Dokumentacja

część 2

Grzanka Karolina, Proniewicz Jan, Rams Gabriela, Sekuła Zuzanna

2020

DOKUMENTACJA ETAP 0 – STWORZENIE GRUP

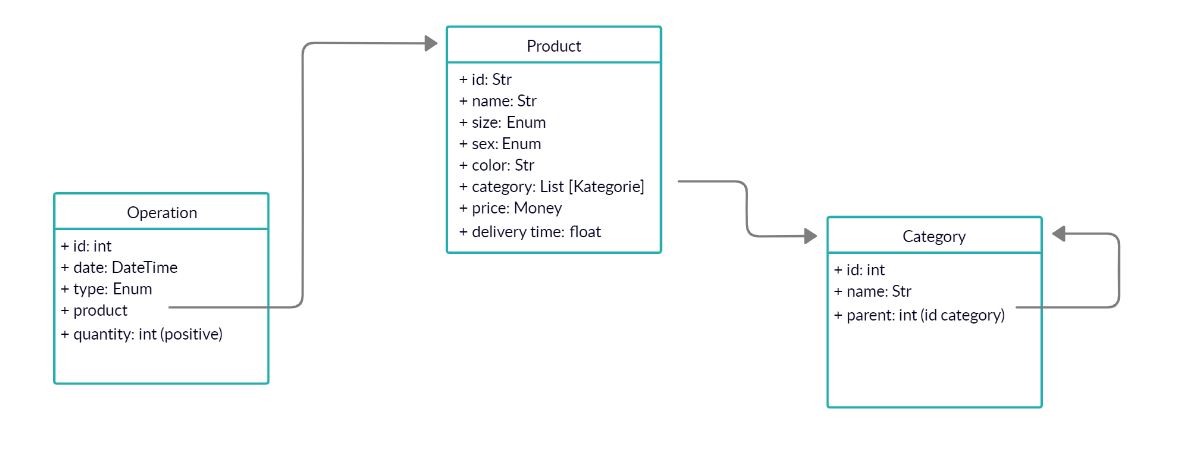
Zerowym etapem było utworzenie zespołów – przydzielenie do grup projektowych. Do tej grupy projektowej zostały przydzielone cztery osoby: Karolina Grzanka, Jan Proniewicz, Gabriela Rams, Zuzanna Sekuła. W tym etapie został wybrany lider grupy, którym została Karolina Grzanka. Pierwszym napotkanym problemem były wstępne rozmowy nad tematem.

DOKUMENTACJA ETAP 1 – SPECYFIKACJA WYMAGAŃ

Pierwszym etapem projektowania systemu wspomagania decyzji było ustalenie specyfikacji wymagań. Dotyczyło to ustalanie jakiego zagadnienia będzie dotyczył projekt. Grupa wykonująca projekt po długiej rozmowie doszła do porozumienia. Ustalone zostało, iż tematem pracy grupowej będzie stworzenie systemu wspomagania decyzji w sklepie internetowym. Zadanie takiego programu to między innymi analiza stanu magazynu oraz wyniki sprzedaży sklepu. Założone zostało, że program będzie pokazywał ile produktów danego rodzaju lub kategorii jest obecnie na składzie, ile udało się sprzedać w ciągu ostatniego okresu (rok, miesiąc itp.) oraz wyliczenie prognozy (ile prawdopodobnie uda się sprzedać) na następny okres. Oprócz tego, program miał pozwalać on na porównanie wyników sprzedaży z dwóch różnych okresów oraz informować o ewentualnych nadwyżkach lub deficytach produktów.

**Główny cel projektu**: system który pozwala analizować dowolny sklep, w którym produkty posiadają trzy podstawowe cechy: kolor, płeć oraz rozmiar.

Aby zrealizować cel prjektu, założone zostało, iż dane muszą posiadać prezentowaną poniżej strukturę.



Operacje posiadają następujące charakterystyczne cechy:

* Numer identyfikacyjny każdej operacji
* Data, w której dana operacja została wykonana
* Typ operacji czyli sprzedaż, dostawa, zwrot
* Produkt, który w danej operacji został sprzedany, dostarczony bądź zwrócony
* Ilość, która oznacza ile produktów zostało sprzedanych, dostarczonych czy zwróconych

Produkty posiadają następujące charakterystyczne cechy:

* Numer identyfikacyjny produktu
* Nazwę produktu
* Rozmiar produktu
* Płeć dla której przeznaczony jest produkt
* Kolor produktu
* Kategoria do której należy produkt
* Ile kosztuje dany produkt
* Jaki jest czas dostawy danego produktu

Kategorie posiadają następujące charakterystyczne cechy:

* Numer identyfikacyjny kategori
* Nazwa danej kategorii
* W przypadku gdy kategoria posiada rodzica w postaci innej kategorii nadrzędnej będzie posiadała numer odwołujący do id

Określona została funkcjonalność przy pomocy metody MoSCow. Wyróżnione zostały zagadnienia, które koniecznie muszą zostać zrealizowane w projekcie, które powinny zostać zrealizowane w projekcie i te które są opcjonalne.

Funkcje **MUST** – czyli niezbędne w naszym projekcie

* Możliwość wczytania danych
* Możliwość wyświetlenia wykresu pokazującego stan magazynu (dla danego produktu)
* Możliwość wyświetlenia wykresu pokazującego stan magazynu (dla danej kategorii)
* Możliwość wyliczenia sprzedaży (przychodu) na dany okres czasu (z podziałem na kategorie)
* Prognoza sprzedaży na podstawie danych historycznych

Funkcje **SHOULD** - czyli powinny znaleźć się w projekcie

* Możliwość zbadania (przeanalizowania) który kolor, rozmiar sprzedaje się najlepiej
* Możliwość porównania sprzedaży danego produktu dla dwóch okresów
* Możliwość porównania sprzedaży danej kategorii dla dwóch okresów
* Wyświetlanie ostrzeżeń o konieczności zamówienia dostawy

Funkcje **COULD** - czyli opcjonalne

* Możliwość analizy na jaki okres czasu wystarczą produkty znajdujące się w magazynie
* Możliwość porównania stanu zarejestrowanego (z programu) z rzeczywistym stanem (po inwentaryzacji) magazynu

Specyfikacja wymagań projektu wymagała również ustalenia jak zamierza się zrealizować dany projekt. Ustalone zostało, iż najlepszym możliwym rozwiązaniem będzie wykonanie systemu wspomagania decyzji w języku Python, ze względu na wszechstronność i dostępność wielu bibliotek. Na potrzeby projektu dane w założeniu powinny zostać wygenerowane i przechowywane w formacie CSV.

DOKUMENTACJA ETAP 2 – HARMONOGRAM I JEGO REALIZACJA

Harmonogram został stworzony 27.10.2019 przez grupę dzięki konsultacjom i rozmowom. Praca nad programem została podzielona na konkretne części i zadania, które powinny być zrealizowane w konkretnym terminie.

**21.10.2019 - 27.10.2019** 🡪 Zgodnie z harmonogramem na ten okres zostały zrealizowane zadania jakimi było przygotowanie samego harmonogramu prac i przygotowanie schematu baz danych.

**28.10.2019 - 03.11.2019** 🡪 W tym okresie zostały zrealizowane również prawie wszystkie zadania: praca nad klasą Warehouse, przygotowanie wzorów służących do analizy danych, wygenerowanie oraz przygotowanie baz danych. Mimo tego, wielkim problemem okazało się dobranie odpowiednich danych. Trzeba było opóźnić prace nad dalszymi etapami projektu. Okazało się, iż nie jest łatwo wygenerować dane które byłyby odpowiednio realistyczne.

**04.11.2019 – 10.11.2019** 🡪 Wyznaczone zadania - funkcja pozwalająca wczytanie danych oraz przetwarzająca dane na obiekty klasy Warehouse zostały zrealizowane. Napotkane zostały inne problemy dotyczące zmienności ceny w czasie oraz wyboru metody prognozy. Mimo tego, wielkim problemem okazało się wykorzystywanie danych. Trzeba było opóźnić prace nad dalszymi etapami projektu. Okazało się, iż nie jest łatwo wygenerować dane które byłyby odpowiednio realistyczne. Poprawa danych z zeszłego tygodnia prac nie została dokończona lecz dalej trwała.

**11.11.2019 – 17.11.2019** 🡪 Ten etap nie został zrealizowane zgodnie z czasem. Nieprzewidziany, trudniejszy czas na studiach zakłócił realizowanie harmonogramu. Zrealizowane zostały zadania z poprzednich tygodni dotyczące stworzenia repozytorium na GitHub, funkcja pozwalająca wyznaczyć stan magazynu dla danego produktu oraz dla danej kategorii. Dane zostały ukończone.

**18.11.2019 – 24.11.2019** 🡪 Początkowo etap rozpoczął się źle, gdyż dane dobrane do poszczególnych operacji i produktów nie spełniały podstawowych zasad. W związku z tym prace zostały podzielone na wszystkie osoby. Zorganizowane zostało spotkanie wirtualne w celu reorganizacji harmonogramu i zadań. Została stworzona funkcja pozwalająca wyznaczyć stan magazynu dla danego produktu oraz dla danej kategorii.

**25.11.2019 – 01.12.2019** 🡪 Całkowita reorganizacja harmonogramu w związku z ciągnącym się problemem danych. Karolina Grzanka przejęła większość funkcji całego projektu. Następujące funkcje zostały zrealizowane: wyliczające przychód ze sprzedaży na dany okres czasu, wizualizacja za pomocą wykresu funkcji pokazujących stan magazynu, funkcja pozwalająca na porównanie sprzedaży danego produktu lub kategorii dla dwóch okresów. Przesunięciu ulęgnęło przeprowadzenie testów funkcji MUST, dokumentacja z wykonanego etapu pracy, funkcja pozwalająca na przeprowadzenie analizy, który kolor, rozmiar sprzedaje się najlepiej oraz funkcja pozwalająca wyświetlić ostrzeżenie o konieczności zamówienia dostawy.

**02.12.2019 – 08.12.2019** 🡪 W tym tygodniu prace uległy znacznej poprawie. Została podjęta decyzja o wygenerowaniu danych od początku na nowo. Zrealizowano następujące funkcje z poprzednich tygodni: przeprowadzenie testów funkcji MUST – część na dany tydzień, funkcja pozwalająca na przeprowadzenie analizy, który kolor, rozmiar sprzedaje się najlepiej. Prace nad wyświetlaniem otrzymanych wyników zostały zakończone na dany etap. Przygotowana została prezentacja dla reszty grupy w celu pokazania zagadnienia innym osobom. Z przyszłego tygodnia zrealizowana została funkcja pozwalająca na porównanie stanu zarejestrowanego z rzeczywistym (po inwentaryzacji).

**09.12.2019 – 15.12.2019** 🡪 Zostały zakończone prace nad wyświetlaniem otrzymanych wyników, przeprowadzenie testów funkcji, funkcja pozwalająca wyświetlić ostrzeżenie o konieczności zamówienia dostawy, dokumentacja etapu MUST i SHOULD.

**16.12.2019 – 05.01.2020 🡪** Funkcja COULD jaką jest możliwość analizy na jaki okres czasu wystarczą produkty znajdujące się w magazynie, została dokończona. Funkcja wyświetlająca ostrzeżeń o konieczności zamówienia dostawy należąca do SHOULD została zrealizowana. Nastąpiła zmiana formatu daty z dd-mm-YY na YY-mm-dd. Zostało stworzone GUI obrazujące działanie programu.

HARMONOGRAM - ZAŁOŻENIA

W poniższej tabelce przedstawione zostaną zadania, które **miały być zrealizowane** przez daną osobę z grupy w określonym przedziale czasowym.



HARMONOGRAM - REALIZACJA

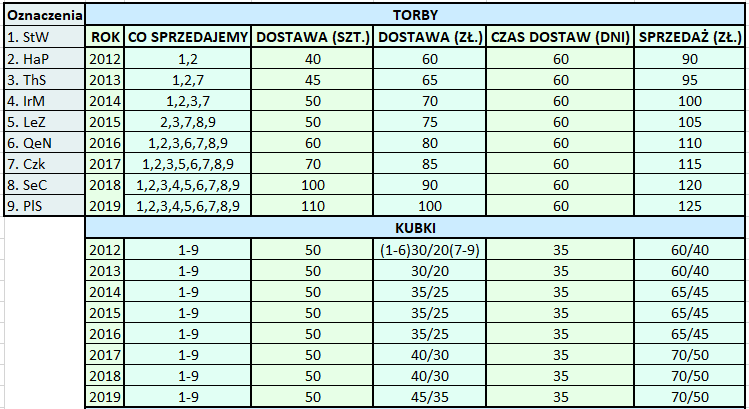
W poniższej tabelce przedstawione zostaną zadania, które zostały **zrealizowane** przez daną osobę z grupy w określonym przedziale czasowym.

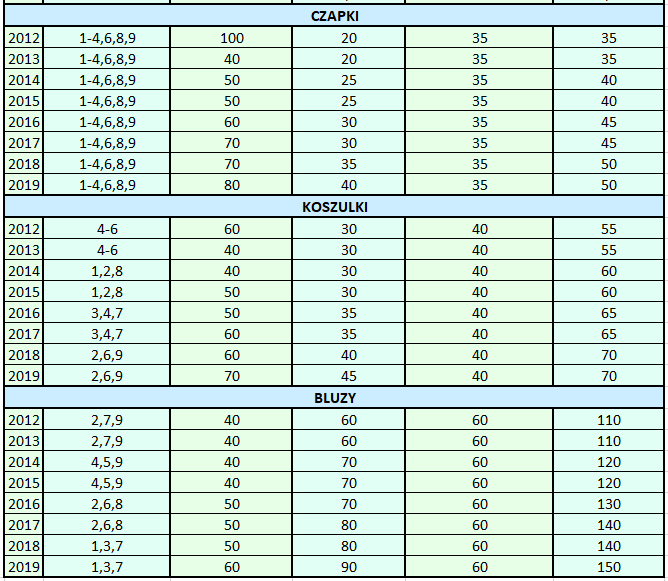


DOKUMENTACJA ETAP 3,4 – WZORY I KODOWANIE

DANE

Dużym problemem z realizacją tego etapu była generacja danych, na których opiera się projekt. Ostateczne dane zostały wygenerowane według poniższego schematu.





Dane zostały zapisane w pliku CSV. Poniżej przedstawiony jest przykładowy fragment dla produktów, kategorii oraz operacji wykonywanych na danych produktach.

**Products**



**Categories**



**Operations**



Z zakładanej początkowo bazy danych przeniesiona została własność opisująca produkt jaką jest cena **price** z Product do Operation. Przyczyną tej zmiany jest własność ceny, która może się zmieniać w czasie. Dzięki temu można również uwzględniać promocje.

PROGNOZA

Innym problemem było ustalenie jakim sposobem będzie się odbywała prognoza. Klasyczna metoda regresji liniowej mogłaby zaburzać rzeczywistość. Po głębokiej analizie metod prognozowania, uznane zostało, iż jednym sposobem będzie prognozowanie dzięki metodzie wskaźników sezonowych.

Prognoza w systemie wspomagania decyzji sklepu jest istotnym zagadnieniem. Głównym celem projektu jest analiza stanu magazynu i prognoza na podstawie danych historycznych. Aby system poprawnie analizował dane konieczne było dobranie odpowiedniej metody prognozowania. Dane, które zostały wygenerowane na potrzeby projektu charakteryzują wahania sezonowe. W okresie świątecznym, walentynkowym, Hallowen i zimowym sprzedawane są produkty w większej ilości, charakterystycznej dla danego okresu czasu. Sprzedaż akcesoriów zimowych praktycznie nie istnieje w lecie, co wydaje się być logiczne pod względem pogody i zapotrzebowani na takie produkty. Aby zanalizować wahania sezonowe najlepszą okazuje się metoda wskaźników sezonowych. W tej metodzie wyznaczane są wskaźniki sezonowości dla poszczególnych faz cykli. Prognozować można na dwa różne sposoby:

* Model addytywny

Model ten jest stosowany jeżeli amplitudy wahań określane jako różnice miedzy rzeczywistymi wartościami prognozowanej **y** a  wartościami teoretycznymi uzyskanymi z obliczeń w modelu tendencji rozwojowej, w analogicznych fazach cyklu są mniej więcej  takie same, mamy do czynienia z **wahaniami bezwzględnie stałymi.**

* Model multiplikatywny

Model ten jest stosowany gdy wielkości amplitud ulegają zmianom  w pewnej prawie równej proporcji, mówimy  wtedy o **wahaniach względnie stałych.**

Procedura konstrukcji prognozy jest następująca:

**Krok 1:** Identyfikacja cykli i faz.

**Krok 2:** Opracowanie prognozy wstępnej jako przypuszczanej tendencji rozwojowej z oszacowaniem parametrów za pomocą KMNK.

**Krok 3:** Wyznaczenie wartości wskaźników sezonowości jako ilorazu (model multiplikatywny) lub różnicy (model addytywny) wartości rzeczywistych i wartości teoretycznych.

**Krok 4:** Wyznaczenie surowych wskaźników sezonowości w celu wyeliminowania wahań przypadkowych we wskaźnikach przez wyznaczenie średniej arytmetycznej tych wartości które odpowiadają jednoimiennym okresom.

**Krok 5:** Obliczenie średniej arytmetycznej surowych wskaźników sezonowości.

**Krok 6:** Wyznaczenie czystych wskaźników sezonowości jako ilorazów (model multiplikatywny) lub różnic (model addytywny) surowych wskaźników sezonowości i wielkości.

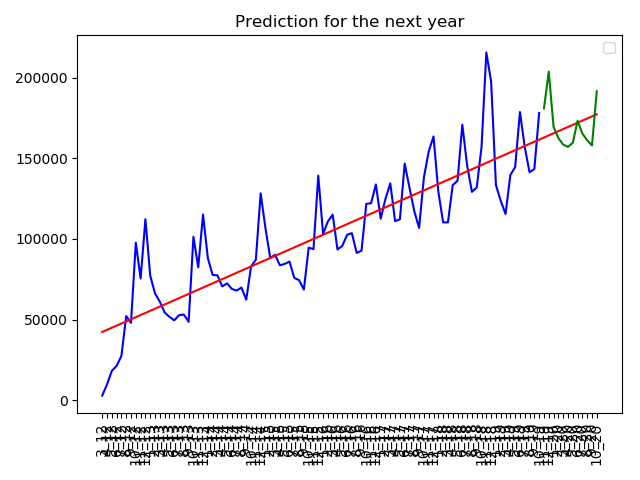
Kolejny problem w tworzeniu systemu wspomagania decyzji to dobór odpowiedniego trendu, gdyż jest on kolejnym istotnym elementem prognozy. W projekcie, wykorzystany został trend liniowy. Najprostszą i zarazem najpowszechniejszą postacią funkcji trendu jest funkcja liniowa, **yt=α+βt** która przedstawia funkcję prostej **α**, wyznaczony parametr jest właśnie współczynnikiem kierunkowym tej prostej. Parametr **β** prezentuje stały przyrost wartości zmiennej prognozowanej **y** w ciągu roku, miesiąca, kwartału , dnia lub jednostki czasu danego szeregu. Nie zawsze model liniowy dobrze odzwierciedla zmiany w czasie zjawiska jakim jest sprzedaż.

**Czerwona linia** prosta to trend liniowy. Na poniższym wykresie widać, iż trend liniowy może nie być odpowiedni do właściwego prognozowania.

**Zielona linia** przedstawia prognozę na przyszłe okresy czasu.

**Niebieska linia** przedstawia dane historyczne sklepu w którym będzie wykorzystany system wspomagania decyzji.

Wygodna dla przyszłego użytkownika programu jest możliwość wyboru miesięcznych bądź kwartalnych danych historycznych. Prognoza wykonana tą metodą prezentuje się następująco:



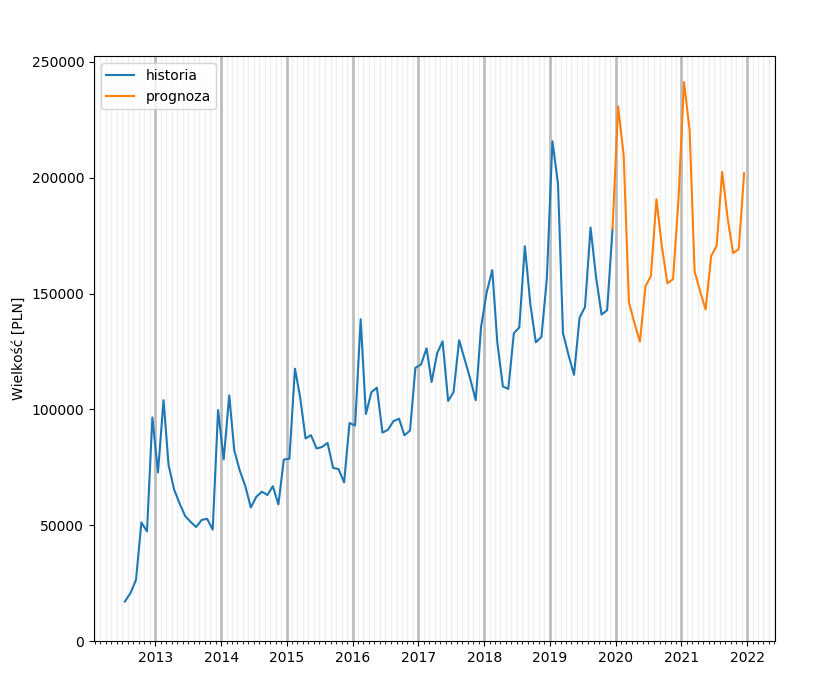
months

Oprócz klasycznej metody prognozowania w projekcie została stworzona inna metoda prognozowania jaką jest SARIMA. Jest to metoda służąca prognozowaniu szeregów czasowych. Ponieważ dane w projekcie zawierają elementy sezonowości zamiast modelu ARIMA (Auto Regressive Integrated Moving Average) korzysta się z modelu SARIMA (Seasonal ARIMA).

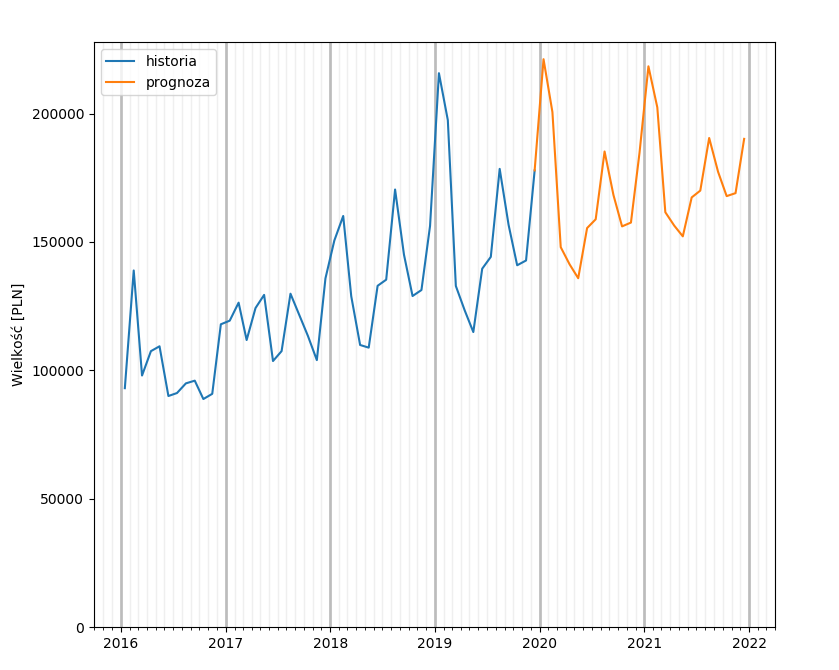
Można podzielić modelowanie szeregów czasowych na etapy:  
**Krok 1:** Sprawdzanie stacjonarności: Jeśli szereg czasowy ma składową trendu lub sezonowość, musi on być stacjonarny, zanim będzie można użyć ARIMA/SARIMA do prognozowania.  
**Krok 2:** Różnica: Jeśli szeregi czasowe nie są stacjonarne, muszą być stacjonarne poprzez różnicowanie.   
**Krok 3:** Podzielenie danych na testowe i treningowe.  
**Krok 4:** Wybranie elementów AR i MA: Użycie ACF i PACF, aby zdecydować, czy uwzględnić AR, MA lub oba.  
**Krok 5:** Zbudowanie modelu i ustawienie liczby okresów do prognozy na N.  
**Krok 6:** Sprawdzenie poprawności modelu: porównanie przewidywanych wartości z rzeczywistymi.

W programie korzystając z tej prognozy, można zdecydować czy chce się prognozować przychody w sklepie internetowym, czy ilość sprzedanych produktów. Te dwie opcje mogą być wybrane osobno dla każdego produktu, dla wybranej kategorii itd. Wygodnym elementem tej prognozy jest dobór długości okresu historycznego na podstawie którego się prognozuje, i wybór długości okresu przyszłego który jest prognozowany.

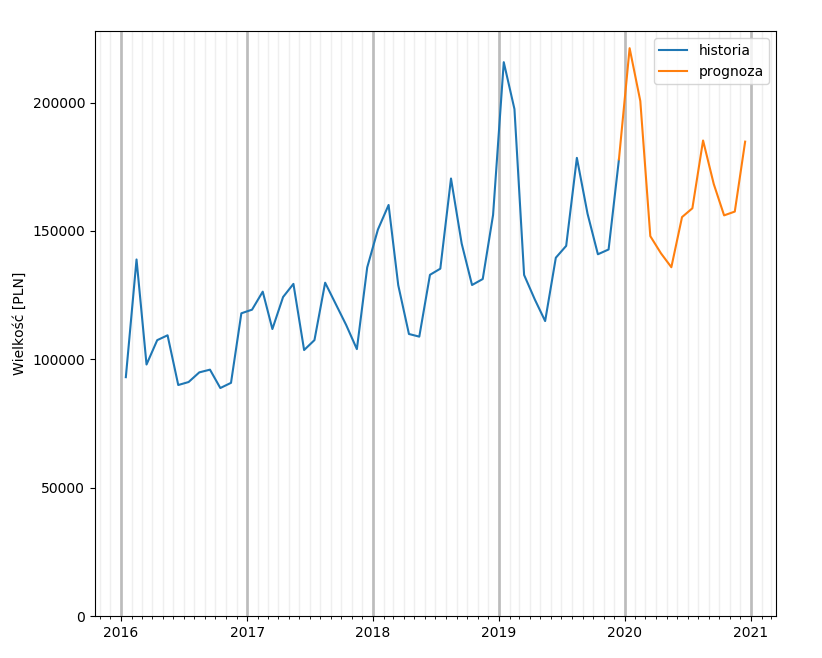
Poniżej przedstawione są wyniki – wykresy obrazujące modele prognozy SARIMA. Wykres prognozy wielkości sprzedaży ogółem na przyszłe 24 miesiące, sporządzony na podstawie 90 miesięcy danych historycznych.



Wykres prognozy wielkości sprzedaży ogółem na przyszłe 2 lata, sporządzony na podstawie 4 lat danych historycznych.



Wykres prognozy wielkości sprzedaży ogółem na przyszły rok, sporządzony na podstawie 4 lat danych historycznych.

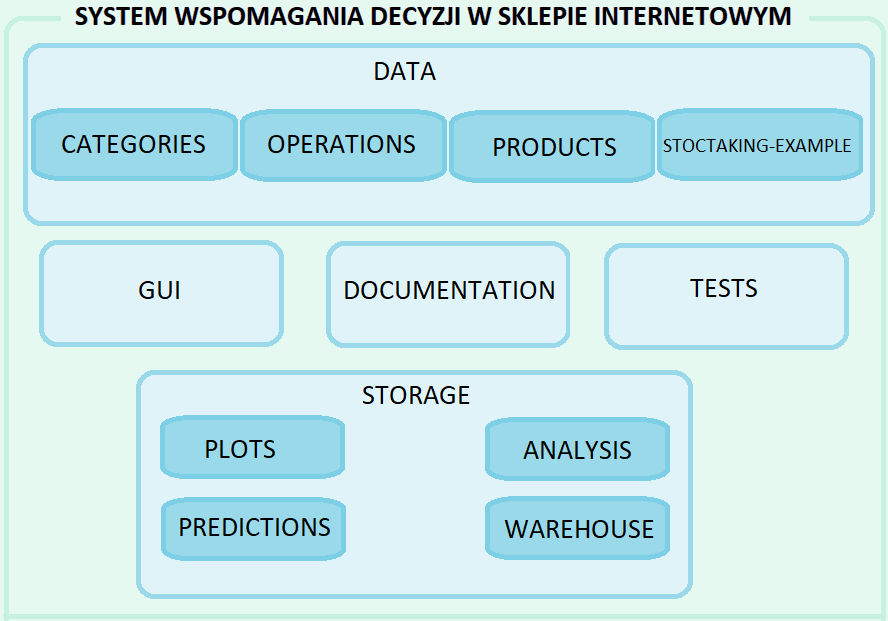


FUNKCJE

Projekt zakładał wykonanie funkcji MUST, SHOULD oraz COULD. Do zakończenia etapu trwającego do 15 grudnia zostały zrealizowane następujące funkcje zaznaczone na zielono, a pozostałe zaznaczone na niebiesko zostały dokończone do 5 stycznia. Funkcja dotycząca wyświetlania ostrzeżeń o konieczności zamówienia dostawy, ze względu na złożoność wylicza zbyt długo, przez co nie jest używana w projekcie.

|  |  |  |
| --- | --- | --- |
| **MUST** – niezbędne w projekcie | **SHOULD** – te, które powinny znaleźć się w projekcie | **COULD** - opcjonalne |
| Możliwość wczytania danych | Możliwość zbadania (przeanalizowania) który kolor, rozmiar sprzedaje się najlepiej | Możliwość analizy na jaki okres czasu wystarczą produkty znajdujące się w magazynie |
| Możliwość wyświetlenia wykresu pokazującego stan magazynu (dla danego produktu) | Możliwość porównania sprzedaży danego produktu dla dwóch okresów | Możliwość porównania stanu zarejestrowanego (z programu) z rzeczywistym stanem (po inwentaryzacji) magazynu |
| Możliwość wyświetlenia wykresu pokazującego stan magazynu (dla danej kategorii) | Możliwość porównania sprzedaży danej kategorii dla dwóch okresów |  |
| Możliwość wyliczenia sprzedaży (przychodu) na dany okres czasu (z podziałem na kategorie) | Wyświetlanie ostrzeżeń o konieczności zamówienia dostawy |  |
| Prognoza sprzedaży na podstawie danych historycznych |  |  |

STRUKTURA PROGRAMU



W pliku dotyczącym magazynu 🡪 „**warehouse.py**” zostały utworzone następujące klasy:

* Category
* Size
* Sex
* Product
* OperationType
* Warehouse (klasa, która przechowuje i zarządza danymi)

Plik dotyczący prognozy 🡪 „**predictions.py**” zawiera klasyczną metodę prognozowania dzięki szeregom czasowym oraz z wykorzystaniem trendu liniowego. Zawiera następujące funkcje:

* only\_sales\_for\_product (Funkcja wybiera z listy operacji tylko te dotyczące sprzedaży oraz ewentualnie określonego produktu)
* sales\_sum (Sumuje ilość sprzedanych produktów oraz tworzy dane historyczne)
* linear\_trend\_parameters (Funkcja obliczająca parametry do trendu liniowego)
* seasonal\_indicators\_intro (Funkcja licząca wskaźniki sezonowości dla poszczególnych okresów)
* cleaning\_indicators (Funkcja obliczająca wskaźniki surowe i oczyszczone)
* counting\_prediction (Funkcja obliczająca prognozę na następne 12 miesięcy / 4 kwartały)
* prediction\_plot (Funkcja rysująca wykres prognozy)

Plik dotyczący analizy 🡪 „**analysis.py**” zawiera wiele funkcji zwracających wybrane przez użytkownika informacje:

* get\_statuses (Zwraca status produktów spełniających kryteria)
* stock\_status\_for\_product (Zwraca ilość podanego produktu w magazynie)
* get\_product\_operations (Zwraca wszystkie operacje dla podanego produktu)
* **get\_products** (Zwraca wszystkie produkty z magazynu które spełniają podane warunki. Jeżeli któryś warunek nie zostanie podany to z domysłu akceptuje on wszystko)
* get\_income (Zwraca przychód na dany domknięto-otwarty okres)
* get\_costs (Zwraca koszty na dany domknięto-otwarty okres)
* get\_sales (Zwraca ilość sprzedanych towarów na dany domknięto-otwarty okres)
* get\_ressuply (Zwraca ilość namówionych towarów na dany domknięto-otwarty okres)
* get\_balance (Zwraca bilans na dany domknięto-otwarty okres)
* get\_products\_balance (Zwraca bilans produktów na dany domknięto-otwarty okres)
* get\_best\_selling\_colors (Zwraca posortowaną ilość sprzedanych sztuk w danym okresie dla poszczególnych kolorów)
* get\_best\_selling\_sizes (Zwraca posortowaną ilość sprzedanych sztuk w danym okresie dla poszczególnych rozmiarów)
* load\_stocktaking (Wczytuje plik z inwentaryzacji)
* compare\_with\_stocktaking (Zwraca porównanie aktualnego stanu magazynu ze stanem z inwentaryzacji. Inwentaryzacja w formie pliku CSV, gdzie pierwsza kolumna to id produktu a druga to zliczona ilość danych produktów)
* get\_monthly\_incomes (Zwraca przychody na przestrzeni ostatnich miesięcy)
* get\_monthly\_sales (Zwraca sprzedaże na przestrzeni ostatnich miesięcy)
* forecast\_values (Funkcja prognozy SARIMA, która dopasowuje model, przewiduje wartości)
* get\_months\_for\_supplies

Najważniejsza funkcja z punktu widzenia programu została pogrubiona (get\_products). Jest ona najczęściej wykorzystywana, w szczególności do rysowania wykresów.

Plik dotyczący wykresów 🡪 „**plots.py**” zawiera funkcji potrzebne do wyświetlania danych na wykresach:

* skróty do wykresów:
* plot\_stock\_for\_category
* plot\_stock\_for\_product\_prefix
* kryteria według których rysowane są wykresy:
* plot\_stock\_by\_color (Wyswietla wykresy (po kolorach) statusu produktów spełniajacych podane kryteria)
* plot\_stock\_by\_size (Wyswietla wykresy (po rozmiarach) statusu produktów spełniajacych podane kryteria)
* plot\_yearly\_balance (Wyświetla wykres kosztów i dochodów na przestrzeni lat)
* plot\_yearly\_products\_balance (Wyświetla wykres dostaw i sprzedaży na przestrzeni lat)
* plot\_income\_periods (Wyświetla dochód dla poszczególnych okresów)
* plot\_costs\_periods (Wyświetla koszty dla poszczególnych okresów)
* plot\_sales\_periods (Wyświetla ilość sprzedanych towarów dla poszczególnych okresów)
* plot\_resupply\_periods (Wyświetla ilość zamówionych przedmiotów dla poszczególnych okresów)
* plot\_balance\_periods (Wyświetla balans dla poszczególnych okresów)
* plot\_products\_balance\_periods (Wyświetla dochód dla poszczególnych okresów)
* plot\_forecast\_income (Wyświetla prognozę na podstawie danych)
* plot\_forecast\_sales (Wyświetla prognozę na podstawie danych)
* użyteczność:
* plot\_error (Wyświetla pusty wykres z tekstem na środku)

DOKUMENTACJA ETAP 5 – TESTY

Testy jednostkowe funkcji programu zostały przeprowadzone przy użyciu pakietu unittest.

Na początku zostały przygotowane dane testowe (**categories\_test.csv, products\_test.csv, operations\_test.csv, stocktaking\_test.csv** – w porównaniu do danych prawidłowych posiadają mniejszą liczbę wierszy).

Wyniki dla poszczególnych funkcji zostały obliczone w programie Excel i zawarte w pliku tests.xslx. Testy zostały przeprowadzone dla prawie wszystkich funkcji. Funkcje pozwalające na wczytywanie plików w formacie csv zostały przetestowane dopiero w testach GUI, gdyż tam zostały napisane wyjątki, które wyłapują błędy takie jak: nieprawidłowy format, czy ścieżka.

Testy zostały przeprowadzone zgodnie z ustalonym wcześniej planem:

**Krok 1:** Sprawdzenie działania funkcji przy podaniu prawidłowych argumentów.

**Krok 2:** Sprawdzenie przypadku gdy data ‘OD’ jest większa od daty ‘DO’.

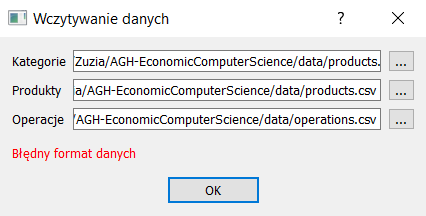
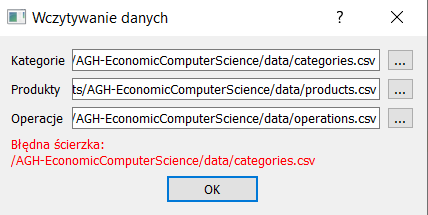
**Krok 3:** Sprawdzenie przypadku gdy liczba miesięcy będzie zerowa lub ujemna.

**Krok 4:** Sprawdzenie przypadku, gdy jako argument zostanie wpisana błędna wartość.

**Krok 5:** Sprawdzenie przypadku, gdy jako wartość danego argumentu zostanie wpisana wartość None.

Kolejnym etapem było przeprowadzenie TESTÓW GUI. Poniżej opisane są jedynie przypadki, w których dana funkcja nie powinna zadziałać.

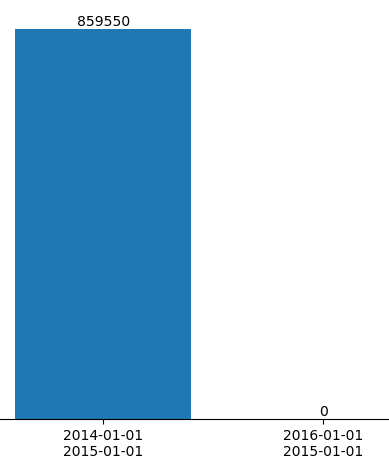
1. Przy wczytywaniu nieprawidłowych danych: Kategorie, Produkty, Operacje, czy później Inwentaryzacja (np. niezgodnych z ustalonym formatem) pojawia się komunikat:



1. Przy uzupełnianiu przez użytkownika filtrów:
2. Jeżeli w polach: Przedrostki ID, Nazwy, Kategorie, Kolory, użytkownik wpisze nieprawidłowe wartości to pojawi się komunikat: „Brak danych spełniających kryteria”
3. Jeżeli w polu Kategorie, użytkownik wpisze występujące w systemie kategorie, jednak będą one z tego samego poziomu np. Bluza, Koszulka to również pojawi się ten komunikat
4. Jeżeli w polach: Rozmiary, Płcie użytkownik wpisze wartości które nie są wyświetlone w ramce to pojawi się komunikat:

lub 

1. W zakładce „Porównywanie okresów”:
2. Jeżeli w polu Okresy data OD będzie większa niż data DO to funkcja zwróci wartość zero:



1. Jeżeli użytkownik wpisze błędny format daty to pojawi się komunikat:



1. W zakładce „Roczny bilans”:
2. Jeżeli ‘Rok od’ będzie większy od ‘Rok do’ to pojawi się komunikat:



1. W zakładce „Dostawy”:
2. Jeżeli nie zostaną uzupełnione żadne kryteria to wynik będzie obliczany bardzo długo
3. Czas wykonywania się funkcji skraca się tym bardziej im więcej filtrów zostanie uzupełnionych przez użytkownika
4. W zakładce „Analiza” oraz „Stan magazynu”:

Użytkownik nie ma możliwości wpisania daty w nieprawidłowym formacie.

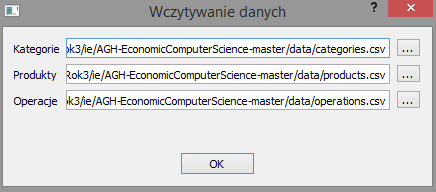
PORADNIK OBSŁUGI PROGRAMU

**Instrukcja obsługi interfejsu GUI.**

Uwaga! Żeby interfejs mógł działać poprawnie, wymagana jest biblioteka *python-dateutil*,oraz wersjA pythona 3.7. Użytkownicy starszej wersji, w tej sytuacji, będą zmuszeni doinstalować pakiet o nazwie *backports-datetime-fromisoformat*, inaczej dane zostaną błędnie odczytane i program ich nie przyjmie.

***Wczytywanie danych:***

Po uruchomieniu programu wyświetli nam się okno wyboru odpowiednich plików csv z danymi dla systemu (kolejno: Kategorie, Produkty i Operacje). Za pomocą przycisków (…) użytkownik może wybrać pliki z komputera. Wewnątrz pól pokażą się wtedy ich ścieżki. Wybór zostaje zatwierdzony o naciśnięciu przycisku OK. Jeśli format danych jest nieprawidłowy, program pokaże ostrzeżenie i odmówi otwarcia następnego okna interfejsu.



***Okno analizy:***

Można powiedzieć, że jest to „główna część” interfejsu GUI. To tutaj użytkownik może wybrać, jakie informacje mają zostać pokazane, na podstawie wprowadzonych wcześniej danych.

Po prawej stronie będą wyświetlane wyniki dla poszczególnych analiz. Najczęściej będą przedstawione w postaci wykresów, z wyjątkiem operacji „Analiza”, „Dostawy” oraz „Inwentaryzacja”.

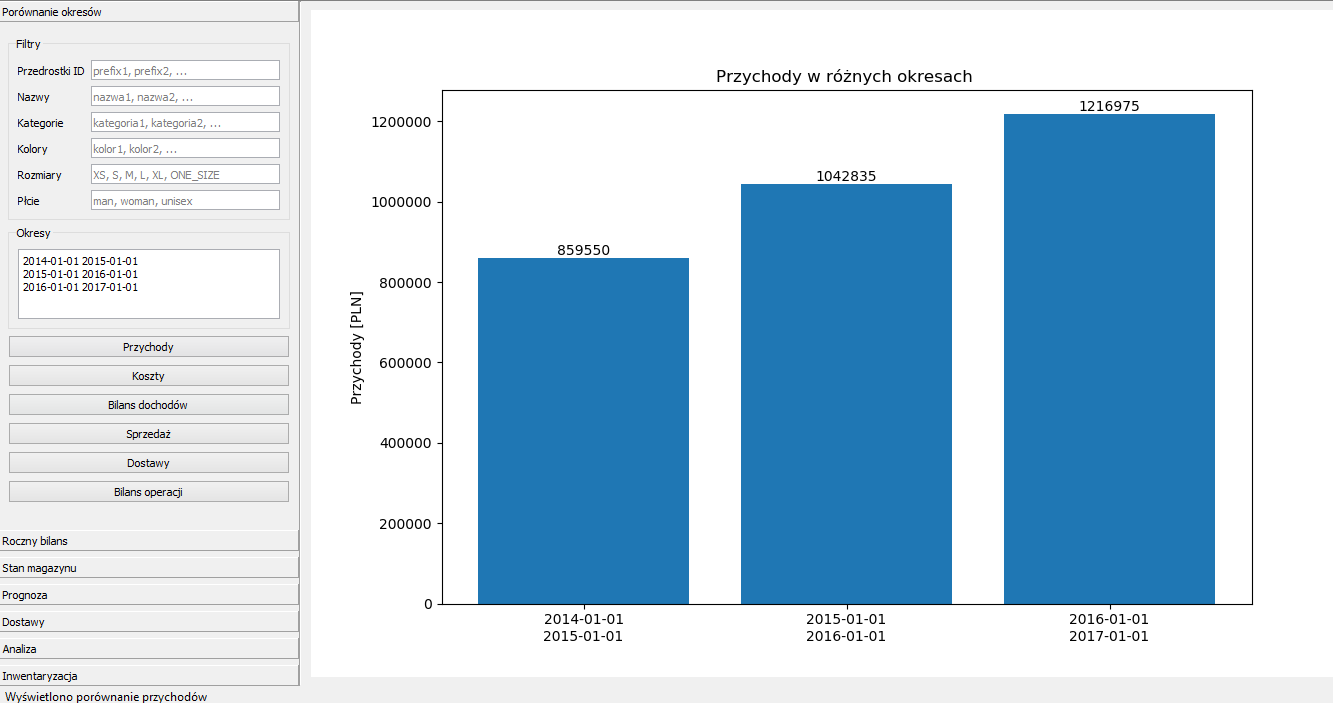
Po lewej stronie można zobaczyć listę dostępnych funkcji systemu decyzyjnego. Po wybraniu jednej z nich, pokaże się odpowiadające jej menu. Każde takie menu składa się z pól do wprowadzania dodatkowych kryteriów oraz przycisków wykonujących przypisane im operacje.

Wszystkie operacje systemu (poza rocznym bilansem, analizą najlepiej sprzedających się produktów i inwentaryzacją) posiadają możliwość filtrowania danych. Jest to przydatne, jeżeli użytkownika interesują jedynie informacje, dotyczące konkretnych produktów, a nie całości. Można filtrować dane wg: przedrostka ID (czyli początkowych znaków identyfikatora lub całości id), pełnej nazwy produktu, jego kategorii, koloru, rozmiaru lub płci.

