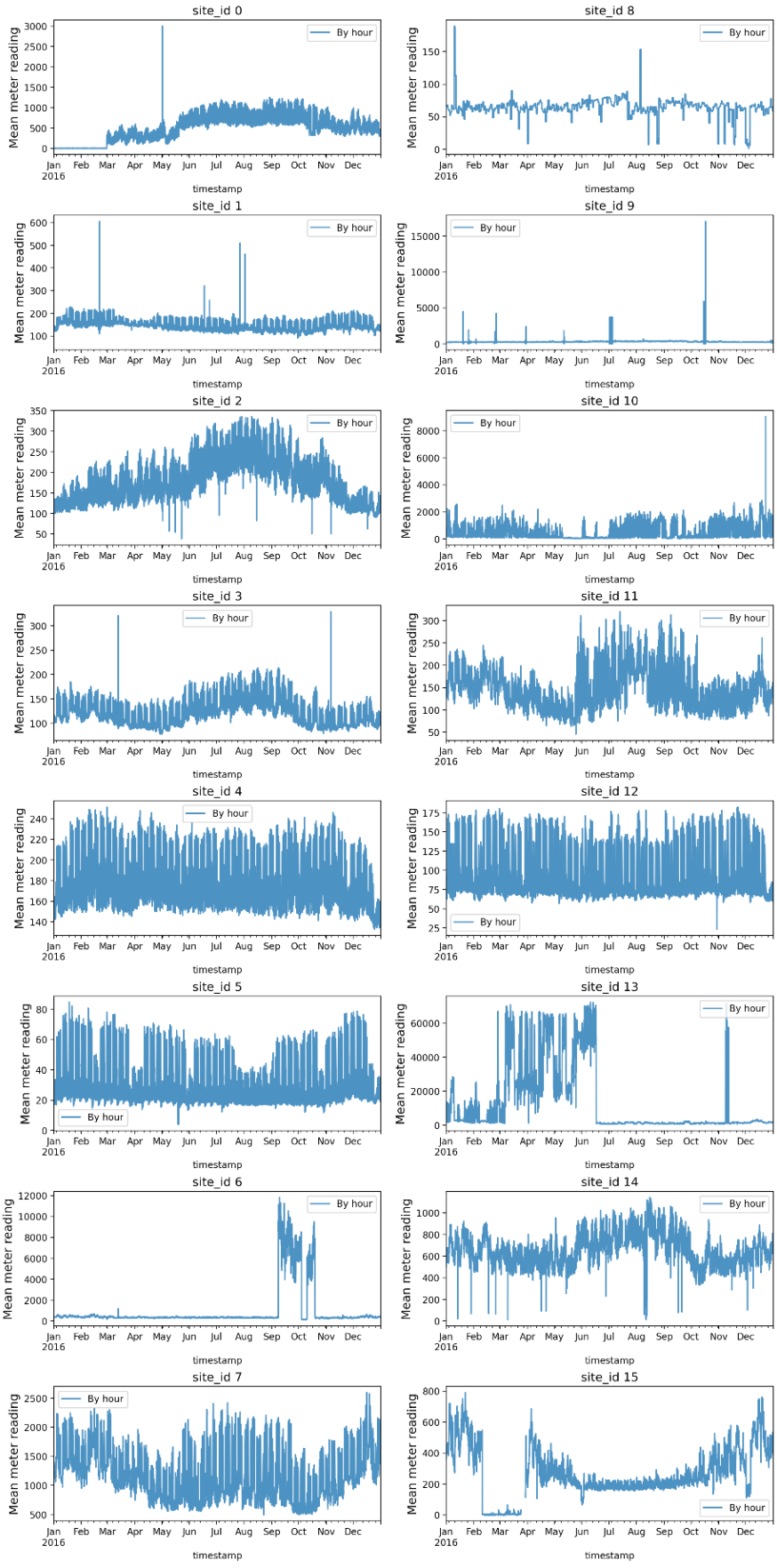
4.4.2 Analiza wyników pomiarów z mierników energii

Przeprowadzenie analizy zmiennej docelowej jest konieczne w celu sprawdzenie czy w zbiorze danych znajdują się punkty odstające, które w znacznym stopniu mogą zaburzać uczenie się modelu. Na Rys. 4.10 zostało przedstawione średnie zużycie energii dla danych treningowych. Na wykresie możemy zaobserwować dziwne anomalie w okresie od czerwca do listopada, w którym to następuje nagły pik danych, a potem powrót do wartości bliskiej zeru.



