Raport 1

Praca magisterska

Zastosowanie metod sztucznej inteligencji w prognozowaniu zbiorów kakaowca

# Przygotowanie środowiska pracy

Jako główne środowisko pracy z kodem wybrano **Visual Studio Code**. Jest to zaawansowany edytor kodu z wieloma wbudowanymi wtyczkami, w tym dla systemu kontroli wersji Git i jupyter notebook, co ułatwia pracę z kodem. **Git** to system kontroli wersji, który umożliwia zarządzanie kodem źródłowym i zmianami w kodzie w zdecentralizowany sposób, co ułatwia śledzenie historii zmian w projekcie. Jako zdalne repozytorium wybrano **GitHub**. Jest to platforma internetowa do hostowania kodu źródłowego oraz narzędzie do kontroli wersji Git, które znacznie ułatwia udostępnianie kodu oraz plików projektu. **Jupyter notebook** to narzędzie do interaktywnego pisania i uruchamiania kodu, które pozwala na łatwe udostępnianie notatników i wyników pracy. Jako narzędzie do zarządzania projektem oraz organizacją pracy wybrano aplikację **Jira**, która umożliwia m.in. śledzenie postępów oraz alokację zadań.

Jako język programowania wybrano **Python**, którego wszechstronność i prostota umożliwiają szybkie prototypowanie i rozwijanie zaawansowanych projektów w tym tych, dotyczących szeroko pojętego data science. Poniżej znajduje się lista głównych modułów użytych w pracy:

* **Pyodbc** – biblioteka umożliwiająca połączenie z bazą danych przy użyciu interfejsu ODBC (Open Database Connectivity). W projekcie umożliwia łączenie się z bazą danych na silniku SQL Server oraz wykonanie odpowiednich zapytań DQL.
* **SQLAlchemy** – umożliwia wygodne i elastyczne zarządzanie bazami danych poprzez użycie obiektowego interfejsu. Biblioteka ta, działa jako warstwa abstrackji pomiędzy językiem Python a różnymi bazami danych. W niniejszym projekcie SQLAlchemy wykorzystano w kolaboracji z Pandas do łatwego wykonywania zapytań na bazie danych SQL Server.
* **Pandas** – wykorzystana do transformacji i analizy danych, jej zaletami są łatwość w użyciu, elastyczność i szybkość.
* **DataPrep** - umożliwia szybkie i łatwe przetwarzanie danych do analizy oraz wstępną eksplorację danych (EDA - Exploratory Data Analysis). Biblioteka ta oferuje wiele funkcji, które ułatwiają przetwarzanie danych, w tym usuwanie duplikatów, usuwanie brakujących wartości, kodowanie zmiennych kategorycznych, normalizację wartości liczbowych i wiele innych. DataPrep pozwala również na łatwe generowanie wykresów i wizualizacji, co ułatwia zrozumienie charakterystyki danych.
* **Statsmodels** - umożliwia przeprowadzanie analizy statystycznej i ekonometrycznej danych. Biblioteka ta oferuje wiele funkcji do przeprowadzania analizy regresji, modelowania szeregów czasowych, testowania hipotez, analizy wariancji i wiele innych. Statsmodels pozwala na łatwe wykonywanie różnych procedur statystycznych, w tym estymację parametrów modelu, testowanie istotności parametrów czy wykrywanie wartości odstających.
* **Pmdarima** - umożliwia automatyczne dopasowanie modeli ARIMA (Autoregressive Integrated Moving Average) do szeregów czasowych. Biblioteka ta wykorzystuje algorytmy uczenia maszynowego, takie jak XGBoost czy Random Forests, do identyfikacji optymalnych wartości parametrów ARIMA. Pmdarima umożliwia automatyczne wykrywanie sezonowości, różnicowanie i transformację szeregów czasowych, co umożliwia dopasowanie modelu ARIMA. Biblioteka ta oferuje również wiele funkcji, które ułatwiają przewidywanie przyszłych wartości szeregów czasowych, takie jak predykcja z przedziałami ufności czy diagnozowanie modeli.
* **Scikit-learn** - umożliwia wykonywanie różnych zadań związanych z uczeniem maszynowym (machine learning), takich jak klasyfikacja, regresja, grupowanie, redukcja wymiarowości i wiele innych. Biblioteka ta oferuje wiele funkcji, które ułatwiają budowanie, trenowanie i ocenę modeli, takich jak selekcja cech, walidacja krzyżowa, optymalizacja hiperparametrów i wiele innych.
* **Lazypredict** – pozwala na szybkie i łatwe budowanie wielu modeli uczenia maszynowego bez konieczności doboru odpowiednich hiperparametrów czy optymalizacji modelu. Biblioteka ta oferuje wiele popularnych algorytmów uczenia maszynowego, takich jak drzewa decyzyjne, regresja liniowa, lasy losowe i wiele innych. Lazypredict generuje raport z wynikami modeli, w tym ich skuteczności, czasu trenowania i predykcji oraz wykresów porównawczych. Biblioteka ta jest często wykorzystywana w celu szybkiej analizy danych i wyboru najlepszego modelu, a także jako narzędzie do testowania i wstępnego dopasowania modelów przed dalszą optymalizacją.

W celu izolacji projektu, uniknięcia konfliktów zależności, powtarzalności wyników oraz łatwego zarządzania, wykorzystano wirtualne środowisko pythona w wersji 3.9.x. Do stworzenia wirtualnego środowiska wykorzystano otwartoźródłowy system zarządzania pakietami i środowiskami Conda, który odznacza się prostym interfejsem wiersza poleceń oraz dobrze zbudowaną, oficjalną dokumentacją. Pondadto instrukcję tworzenia wirtualnego środowiska można zapisać w pliku .yml, który pełni rolę backupu mogącego być łatwo wczytanym na innych urządzeniach.

# Przygotowanie danych

## Wczytanie danych z serwera SQL Server

Zarówno dane dotyczące zbiorów kakaowca jak i dane pogodowe umieszczone są w bazie danych na silniku SQL Server. W celu ich uzyskania wykorzystano własną klasę DatabaseManager w paczce Databases, która implementuje podstawowe funkcjonalności takie jak nawiązywanie połączenia z wybranym serwerem SQL Server w konstruktorze klasy oraz pobieranie danych z serwera poprzez metodę get\_data.

Dane zostały pobrane z bazy danych bezpośrednio do formy ramki danych (DataFrame) oferowanej przez bibliotekę pandas. Następnie surowe dane zostały zapisane w formacie .csv, co pozwoliło na znaczne skrócenie czasu ładowania danych w późniejszych etapach analizy.

## Eksploracyjna analiza danych zbiorów kakaowca

## Wstępne przetwarzanie oraz eksploracyjna analiza danych pogodowych NOAA