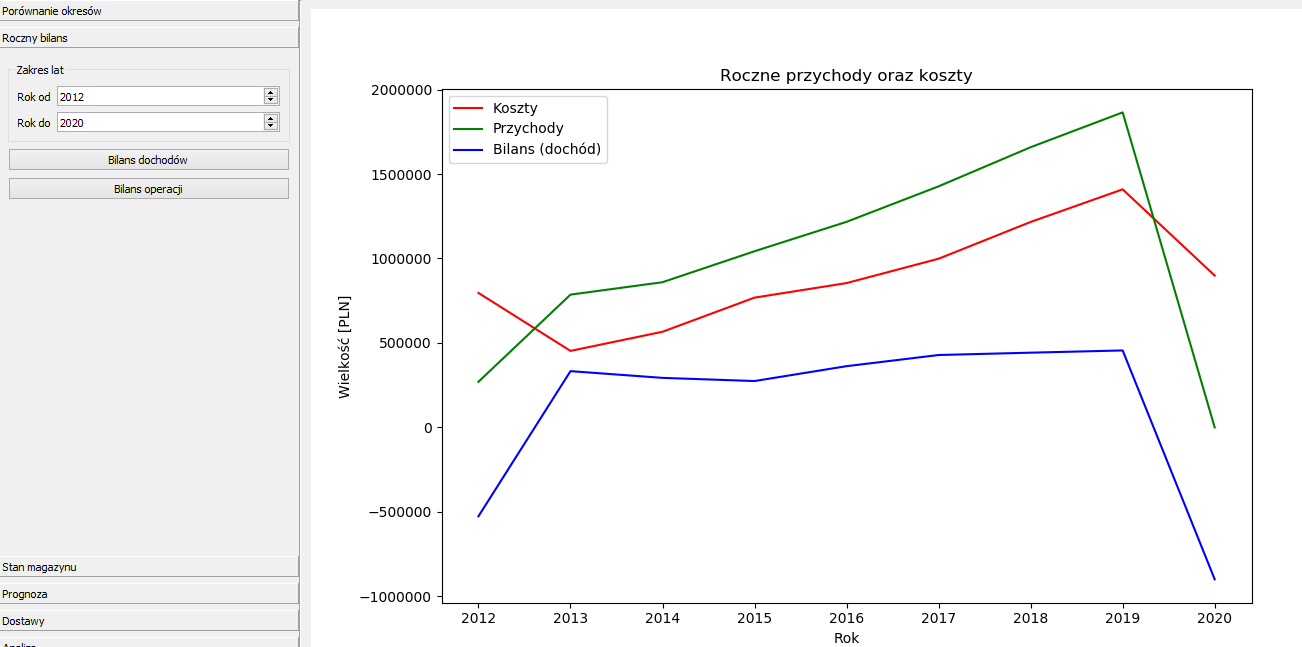


Dostępne funkcje systemu wspomagania decyzji prezentują się następująco:

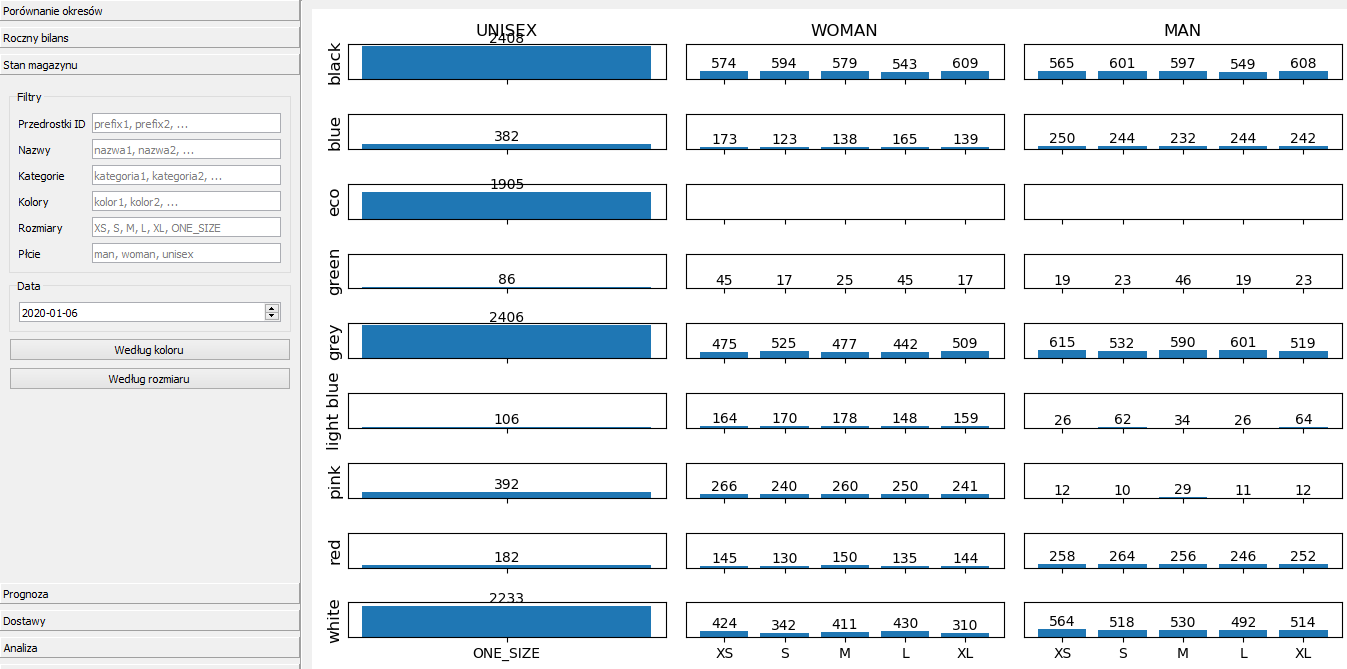
* *Porównanie okresów* – operacja ta pokazuje różnice między kilkoma  różnymi okresami pod względem różnych czynników (może to być zestawienie przychodów, kosztów, sprzedaży itp.). Użytkownik podaje pary okresów (rok-miesiąc-dzień), które mają być ze sobą porównane. Następnie wybiera jedną z możliwych opcji, jakie różnice go interesują. Istnieje możliwość filtrowania danych.



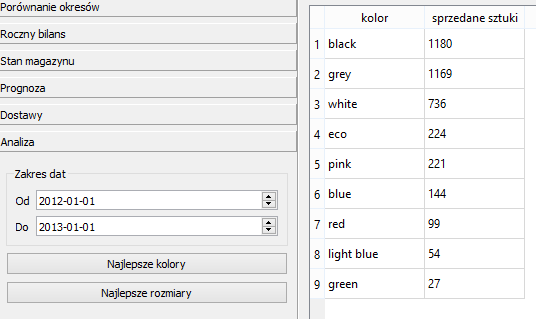
* *Roczny bilans* – funkcja obrazująca za pomocą wykresu, jak zmieniały się dochód sklepu lub dostawy i sprzedaż w czasie. Jedyna rzecz, którą użytkownik musi zrobić, to ustalić interesujący go zakres bilansu oraz wybrać, czego ma on dotyczyć.



* *Stan magazynu* – pokazuje, ile produktów dla poszczególnych płci znajdowało się w magazynie w wybranym dniu. Mogą one zostać podzielone wg kolorów lub rozmiarów. Istnieje możliwość filtrowania danych.



* *Prognoza* – na podstawie danych historycznych obliczana jest prognoza sprzedaży (ilość sprzedanych produktów) lub przychodów na kilka następnych miesięcy. Użytkownik wybiera zakres danych historycznych, cykl sezonowości oraz liczbę miesięcy dla prognozy. Potem tylko podejmuje decyzję, jakie informacje mają zostać wyświetlone. Istnieje możliwość filtrowania danych.
* *Dostawy* – ta funkcja odpowiada za wyświetlanie tabeli z informacją na jaki czas (według prognozy) posiadamy dany produkt w magazynie (rozpatruje y tylko do pół roku jeśli jest powyżej to wyświetla się powyżej 6) oraz ostrzeżenia o konieczności dokonania ewentualnych dostaw. Istniej możliwość filtrowania
* *Analiza* – pokazuje ranking najlepiej sprzedających się kolorów lub rozmiarów. Użytkownik jedynie musi wybrać zakres dat, względem którego zostanie dokonany taki ranking oraz wybrać, co go bardziej interesuje.



* *Inwentaryzacja* – ostatnia funkcja systemu, która porównuje stan magazynu zarejestrowany przez program z rzeczywistym stanem po inwentaryzacji.  Do programu wprowadzamy plik csv z stanem z inwentaryzacji.