Raport z projektu - eksploracyjna analiza danych

Krzysztof Bugajski

10.05.2024

1 Portal dane.gov.pl

Rodzaje dostępnych danych

Portal dane.gov.pl udostępnia z wielu dziedzin podzielone są one na następujące kategorie: Edukacja, kultura i sport; Energia; Gospodarka i finanse; Kwestie międzynarodowe; Ludność i społeczeństwo; Nauka i technologia; Regiony i miasta; Rolnictwo, rybołówstwo, leśnictwo i żywność; Rząd i sektor publiczny; Sprawiedliwość, ustrój sądów i bezpieczeństwo publiczne; Środowisko; Transport; Ukraina; Zdrowie
Dane dostępne są do pobrania w formatach: CSV, JSON, XML lub za pomocą interfejsu API

Dostęp do danych

Portal dane.gov.pl umożliwia bezpłatne korzystanie z danych publicznych, dane są udostępniane bezpłatne do ponownego wykorzystania bez żadnych ograniczeń, dane są dostępne bez rejestracji w serwisie.
Ponadto portal promuje wykorzystanie zebranych danych do tworzenia różnych aplikacji i serwisów internetowych i zamieszcza takie projekty na swojej stronie.

Rodzaje API

Portal udostępnia swoje oficjalne API do pobierania wybranych danych, niektóre dane udostępniane są również poprzez API partnerów. Na portalu widnieje informacja, że API powinno obsługiwać sortowanie, filtrowanie i przeszukiwanie danych za pomocą prostego w obsłudze interfejsu webowego. Wszystkie odpowiedzi z API są zwracane w formacie JSON.

Jakość danych

Na portalu można określić jakoś danych za pomocą 5 stopniowej skali otwartości danych, która określa jak bardzo dane są przygotowane do dalszego przetwarzania. Istnieje również możliwość szybkiego podglądu niektórych danych w formacie tabeli. Istnieją również różne kategorie danych które pozwalają wybrać np. Dane o wysokiej wartości, Dane dynamiczne, Dane badawcze itp.

2 Wybór danych do projektu

Do realizacji projektu wybrano dane z testów modernizowanego układu zasilania elektrycznego w dźwignicy żurawia zastosowanego jako alternatywy dla silnika Diesla w kontekście parametrów dźwignicy, kosztów zużycia energii elektrycznej i paliw, podczas transportu pionowego ładunków w różnych warunkach pogodowych.
Dane zawierają następujące kolumny:

- Nr. pomiaru
- Data
- Ciężar ładunku [T]
- Długość wysięgnika [m]
- Odległość od osi [m]
- Wysokość podnoszenia [m]
- Maksymalne, chwilowe zużycie ON [l/h]
- Dzienne zużycie ON [l/8h]
- Cena hurtowa ON 1000l [PLN]
- Maksymalne, chwilowe zużycie energii elektrycznej [kW]
- Dzienne zużycie energii elektrycznej [kW/8h]
- Cena energii elektrycznej [kWh]
- Koszt dzienny [PLN]
- Prędkość wiatru [km/h]
- Prędkość wiatru [m/s]
- Temperatura [C]
- Ciśnienie [hPa]

3 Wykonanie pierwszego etapu pipeline'u ML

3.1 Proponowane użycie danych w ujęciu uczenia maszynowego

Dane te mogą być wykorzystane do opracowania modeli ML, które pomagają w przewidywaniu oraz optymalizacji zużycia energii elektrycznej w zależności od warunków pracy żurawia.
Można zastosować techniki uczenia nienadzorowanego, takie jak klastrowanie, aby zidentyfikować wzorce w danych pogodowych dotyczących zużycia energii elektrycznej.
Dodatkowo można wykorzystać uczenie nadzorowane do budowy modeli, które przewidują koszty eksploatacji żurawia na podstawie nowych danych.

