

System wspomagania decyzji - dokumentacja

Zespół: Maciej Wiraszka, Mateusz Figurski, Piotr Walas - (wt 8-10)

1. Opis ogólnej struktury systemu:

Jedyną wykorzystaną do implementacji technologią jest język Python (wraz z bibliotekami: "Pandas" - do obsługi plików .xls z danymi oraz "Matplotlib" do tworzenia wykresów). Uznaliśmy, że jest on wystarczającym narzędziem do implementacji systemu wspomagania decyzji dla rozważanej symulacji. Daje on możliwość prostej i efektywnej obróbki oraz analizy danych wejściowych. Dlatego inne rozwiązania są zbędne. Program posiada dwa pełnoekranowe okna graficzne składające się z wykresów opartych na modelach/przebiegach. W trakcie dwóch pierwszych laboratoriów udało nam się zebrać dane pozwalające na zbudowanie modeli, na podstawie którego działa program. System dla przejrzystości wyświetla jedynie cztery najlepsze (wygrywające) wyniki oraz wartość średnią z wyników gier, przedstawioną jako kolejny przebieg. Przebieg aktualnej symulacji gry rynkowej wyświetlany jest na bieżąco, etap po etapie. Dla ułatwienia odczytywania danych w prawym dolnym rogu programu wyświetlają się współrzędne dla miejsca na wykresie, na które wskazuje kursor.

2. Opis procesu decyzyjnego:

W programie mamy możliwość obejrzenia decyzji podjętych w innych, owocnych (zakończonych dobrym wynikiem) grach, z wyszczególnieniem na poszczególne etapy. W przypadku każdej z tur gry, wprowadzamy aktualnie wypracowany wynik firmy, analizujemy wybrane fragmenty wykresów i uwzględniając aktualny wynik podejmujemy decyzję, co do poszczególnych parametrów wejściowych takich jak produkowany wolumen, cena, jakość, kwoty zainwestowane w marketing. Każdy z parametrów prezentowany jest na oddzielnym wykresie. Ze względu na duży niedeterminizm wynikający z obecności innych graczy oraz nieznajomości działania systemu, uznaliśmy, że system nie może jednoznacznie wyznaczyć parametrów dających wygraną w każdej grze, dlatego system takowych nie podaje (nie jest to system ekspercki). Użytkownik może jednak odczytać dane z wykresów i wzorując się na nich oraz porównując z poprzednimi przebiegami - modelami, może podjąć trafną decyzję.

3. Obserwacje:

Podczas przebiegu symulacji i dokładniejszej analizie wykresów najlepszych gier, zauważyliśmy, że najlepszym rozwiązaniem jest rozpoczęcie gry od wzięcia dużego kredytu. Pozwala to na produkcję większej ilości produktów, co przekłada się na udział w rynku, który okazał się w naszym mniemaniu ważnym czynnikiem i późniejszym wyznacznikiem sukcesu. Cena produktów przez cały okres gry nie powinna przekraczać 25 zł, co zapewnia wysoki odsetek sprzedanych produktów. Ustaliliśmy, że przez cały czas symulacji minimalny poziom jakości produktów będzie stały (około 60 jednostek), wynika to również z zaobserwowanych wartości (wykresy). Kwestią sporną jest inwestowanie w marketing. Nie dostrzegliśmy klarownej korelacji między wysokością pieniędzy zainwestowanych w marketing, a wysokością sprzedanego woluminu - było to spowodowane zbyt małą ilością rozegranych na

zajęciach rozgrywek. Ze względu jednak na fakt obecności innych graczy (niedeterminizmu), nie możemy stwierdzić, czy inwestowanie w marketing jest nieopłacalne. Sądzymy, że przy dużej ilości produkowanego woluminu, warto inwestować małe kwoty w reklamę na początku gry, a w późniejszych etapach rozgrywki zainwestować pewną sumę pieniędzy w marketing (szczególnie, jeśli firma dobrze prosperuje, i ma znaczną ilość wolnej gotówki).

4. Obsługa programu i przechowywanie danych:

Skrypt programu zapisany jest w pliku *data_visualization.py*, który należy uruchomić na komputerze z zainstalowanym środowiskiem Python (wraz z bibliotekami Pandas oraz Matplotlib). W tej samej lokalizacji powinien znajdować się folder *old_games* z danymi z poprzednich przebiegów. Pliki aktualnego przebiegu zapisujemy w konwencji *wyniki<numer zakończonej rundy>.xls* na pendrive w folderze *new_data* (na komputerach laboratoryjnych nie ma zainstalowanego środowiska Python, dlatego program uruchamiamy na własnym laptopie). Uruchamiamy program np. z wiersza poleceń (będąc w lokalizacji pliku .py) komendą *python data_visualization.py*. Okna z wynikami znajdują się w dwóch pełnoekranowych widokach, które można przełączać standardowymi komendami systemu (zazwyczaj *ALT+Tab*). Aby zakończyć program/załadować wyniki następnego etapu, przełączamy widok na wiersz poleceń, z którego uruchomiliśmy skrypt i naciskamy Enter.

5. Screeny z programu:

