Programowanie w Pythonie. Projekty

Poniżej znajduje się lista proponowanych tematów projektów. Jeżeli macie własne pomysły, czy to od zera, czy uszczególniające/ukierunkowujące/modyfikujące jeden z tematów poniżej, gorąco zachęcam do ich realizacji.

Tematy w zamyśle są dosyć ogólne. Wszystkie wyszczególnione elementy powinny znaleźć się w projekcie (np. koszyk, wizualizacja danych, etc., w zależności od tematu), jednak **konkretna realizacja zależy od Was**. Czyli np. wybieracie stworzenie gry tekstowej, a to, czy jej akcja będzie się toczyć w średniowiecznych zamkach czy na planecie Xander, czy do projektu podejdziecie z paradygmatu obiektowego czy funkcyjnego, pakiety i funkcje, z których korzystacie (o ile nie są jasno określone – lecz zawsze można sięgnąć po dodatkowe) – zależy od Was :) Jeżeli będzie to problematyczne, to proszę zwrócić się do mnie (laura.grzonka@ug.edu.pl lub laura.grzonka@phdstud.ug.edu.pl), trochę Państwa naprowadzę ;)

W związku z tym, jeżeli parę osób zainteresuje się tym samym tematem, wszystkie z nich mogą go podjąć, **pod warunkiem, że każdy_a zrealizuje go inaczej**. Przykładowo można sięgnąć po inny zbiór danych, inną metodę klasyfikacji lub stworzyć aplikację mobilną, która będzie robiła coś innego niż apka koleżanki czy kolegi. O wyborze projektu proszę mnie poinformować na Teamsach lub mailowo.

Sposób realizacji i terminy

1. Wybranie i zgłoszenie tematu.
2. Założenie repozytorium na GitHubie i dodanie mnie do niego.
3. Research w Internecie (bibliografia fajnych stron, metody, materiały).
4. Proste testy kodu z tutoriali z Internetu.
5. Poważniejsze własne eksperymenty.
6. Gotowe projekty proszę przekazać mi **do 22.05**.
7. Sprawdzę je w ciągu 2-3 tygodni i odeślę do Państwa z informacją zwrotną.
8. Następnie będą mieli Państwo czas na poprawki (prawdopodobnie tydzień).
9. Ostateczną ocenę projektu przekażę pod koniec semestru :)

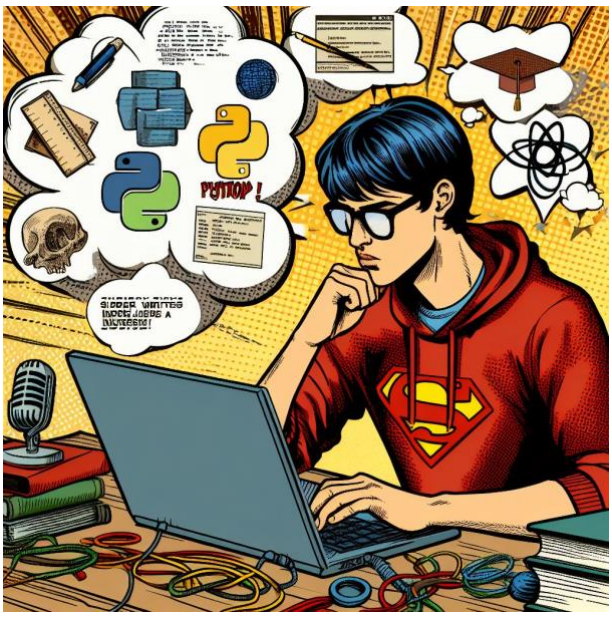
Projekt można zrealizować na dwa główne sposoby:

1. **Raport badawczy**

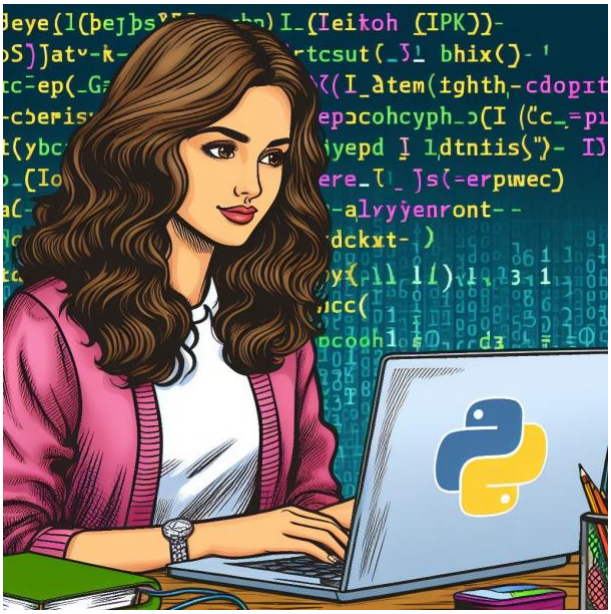
Dla wybranego problemu/bazy danych – sprawdzam jakie modele/techniki działają próbując osiągnąć jak najlepsze wyniki, np. klasyfikacji. Robię porównanie wydajności w formie sprawozdania.

2. **Aplikacja/serwis**

Stosuję wybrany, najlepszy model klasyfikujący w aplikacji/serwisie do praktycznych zastosowań. Prezentuję aplikację z jakąś prostą dokumentacją.



Jeśli ktoś chciałby zrealizować projekt w inny sposób, proszę się ze mną skontaktować w celu omówienia szczegółów.