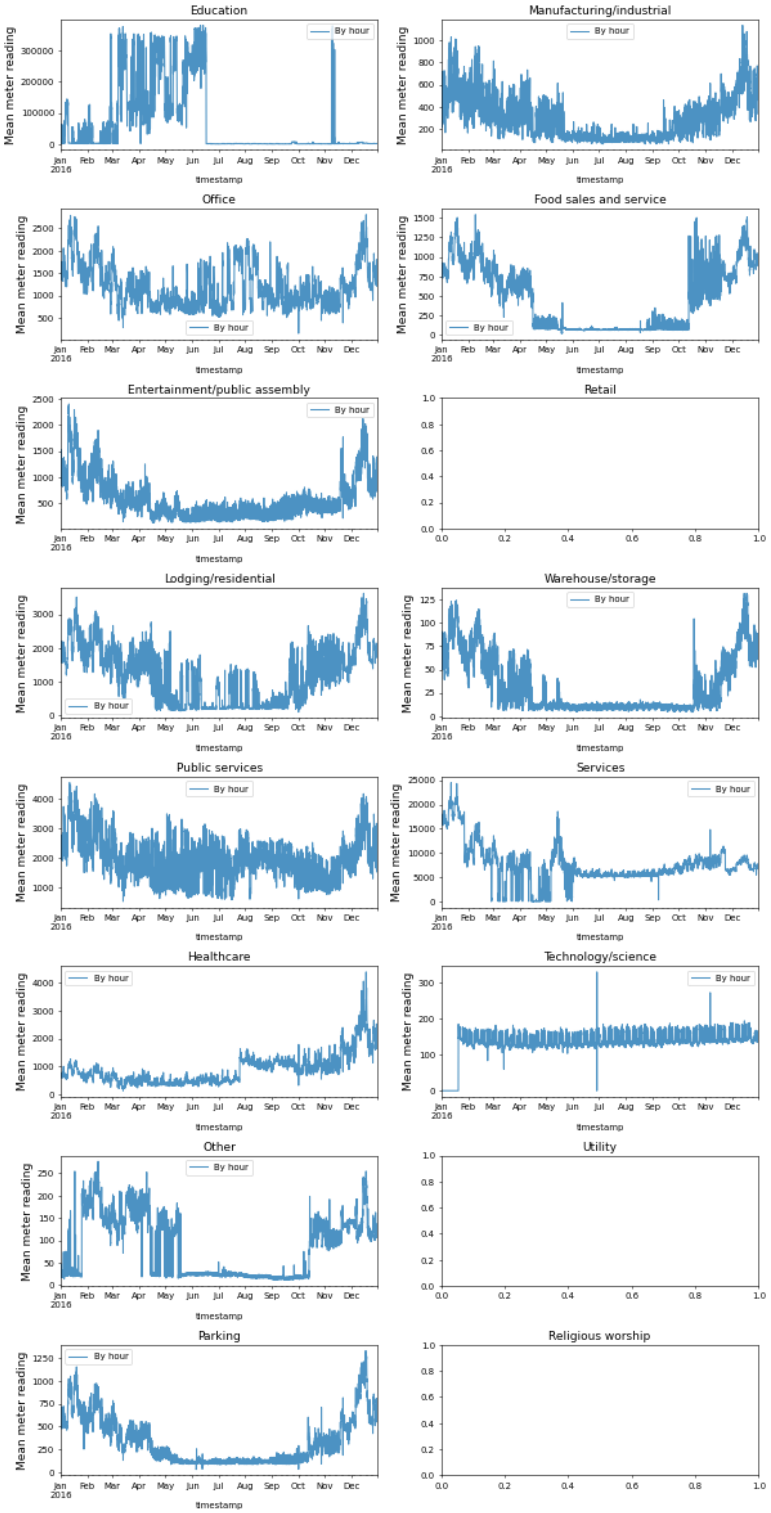
Rys. 4.10 Średnie zużycie energii przez budynki
Źródło:https://github.com/Senek18/engineer-thesis/blob/main/Data%20preprocesing%202.ipynb

Aby lepiej zrozumieć zaburzenia występujące w danych zostało sporządzone szesnaście wykresów średniego zużycia energii, w zależności od regionu w którym się znajdują. Rezultaty zostały przedstawione na Rys. 4.11. Na początek przyjrzymy się wykresowi z identyfikatorem 0. Jak można zaobserwować w okresie od stycznia do lutego pomiary dla wszystkich budynków o takim identyfikatorze wynoszą 0. Możliwe, że budynki w tym rejonie dopiero był budowane lub dopiero od marca budynki zostały opomiarowane. Po później przeprowadzonej dokładniejszej analizie zostało stwierdzone, że okres ten nie ma większego wpływu na wyniki. Kolejną dziwną anomalię możemy zaobserwować na wykresie o identyfikatorze 13. Przypomina on jeden do jednego wykres zużycia energii dla wszystkich budynków. Gdy spojrzymy na wykres wartości energii możemy zobaczyć, że budynki z tego regionu średnio zużywają znacznie większą ilość energii, przez co dość mocno wpływają na kształt krzywej na Rys 4.10.



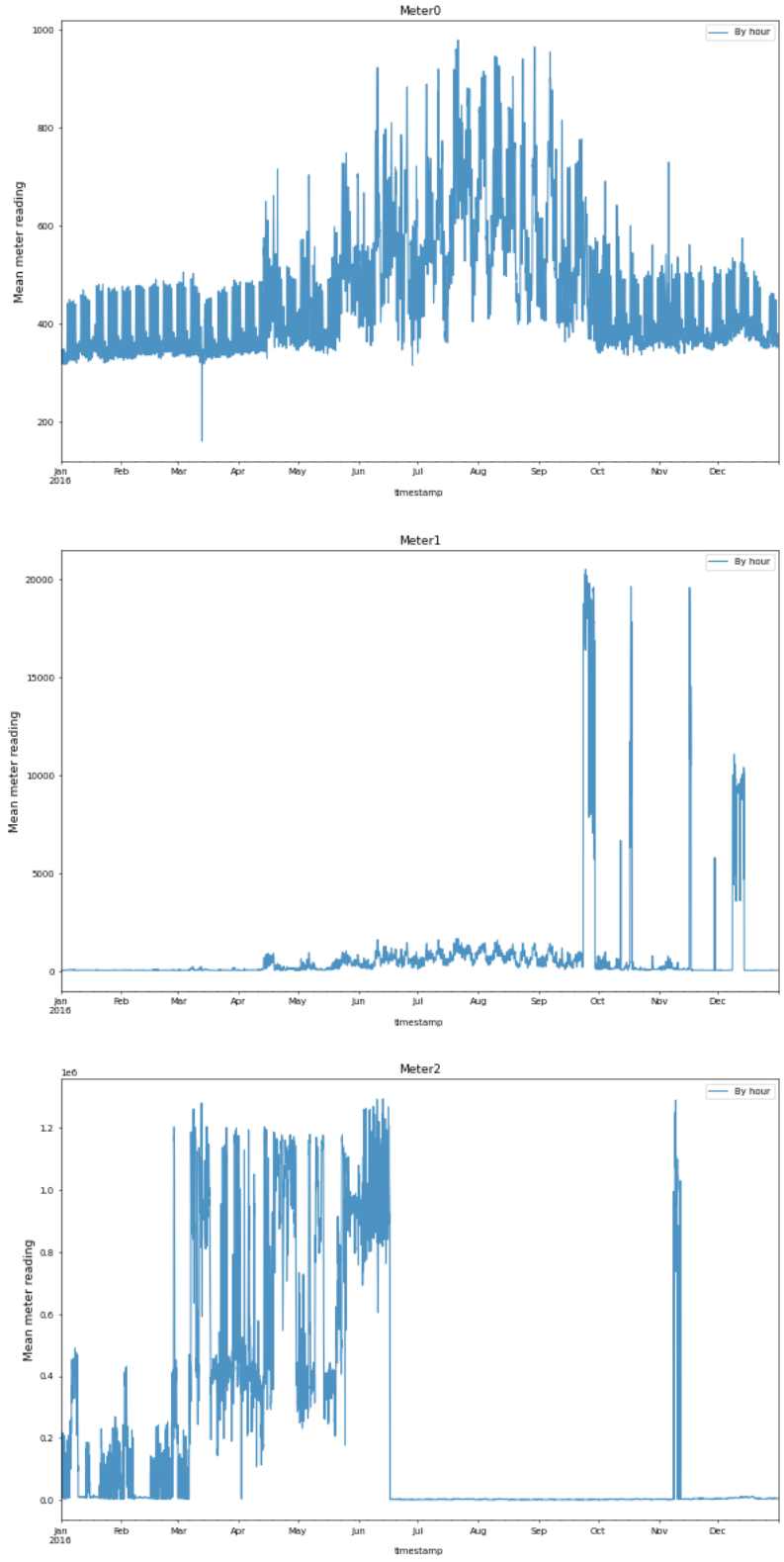
Rys. 4.11 Średnie zużycie w zależności od regionu w którym się znajdują
Źródło:https://github.com/Senek18/engineer-thesis/blob/main/Data%20preprocesing%202.ipynb

Następnym krokiem było spojrzenie na zużycie energii w tym regionie w zależności od użyteczności poszczególnych budynków, które zostały przedstawione na Rys 4.13. Puste wykresy oznaczają, że w podanym regionie nie znajdują się żadne budynki o takiej kategorii użyteczności. Kolejny raz powtarza się sytuacja, że jeden z wykresów wygląda identycznie jak wykres dla wszystkich budynków. Jak widzimy budynki o przeznaczeniu edukacyjnym mają znaczne zapotrzebowanie na energię.



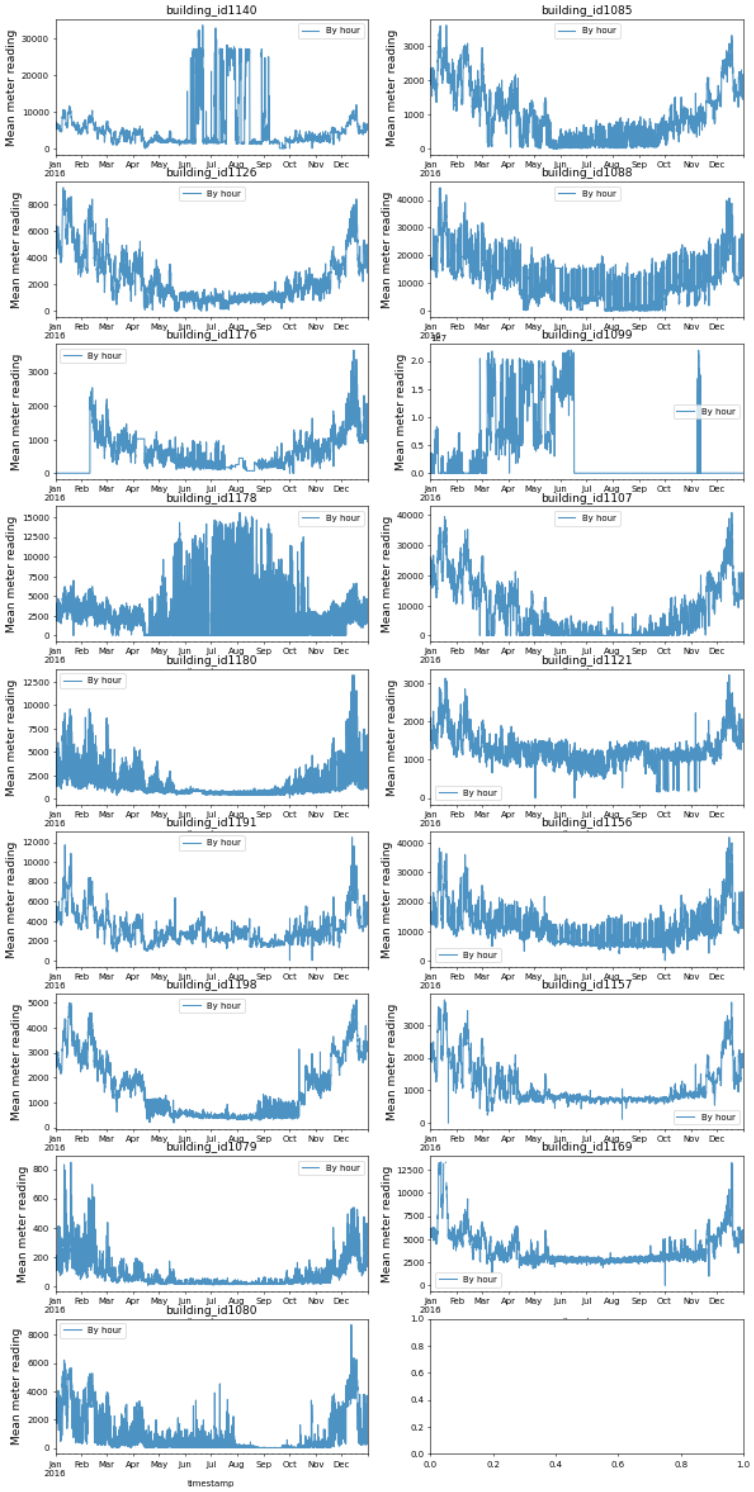
Rys. 4.12 Wykresy zużycie energii dla regionu 13 w zależności od przeznaczania
Źródło:https://github.com/Senek18/engineer-thesis/blob/main/Data%20preprocesing%202.ipynb

Na Rys. 4.14 przyjrzeliśmy się na jaki rodzaj energii jest tak duże zapotrzebowanie. Widzimy teraz, że największe zapotrzebowanie budynki mają na energię transportowaną za pomocą pary. Ostatnim krokiem będzie sprawdzenie czy każdy z budynków ma tak duże zapotrzebowanie.