3.2 Wykonanie inżynierii cech i eksploracyjnej analizy danych

3.2.1 Brakujące wartości w zbiorze

Na początku sprawdzono brakujące dane w zbiorze

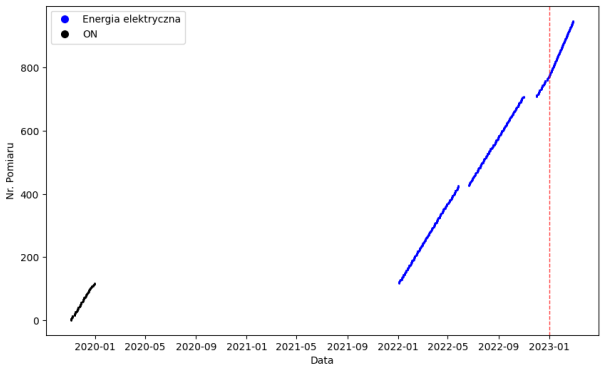
column	nans	percent
Maksymalne, chwilowe zużycie ON [l/h]	831	87.66
Dzienne zużycie ON [l/8h]	831	87.66
Cena hurtowa ON 1000l [PLN]	861	90.82
Maksymalne, chwilowe zużycie energii elektrycznej [kW]	117	12.34
Dzienne zużycie energii elektrycznej [kW/8h]	117	12.34
Cena energii elektrycznej [kWh]	117	12.34
Koszt dzienny [PLN]	30	3.16

Rysunek 1: Brakujące wartości

Braki na temat danych o zużyciu energii elektrycznej i zużyciu ON sumują się do 100% zbioru, wynika to z tego, że dane te miały za zadanie porównać 2 rodzaje silników wykorzystywane przy pracy dźwignicy.
W zbiorze mamy brak 30 rekordów związanych z ceną hurtową ON w danym dniu oraz wynikającego z tej ceny kosztu dziennego.

3.2.2 Częstotliwość pomiarów

Zapoznano się z częstotliwością zbierania danych w czasie



Rysunek 2: Przyrost pomiarów w czasie

Dane układają się w 4 widoczne okresy czasowe w których prowadzone były badania. Zauważono również, że linia po 1 stycznia 2023 roku staje się gładsza, co oznacza, że dane były zbierane wtedy z większą częstotliwością.

Różnice w dniach następujących po sobie danych prezentują się w taki sposób:

	diff	count
0	0.0	632
1	1.0	257
4	2.0	3
2	3.0	49
3	4.0	3
6	25.0	1
7	31.0	1
5	734.0	1

Rysunek 3: Różnice dniowe następujących po sobie danych

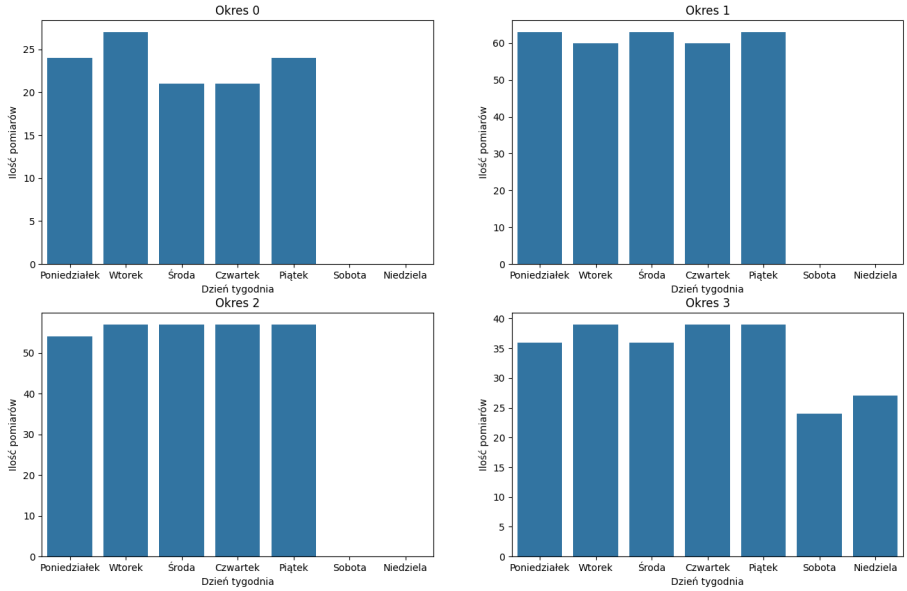
Różnica o wartości 0 wynika z faktu, że dla każdego dnia zgromadzone zostały 3 pomiary i jest to różnica dni między wartościami przypadającymi na ten sam dzień.
Przyjęto, że różnica mniejsza lub równa 4 dni wynika z dni w których pomiary nie były prowadzone z powodów losowych lub dni wolnych (święta, weekendy), natomiast wartości 25 31 oznaczają odstępy w okresach w trakcie których badano układ zasilania elektrycznego a różnica 734 dni to okres między badaniami efektywności ON a energii elektrycznej.

	diff	count
0	0.0	116
1	1.0	57

Rysunek 4: Różnice dniowe po 1 stycznia 2023

Dla danych od 1 stycznia 2023 nie występują wartości większe niż 1. Oznacza to, że dane te musiały być zbierane codziennie.

Następnie dane zostały podzielone na 4 okresy zgodnie z wspomnianym wcześniej podziałem. Ponadto została utworzona kolumna zawierająca dzień tygodnia w którym przeprowadzano dany pomiar i na jej podstawie stworzono wykres przedstawiający rozkład dni tygodnia w danych okresach.



Rysunek 5: Rozkład dni tygodnia w poszczególnych okresach

Zauważono, że jedynie w ostatnim okresie występują dane dla sobót i niedziel, jest ich jednak proporcjonalnie mniej względem innych dni tygodnia ze względu na to, że ostatni okres zawiera w sobie również końcówkę roku 2022 kiedy dane nie były zbierane w dni wolne od pracy.

Następnie zostało sprawdzone, czy przerwy w zbieraniu danych w danych okresach występują w dni robocze w Polsce tzn. Pon-Pt z wyłączeniem dni świątecznych. W tym celu utworzone przedział dni zawierający się od pierwszego do ostatniego dnia danego okresu

```python
import holidays

pl_holidays_dates = pd.Series([pd.Timestamp(date) for date in holidays.Poland(years=range(2018, 2024)).keys()])

for okres in df['okres'].unique():
    data = df[df['okres'] == okres]
    start_date = data['Data'].min()
    end_date = data['Data'].max()
    date_range = pd.date_range(start=start_date, end=end_date)
    missing_dates = date_range.difference(data['Data'])
    missing_dates = missing_dates[missing_dates.dayofweek != 5]
    missing_dates = missing_dates[missing_dates.dayofweek != 6]
    missing_dates = missing_dates[~missing_dates.isin(pl_holidays_dates)]
    print(f'Okres: {okres}, ilość brakujących dni: {len(missing_dates)}, brakujące dni: {missing_dates}')
```

✓ 0.0s

```
Okres: 0, ilość brakujących dni: 0, brakujące dni: DatetimeIndex([], dtype='datetime64[ns]', freq=None)
Okres: 1, ilość brakujących dni: 0, brakujące dni: DatetimeIndex([], dtype='datetime64[ns]', freq=None)
Okres: 2, ilość brakujących dni: 0, brakujące dni: DatetimeIndex([], dtype='datetime64[ns]', freq=None)
Okres: 3, ilość brakujących dni: 0, brakujące dni: DatetimeIndex([], dtype='datetime64[ns]', freq=None)
```

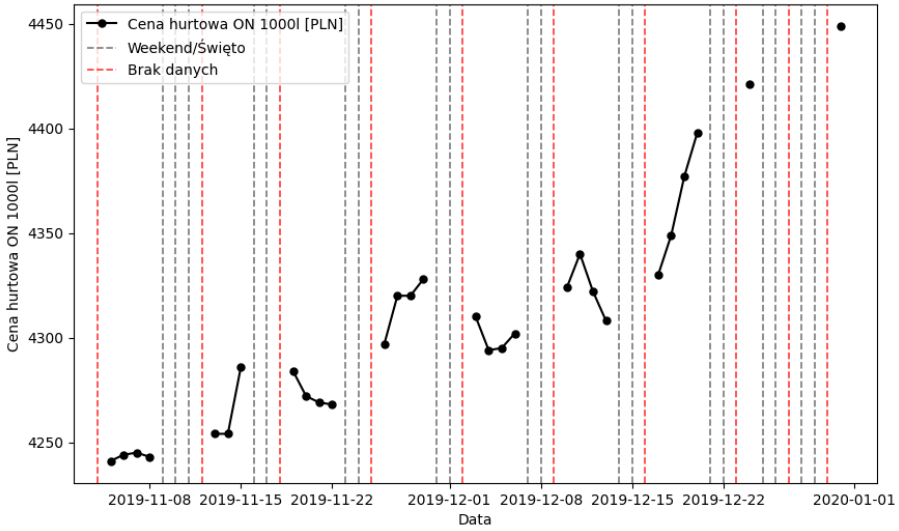
Rysunek 6: Sprawdzenie przerw w ciągłości zbierania danych

Można stwierdzić, że dla każdego z okresów obejmując przedział od pierwszego do ostatniego dnia pomariów nie ma brakujących dni, które nie byłyby dniami wolnymi od pracy. Sytuacja ta zmienia się od 1 stycznia 2023 roku i dane były wtedy zbierane codziennie.

Niestety nie ma informacji na temat dlaczego od 2023 roku dane były zbierane codziennie, czy zostało to w jakiś sposób zautomatyzowane, dane zostały uzupełnione w jakiś inny sposób lub czy też prace były prowadzone w dni ustawowo wolne od pracy.

3.2.3 Uzupełnienie brakujących wartości w zbiorze

Dane zawierają 30 brakujących rekordów na temat hurtowej ceny oleju napędowego w różnych dniach. Z racji, że w zbiorze dla każdego dnia znajdują się po 3 wartości oznacza to, że brakuje wartości ceny w 10 różnych dniach. Wartości te najlepiej byłoby uzupełnić sprawdzając faktyczne ceny oleju napędowego w tych konkretnych dniach, ze względu na to, że są to dane historyczne. Na potrzeby tej analizy zostaną one uzupełnione.



Rysunek 7: Cena hurtowa ON w czasie

Na wykresie przedstawiono trend dla danych na temat cen oleju napędowego, szare przerywane linie to dni dla których nie ma żadnych rekordów i nie były zbierane dane, natomiast czerwone linie to dni w których w rekordach brakuje ceny i należy ją uzupełnić.
Ponadto zauważono, że brakujące rekordy danych w tym okresie występują tylko w dniach, które są dniami po dniach wolnych

	Data	Dzień tygodnia	Czy dzień po dniu wolnym
0	2019-11-04	Poniedziałek	Tak
1	2019-11-12	Wtorek	Tak
2	2019-11-18	Poniedziałek	Tak
3	2019-11-25	Poniedziałek	Tak
4	2019-12-02	Poniedziałek	Tak
5	2019-12-09	Poniedziałek	Tak
6	2019-12-16	Poniedziałek	Tak
7	2019-12-23	Poniedziałek	Tak
8	2019-12-27	Piątek	Tak
9	2019-12-30	Poniedziałek	Tak

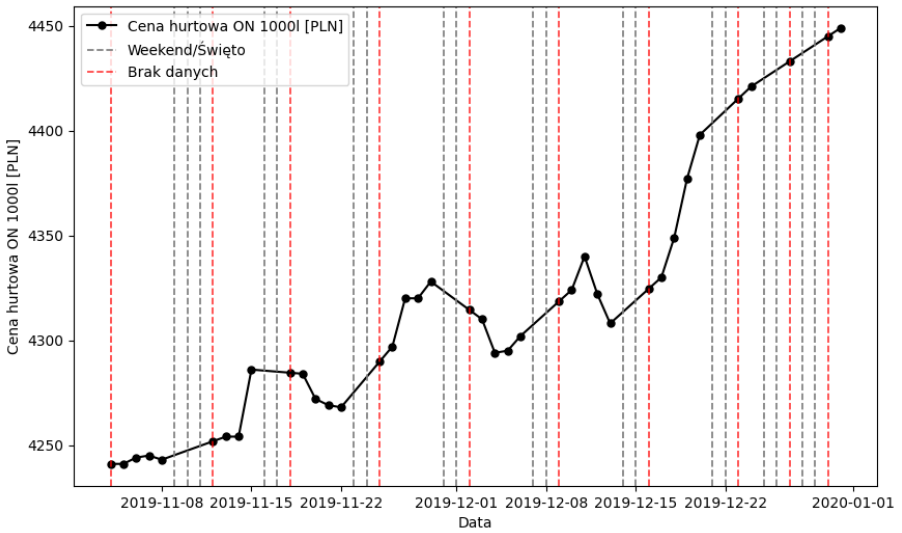
Rysunek 8: Konkretne brakujące dni

Jak można zauważyć większość dni to poniedziałki, które występują po dniu wolnym, 2 pozostałe dni to wtorek 12 listopada, dzień po święcie niepodległości i 27 grudnia dzień po 2 dniu świąt bożego narodzenia, oba te dni (święto niepodległości i 2 dzień świąt bożego narodzenia) to dni ustawowo wolne od pracy.