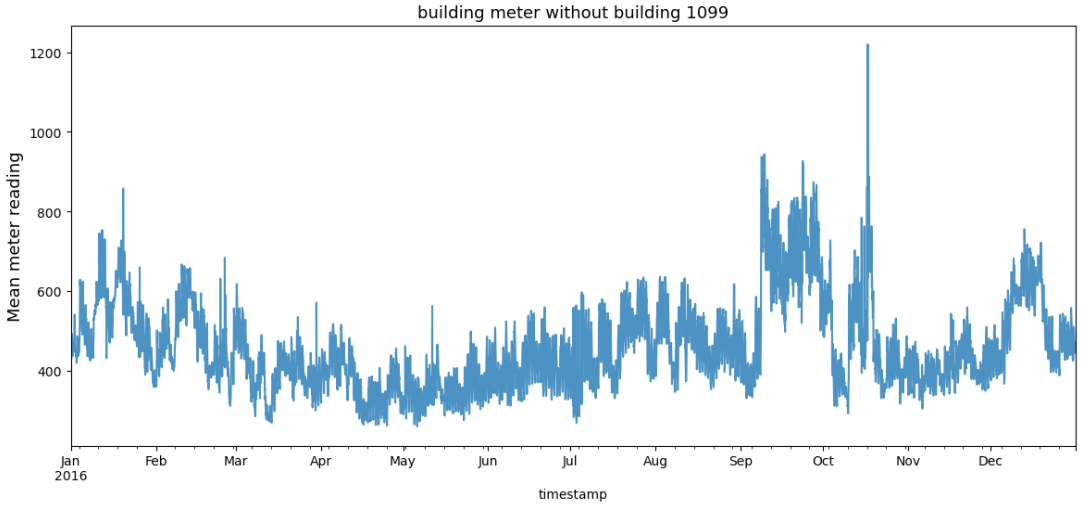
Rys. 4.13 Wykresy zużycia energii w zależności od jej typu dla regionu 13 i wykorzystywanych w celach edukacyjnych
Źródło:https://github.com/Senek18/engineer-thesis/blob/main/Data%20preprocesing%202.ipynb

W tym celu na Rys. 4.15 zostały zaprezentowane wykresy zużycia pary dla siedemnastu budynków edukacyjnych dla regionu trzynastego. Jak możemy zaobserwować budynek o numerze 1099 potrzebuje tak dużej ilości energii w postaci pary.



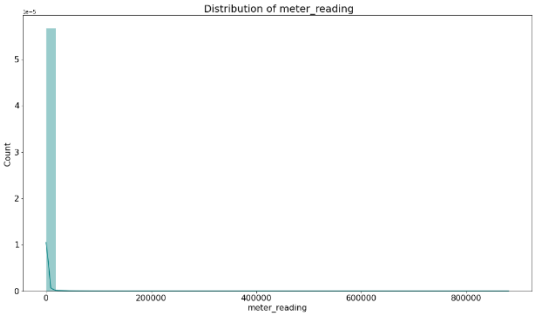
Rys. 4.14 Wykres zużycia pary dla budynków edukacyjnych w regionie 13.
Źródło:https://github.com/Senek18/engineer-thesis/blob/main/Data%20preprocesing%202.ipynb

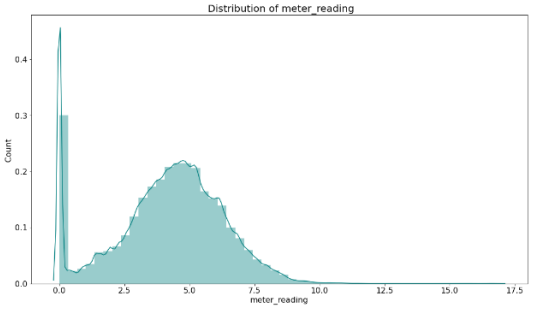
Na Rys. 4.12 został przedstawiony nowy wykres średniego zużycia energii dla wszystkich budynków. Jak możemy zauważyć zbiór danych po wykluczeniu budynku 1099 zachowuje się regularnie bez większych długotrwałych zaburzeń.



Rys. 4.15 Średnie zużycie energii bez uwzględnienia budynku o numerze 1099
Źródło:https://github.com/Senek18/engineer-thesis/blob/main/Data%20preprocesing%202.ipynb

Na koniec analizy, został sporządzony wykres rozkładu pomiarów. Został on przedstawiony na Rys. 4.16. W celu znormalizowania rozkładu wartości pomiarów zostały zlogarytmizowane i przedstawione na Rys. 4.17.





Rys. 4.16 Oryginalny rozkład

Rys. 4.17 Rozkład po zlogarytmowaniu

Źródło:https://github.com/Senek18/engineer-thesis/blob/main/Data%20preprocesing%202.ipynb

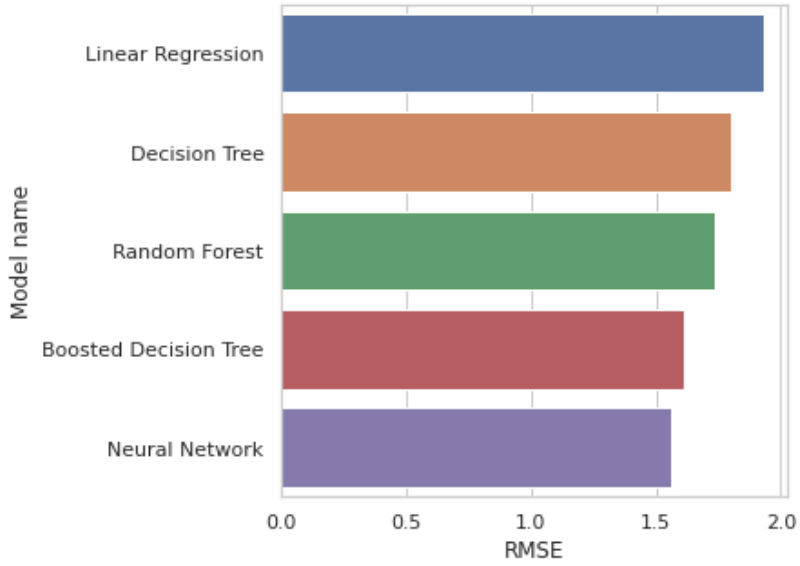
Dzięki takiemu zabiegowi algorytmy będą znacznie lepiej przewidywały wartości. Na początku zamiana ta nie została przeprowadzona, co za skutkowało złymi modelami, którym zdarzało się nawet przewidywać wartości ujemne. Zabieg ten znacząco poprawił skuteczności wszystkich modeli.

4.5 Trenowanie modeli

Algorytmy uczenia maszynowego został wytrenowane na nowym zaktualizowanym zestawie danych. Do tworzenia standardowych modeli uczenia maszynowego zostały użyte gotowe moduły zawarte w bibliotece MLlib pochodzącej z narzędzia Apache Spark oraz w celu stworzenia modelu sieci neuronowych wykorzystano bibliotekę TensorFlow. W projekcie utworzono modele regresyjne za pomocą regresji liniowej, drzewa

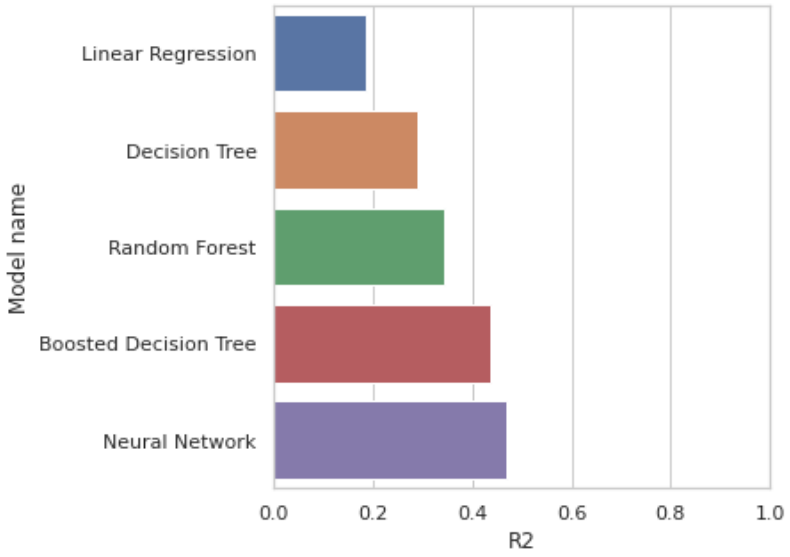
decyzyjnego, lasów losowych, wzmocnionego drzewa decyzyjnego oraz sieci neuronowych.

W celu porównania skuteczności modeli wykorzystano pierwiastek błędu średnio kwadratowego oraz współczynnik R-kwadrat obliczonych na danych, których modele wcześniej nie widziały. Na Rys 4.18 został przedstawiony wykres podsumowujący pierwiastek błędu średniokwadratowego modeli. Najniższą wartość wykazał model oparty o sieci neuronowe, ale wartość błędu jest tylko nie znacznie niższa od błędu modelu opartego na wzmocnionych drzewach decyzyjnych.



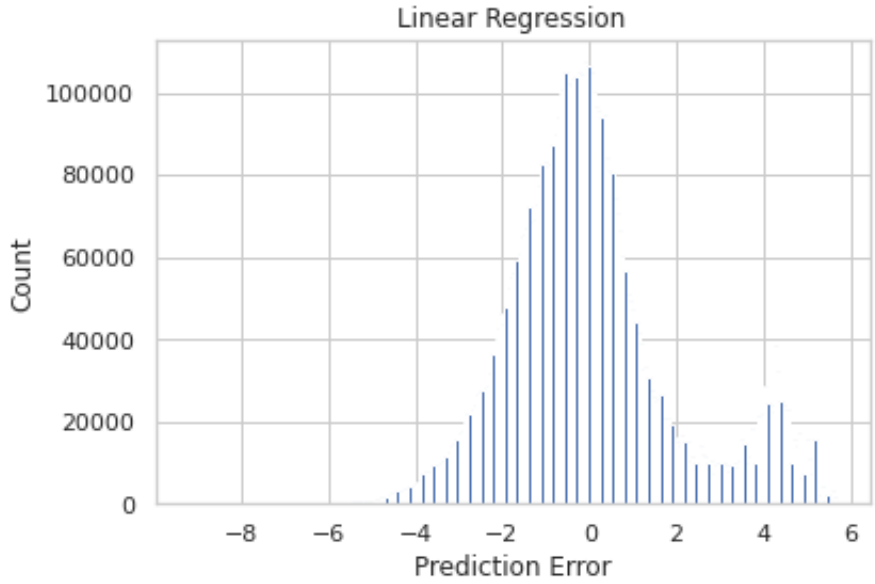
Rys 4.18 Wykres porównawczy pierwiastka błędu średniokwadratowego
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/ML%20models.ipynb

Na Rys 4.19 przedstawiono wykres współczynnika R-kwadrat dla wszystkich modeli. W przypadku tego parametru wykorzystywanego do oceny modeli, dążymy do tego, aby wartość był jak najbliższa jedynce.



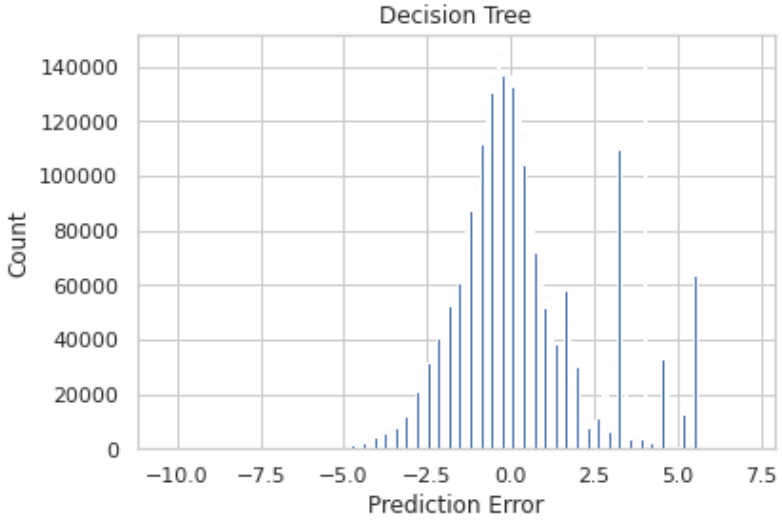
Rys 4.19 Wykres porównawczy współczynnika R-kwadrat
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/ML%20models.ipynb

Zostały również przygotowane wykresy rozkładu błędu obliczanego jako różnica wartości przewidywanej przez model oraz wartości rzeczywistej. W ten sposób możemy sprawdzić jak często występują odchylenia od rzeczywistych danych. Na Rys. 4.20 przedstawiono rozkład błędu dla regresji liniowej. Jak możemy zaobserwować wykres ma tendencje rozkładu normalnego ze szczytem zlokalizowanym blisko wartości zerowej.



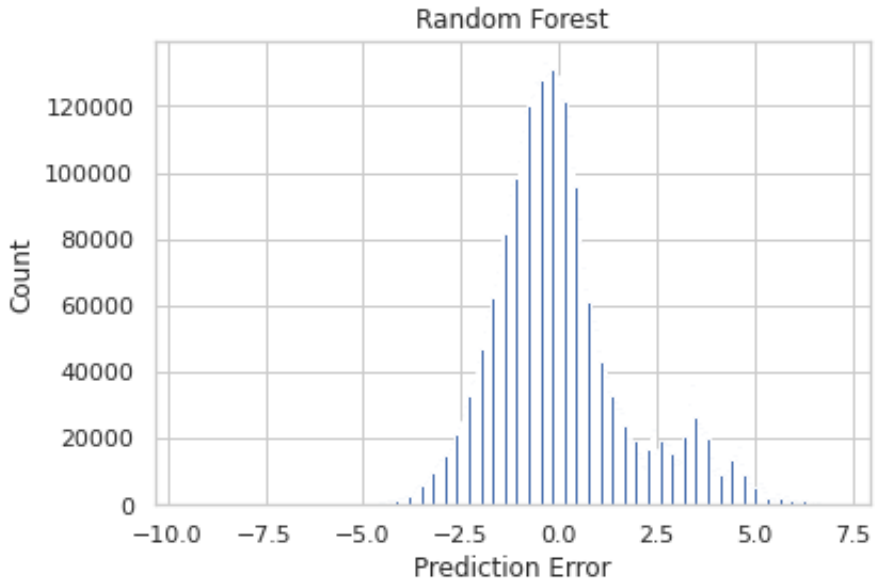
Rys. 4.20 Wykres błędu dla regresji liniowej
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/ML%20models.ipynb

Na kolejnym wykresie przedstawionym na Rys 4.21 możemy zaobserwować również wykres zbliżony wyglądem do rozkładu normalnego. Jednak w dodatniej części wykresu możemy zaobserwować kilka zaburzeń. Sugeruje to nam, że algorytm może mieć tendencję do zawyżania wartości przewidywanych. Szczyt rozkładu również zlokalizowany jest w okolicach zera.



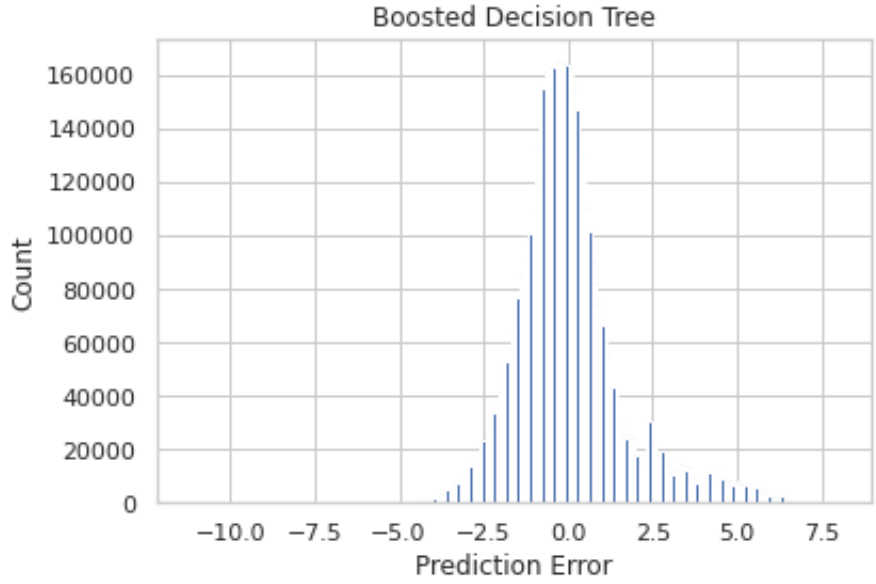
Rys. 4.21 Wykres błędu dla drzewa decyzyjnego
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/ML%20models.ipynb

Na Rys. 4.22 przedstawiono rozkład błędu dla algorytmu lasów losowych. Rozkład również wykazuje tendencje rozkładu Gaussa. Możemy również zaobserwować większą ilość przypadków z zawyżona wartością przewidywania.



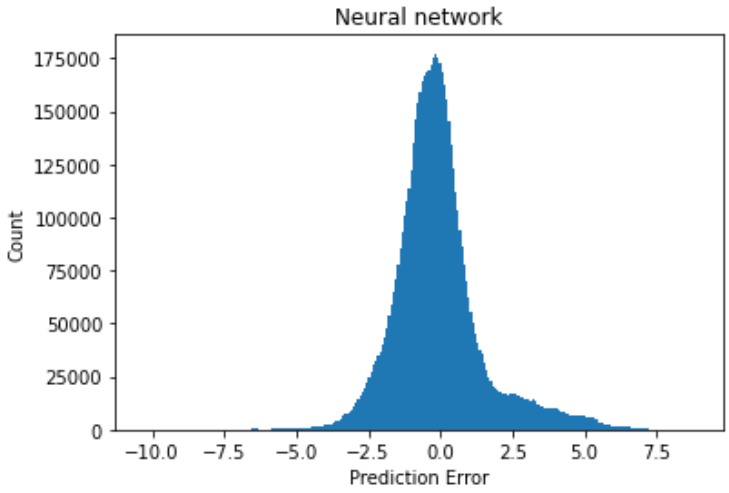
Rys. 4.22 Wykres błędu dla lasów losowych
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/ML%20models.ipynb

Rozkład wzmocnionego drzewa decyzyjnego przedstawiony na Rys. 4.23 wraz z rozkładem dla sieci neuronowych przedstawionym na Rys. 4.24 w największym stopniu przypominają rozkład normalny. Wykresy są praktycznie symetryczne z szczytami zlokalizowanymi w okolicach zera.



Rys. 4.23 Wykres błędu dla wzmocnionego drzewa decyzyjnego
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/ML%20models.ipynb

Algorytm sieci neuronowych posiada nie znacznie większą ilość wartości przewidzianych prawidłowo lub z bardzo małym błędem.



Rys. 4.24 Wykres błędu dla sieci neuronowych
Źródło: https://github.com/Senek18/engineer-thesis/blob/main/DLmodel.ipynb

W Tab. 4.1 przedstawiono tabele podsumowująca parametry, wykorzystane do wyboru optymalnego algorytmu. W tabeli oprócz pierwiastka błędu średniokwadratowego i współczynnika R-kwadrat znalazł się parametr czas. Wyraża on długość trenowania modelu w minutach.

Tab. 4.1. Porównanie modeli

Model	RMSE	R^2	czas [min]
Regresja liniowa	1,931	0,186	2,93
Drzewo decyzyjne	1,803	0,291	6,85
Lasy losowe	1,734	0,344	8,79
Wzmacniające drzewo decyzyjne	1,609	0,435	21,45
Sieci neuronowe	1,561	0,468	32,85

5. Podsumowanie

Coraz większa ilość gromadzonych danych otwiera furtkę dla wykorzystania uczenia maszynowego w sektorze energetycznym. Jego zastosowanie jest wstanie usprawnić, zabezpieczyć i obniżyć koszty działania tego sektora.

Celem projektu inżynierskiego było porównanie pięciu algorytmów uczenia maszynowego wykorzystanych do prognozowania zapotrzebowania na energię. Podczas pracy zostały przeanalizowane dane wykorzystywane do szkolenia modeli. W trakcie analiz zauważono, że jeden z budynków w znacznym stopniu odstaje od pozostałych budowli. W celu uniknięcia złego wyszkolenia modelu dane zostały usunięte ze zbioru treningowego. Ważnym zagadnieniem, który również udało się zaobserwować jest normalizacja danych wejściowych. Podczas pierwszej próby modele wykazywały słabą skuteczność, a niektóre wartości przewidywane były jako liczby ujemne, co było niedopuszczalne. Na skutek normalizacji uzyskaliśmy przyzwoite wyniki. Najwyższą skutecznością wykazał się model wykorzystujący algorytm sieci neuronowych. Błąd średniokwadratowy dla modelu wyniósł 1,561 potrzebując około 33 minut do ukończenia treningu. Natomiast najszybszym modelem była regresja liniowa, która potrzebowała nie całych 3 minut treningu, osiągając skuteczność na poziomie 1,931, która natomiast była najgorszą ze wszystkich pięciu modeli. Jednak optymalnym algorytmem, który można z powodzeniem wykorzystywać jest wzmocnione drzewo decyzyjne. Jego pierwiastek błędu średniokwadratowego różni się o 0,048 od błędu sieci neuronowych, ale czas potrzebny do trenowania modelu jest niższy o prawie 12 minut.

Przeprowadzone badania mają szereg możliwości usprawnienia, które można zaimplementować w przyszłości. Na przykład można przygotować kilka modeli opartych na tym samym algorytmie uczącym w celu znalezienia optymalnych parametrów czy jeszcze dokładniejsze przeanalizowanie parametrów wejściowych w celu odnalezienia innych odczytów mogących zaburzać szkolenie.

W przyszłości prognozowanie energii oparte o algorytmy wymienione w pracy inżynierskiej mogą być wykorzystywane przez firmy dystrybucyjne w celu jak najlepszego sterowania system elektroenergetycznym i ciepłowniczym. Modele mogą również wspierać inwestorów w ocenie skuteczności modernizacji energetycznej budynków.

Literatura

[1] „John McCarthy”, *Wikipedia, wolna encyklopedia*. 20 październik 2021. Dostęp: 16 grudzień 2021. [Online]. Dostępne na: https://pl.wikipedia.org/w/index.php?title=John_McCarthy&oldid=64851395

[2] X. Wang, „MACHINE LEARNING APPLICATIONS IN POWER SYSTEMS”, *Electr. Eng. Theses Diss.*, lip. 2020, [Online]. Dostępne na: https://scholar.smu.edu/engineering_electrical_etds/39

[3] „Raporty za rok 2020 - PSE”. https://www.pse.pl/dane-systemowe/funkcjonowanie-kse/raporty-roczne-z-funkcjonowania-kse-za-rok/raporty-za-rok-2020#r6_1 (dostęp 24 listopad 2021).

[4] M. H. Brown i R. P. Sedano, *Electricity transmission: a primer*. Denver, Colo.: National Council on Electric[ity] Policy, 2004.

[5] W. Mielczarski, *Rynki energii elektrycznej: wybrane aspekty techniczne i ekonomiczne*. Agencja Rynku Energii, 2000.

[6] V. Masson-Delmotte *i in.*, Red., *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2021.

[7] D. C. P. Barbosa *i in.*, „Machine Learning Approach to Detect Faults in Anchor Rods of Power Transmission Lines”, *IEEE Antennas Wirel. Propag. Lett.*, t. 18, nr 11, s. 2335–2339, lis. 2019, doi: 10.1109/LAWP.2019.2932052.

[8] Van Nhan Nguyen, Robert Jenssen, i Davide Roverso, „Intelligent Monitoring and Inspection of Power Line Components Powered by UAVs and Deep Learning”, *IEEE Power and Energy Technology Systems Journal*, t. 6, nr 1, s. 11–21, marzec 2019.

[9] B. Doucoure, K. Agbossou, i A. Cardenas, „Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data”, *Renew. Energy*, t. 92, s. 202–211, lip. 2016, doi: 10.1016/j.renene.2016.02.003.