Dane zostały uzupełnione za pomocą funkcji interpolate przy użyciu metody 'time' z pakietu pandas, założono, że pierwsza wartość jest taka sama jak wartość dzień później.

```python
df_ceny['Cena hurtowa ON 1000l [PLN]'] = df_ceny['Cena hurtowa ON 1000l [PLN]'].interpolate(method='time')
df_ceny['Cena hurtowa ON 1000l [PLN]'] = df_ceny['Cena hurtowa ON 1000l [PLN]'].fillna(method='bfill')

plt.figure(figsize=(10, 6))
plt.plot(df_ceny.index, df_ceny['Cena hurtowa ON 1000l [PLN]'], 'o-', markersize=5, color='black', label='Cena hurtowa ON 1000l [PLN]')
plt.xlabel('Data')
plt.ylabel('Cena hurtowa ON 1000l [PLN]')
plt.legend()

data_range = pd.date_range(start=df_ceny.index.min(), end=df_ceny.index.max())
for date in data_range:
    if date.dayofweek == 5 or date.dayofweek == 6 or date in pl_holidays_dates.values:
        plt.axvline(date, color='grey', linestyle='--', linewidth=1.25, label='Weekend/Święto')

for date in df_braki['Data']:
    plt.axvline(date, color='red', linestyle='--', linewidth=1.25, alpha=0.75, label='Brak danych')

handles, labels = plt.gca().get_legend_handles_labels()
by_label = dict(zip(labels, handles))
plt.legend(by_label.values(), by_label.keys())
✓ 0.1s
```

Rysunek 9: Sposób uzupełnienia brakujących wartości



Rysunek 10: Cena hurtowa ON w czasie - uzupełnione braki

3.3 Wybór proponowanej zmiennej target do modelu ML

W przypadku tego zbioru danych najlepszą zmienną do modelowania (TARGET) wydaje się być zmienna 'Dzienne zużycie energii elektrycznej [kW/8h]'. Analogicznym wyborem byłaby zmienna 'Dzienne zużycie ON [l/8h]', jednak ze względu na zdecydowanie niższy koszt dzienny w przypadku pracy na silniku elektrycznym oraz to, że modernizacja układu napędowego została już przeprowadzana i wykorzystywany jest silnik elektryczny do pracy dźwignicy to przewidywanie zużycia ON wydaje się być pozbawione sensu.

Niestety zmienna ta może stanowić problem w momencie uczenie modelu i odnajdywania jakichkolwiek powiązań z innymi zmiennymi przez swoją niezwykle małą kardynalność. A także w momencie przewidywania, ponieważ model przewidywałby jedną z tych 3 wartości, co najprawdopodobniej nie byłoby faktycznym zużyciem energii elektrycznej przez dźwignicę.

```
Kardynalność zmiennej 'Dzienne zużycie energii elektrycznej [kW/8h]': 3

Dzienne zużycie energii elektrycznej [kW/8h]
79.2    588
80.1    183
80.0     60
Name: count, dtype: int64
```

Rysunek 11: Kardynalność zmiennej 'Dzienne zużycie energii elektrycznej [kW/8h]'

Ponadto w przypadku zmiennej 'Dzienne zużycie ON [l/8h]' również pojawia się ten sam problem, kardynalność tej zmiennej jest niewiele większa.

```
Kardynalność zmiennej 'Dzienne zużycie ON [l/8h]': 5

Dzienne zużycie ON [l/8h]
48.0    42
47.2    33
48.8    33
46.4     6
49.6     3
Name: count, dtype: int64
```

Rysunek 12: Kardynalność zmiennej 'Dzienne zużycie ON [l/8h]'

3.4 Wybór podzbioru zmiennych features do wyznaczenia zmiennej target

Jako zmienne objaśniające (FEATURES) najlepsze wydają się być zmienne które oznaczają fizyczne parametry pracy dźwigu i warunki pogodowe, czyli: 'Ciężar ładunku [T]', 'Długość wysięgnika [m]', 'Odległość od osi [m]', 'Wysokość podnoszenia [m]', 'Prędkość wiatru [m/s]', 'Temperatura [C]', 'Ciśnienie [hPa]'.
Korzystając z tych zmiennych można by przewidywać a następnie próbować optymalizować zużycie energii elektrycznej w zależności od warunków pogodowych oraz tego jaki ładunek i na jaką wysokość należy wynieść.

Kolejnym problemem jest to, że zmienne określające warunki pogodowe są wpisane jako wartości dla całego 8 godzinnego czasu pracy, prawdopodobnie może to być wartość średnia jednak ze względu na to, że warunki pogodowe zmieniają się w ciągu dnia tracimy wiele informacji na temat tego jak wpływają na zużycie prądu