[10] M. Khodayar, J. Wang, i M. Manthouri, „Interval Deep Generative Neural Network for Wind Speed Forecasting”, *IEEE Trans. Smart Grid*, t. 10, nr 4, s. 3974–3989, lip. 2019, doi: 10.1109/TSG.2018.2847223.

[11] M. Cellura, F. Guarino, S. Longo, i G. Tumminia, „Climate change and the building sector: Modelling and energy implications to an office building in southern Europe”, *Energy Sustain. Dev.*, t. 45, s. 46–65, sie. 2018, doi: 10.1016/j.esd.2018.05.001.

[12] „Inteligentny budynek”, *Wikipedia, wolna encyklopedia*. 17 listopad 2021. Dostęp: 4 styczeń 2022. [Online]. Dostępne na: https://pl.wikipedia.org/w/index.php?title=Inteligentny_budynek&oldid=65360871

[13] N.-S. Truong, N.-T. Ngo, i A.-D. Pham, „Forecasting Time-Series Energy Data in Buildings Using an Additive Artificial Intelligence Model for Improving Energy Efficiency”, *Comput. Intell. Neurosci.*, t. 2021, s. e6028573, lip. 2021, doi: 10.1155/2021/6028573.

[14] A. Ahmad, S. Ganguly, i F. Wang, „Optimised building energy and indoor microclimatic predictions using knowledge-based system identification in a historical art gallery”, *Neural Comput. Appl.*, t. 32, kwi. 2020, doi: 10.1007/s00521-019-04224-7.

[15] S. Seyedzadeh, F. Pour Rahimian, P. Rastogi, i I. Glesk, „Tuning Machine Learning Models for Prediction of Building Energy Loads”, *Sustain. Cities Soc.*, t. 47, mar. 2019, doi: 10.1016/j.scs.2019.101484.

[16] S. Kalogirou, C. Neocleousp, i C. Schizas, „Building heating load estimation using artificial neural networks”, *Proc. Int. Conf. CLIMA 2000*, sty. 1997.

[17] S. A. Kalogirou, G. A. Florides, C. Neocleous, i C. N. Schizas, „Estimation of the Daily Heating and Cooling Loads Using Artificial Neural Networks”, wrz. 2001, Dostęp: 4 styczeń 2022. [Online]. Dostępne na: https://ktisis.cut.ac.cy/handle/10488/883
[18] B. Dong, C. Cao, i S. E. Lee, „Applying support vector machines to predict building energy consumption in tropical region”, 2005, doi: 10.1016/J.ENBUILD.2004.09.009.
[19] A. Tsanas i A. Xifara, „Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools”, *Energy Build.*, t. 49, s. 560–567, cze. 2012, doi: 10.1016/j.enbuild.2012.03.003.
[20] Sara Brown, „Machine learning, explained”, *MIT Sloan*. https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained (dostęp 2 grudzień 2021).
[21] S. J. (Stuart J. Russell, *Artificial intelligence : a modern approach*. Third edition. Upper Saddle River, N.J. : Prentice Hall, [2010] ©2010, 2010. [Online]. Dostępne na: https://search.library.wisc.edu/catalog/9910082172502121
[22] T. Hastie, R. Tibshirani, i J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, 2nd edition. New York, NY: Springer, 2016.
[23] Y. Luo, H. Xu, Y. Li, Y. Tian, T. Darrell, i T. Ma, „Algorithmic Framework for Model-based Deep Reinforcement Learning with Theoretical Guarantees”, *ArXiv180703858 Cs Stat*, luty 2021, Dostęp: 12 grudzień 2021. [Online]. Dostępne na: http://arxiv.org/abs/1807.03858
[24] G. Bonaccorso, *Mastering Machine Learning Algorithms - Second Edition: Expert techniques for implementing popular machine learning algorithms, fine-tuning your models, and understanding how they work, 2nd Edition*. Birmingham Mumbai, 2020.
[25] stanfordonline, *Lecture 2 - Linear Regression and Gradient Descent | Stanford CS229: Machine Learning (Autumn 2018)*, (2020). Dostęp: 13 grudzień 2021. [Online]. Dostępne na: https://www.youtube.com/watch?v=4b4MUYve_U8
[26] stanfordonline, *Lecture 10 - Decision Trees and Ensemble Methods | Stanford CS229: Machine Learning (Autumn 2018)*, (2020). Dostęp: 28 grudzień 2021. [Online]. Dostępne na: https://www.youtube.com/watch?v=wr9gUr-eWdA
[27] S. Shalev-Shwartz, *Understanding Machine Learning: From Theory to Algorithms*, 1st edition. New York, NY, USA: Cambridge University Press, 2014.
[28] stanfordonline, *Lecture 12 - Backprop & Improving Neural Networks | Stanford CS229: Machine Learning (Autumn 2018)*, (2020). Dostęp: 2 styczeń 2022. [Online]. Dostępne na: https://www.youtube.com/watch?v=zUazLXZZA2U
[29] stanfordonline, *Lecture 11 - Introduction to Neural Networks | Stanford CS229: Machine Learning (Autumn 2018)*, (2020). Dostęp: 2 styczeń 2022. [Online]. Dostępne na: https://www.youtube.com/watch?v=MfIjxPh6Pys
[30] „Preparing data for a machine learning model.”, *Jeremy Jordan*, 30 maj 2017. https://www.jeremyjordan.me/preparing-data-for-a-machine-learning-model/ (dostęp 27 grudzień 2021).
[31] J. Brownlee, „Regression Metrics for Machine Learning”, *Machine Learning Mastery*, 19 styczeń 2021. https://machinelearningmastery.com/regression-metrics-for-machine-learning/ (dostęp 3 styczeń 2022).
[32] „pandas (software)”, *Wikipedia*. 11 grudzień 2021. Dostęp: 5 styczeń 2022. [Online]. Dostępne na: https://en.wikipedia.org/w/index.php?title=Pandas_(software)&oldid=1059759509
[33] „What is NumPy? — NumPy v1.22 Manual”. https://numpy.org/doc/stable/user/whatisnumpy.html (dostęp 5 styczeń 2022).

Spis rysunków

Spis tabel