Co składa się na projekt?

1. Aplikacja/serwis/raport badawczy.
    a. W tym bibliografia. źródła internetowe również proszę uwzględnić, dopisując datę dostępu.
2. Kod dostępny na repozytorium.
3. **Krótkie,** lecz treściwe sprawozdanie/raport (*clear, consise, & consistent*).
4. Prezentacja na zajęciach. Czas będzie ograniczony, najprawdopodobniej do 10 minut wraz z pytaniami/dyskusją. Forma prezentacji (slajdy, raport, pokazanie samego kodu i jego działania) – dowolna.

Ocena projektu

Niestety w przypadku projektów o tak różnorodnej tematyce ciężko wyznaczyć jednoznaczną miarę oceniania. Ogólnie będę brała pod uwagę następujące kryteria:

- Czy Wasz wkład (czas, energia, własny kod programistyczny) w projekt był duży czy mały? Tutaj warto, żeby student_ka wskazał co jest zrealizowane samodzielnie, co skopiowane z samouczka, a co wygenerowane przez AI. Należy też uwzględnić wszystkie źródła, z których korzystaliście.
- Czy projekt jest oryginalny/nowatorski, czy projekt jest raczej dobrze zbadany/odtwórczy?
- Jak ambitny jest projekt? Czy np. zawiera tylko podstawowe funkcjonalności wybrane strony internetowej, czy również bardziej zaawansowane?
- Czy projekt jest dobrze zrealizowany i zawiera wszystkie istotne komponenty, które będą się różnić w zależności od tematu; przykładowo preprocessing, algorytmy, modele, funkcjonalności?
- Czy projekt sięga po stare oklepane schematy, czy raczej student_ka starał się korzystać z najnowocześniejszych technik, algorytmów udostępnianych w artykułach naukowych, blogach naukowych itp.?
- Jak projekt prezentuje się pod względem estetycznym? Czy student_ka zadbał_a o odpowiedni font, wielkość liter, dopasowanie do szerokości ekranu, kontrast, kolorystykę – w tym taką, by była odpowiednia dla osób z daltonizmem?
- Czy student_ka był w stanie dobrze zaprezentować projekt?

Trudno jest stworzyć jednolity system kryteriów oceniania dla tak różnorodnych projektów. Generalnie:

- Projekt jest na 25 pkt.
- Bardzo istotne jest Państwa zaangażowanie w projekt, dbałość o szczegóły, dopracowanie poszczególnych etapów.
- Punktowana jest oryginalność.

- Prezentacja projektu też będzie oceniana. Czas prezentacji będzie określony, prawdopodobnie 10 min **wraz z pytaniami/dyskusją** i nie będzie można go przekroczyć (choćby z tego banalnego powodu, że inaczej nie zdążymy w czasie zajęć).
- Zachęcam do oddania projektu przed czasem -- takie prace będę oceniała nieco łagodniej :)
- **Ważne:** jeśli coś Wam nie wychodzi mimo licznych/pogłębionych prób naprawienia sytuacji, opiszcie to w sprawozdaniu/raporcie i przedstawcie w prezentacji. **Skomentujcie, dlaczego Waszym zdaniem nie działa to tak, jak powinno**. Wówczas ocenię to znacznie łagodniej, a może i nawet na plus, gdyż oznacza to, że zdajecie sobie sprawę z problemu i próbowaliście go rozwiązać, a cóż, nie zawsze jest to takie łatwe :)

Jeżeli chodzi o punkty, to mniej więcej będzie wyglądało to tak:

- 13-17% punktów (52-68% = trójki i trójki plus) to projekty z dużymi niedoróbkami lub bardzo odtwórcze i niezbyt ciekawe.

- 18-19 punktów (72-76% = czwórka) i 20-22 punktów (80-88% = czwórka plus): są niedoróbki, jakieś eksperymenty są niedokończone, może zabrakło porównań między modelami, sprawozdanie jest niedopracowane (np. brak komentarzy, wykresów, bibliografii -- źródła internetowe **też** uwzględniamy! Wraz z datą dostępu do strony).

- 23-24 punktów (92-96% = piątka): wszystko jest dobrze, poza drobnymi niedoróbkami, np. brak drobnej funkcjonalności (np. koszyk działa *prawie* zawsze), jakiegoś klasyfikatora, brak ciekawego wykresu (np. krzywej uczenia się i jej analizy), brak komentarzy, ale ogólnie projekt jest bardzo dobrze zrealizowany.

- 25 punktów (100% = piątka): wszystko jest ładnie zrobione, nie ma się do czego przeczepić, prezentacja jest czytelna, student_ka sprawdził_a sporo technologii poznanych na zajęciach i może nawet wykazał_a się inicjatywą i znalazł_a inne ciekawe technologie.

Ogólne elementy brane pod uwagę w ocenie projektu

Wszystkie tematy

Sprawozdanie

Eksperymenty opisz w przejrzystym sprawozdaniu podzielonym na rozdziały. Można je zrobić w formie docx/pdf lub notatnika jupyterowego, albo nawet prezentacji powerpoint czy pliku latexowego. Sprawozdaniu powinny znajdować się Twoje objaśnienia, komentarze, wstawki kodu (najbardziej istotne), wykresy i grafiki, tabelki z wynikami. Dobrze będzie, jeśli sprawozdanie rozpocznie się krótkim wstępem i zakończy konkluzjami podsumowującymi eksperymenty. Dołącz bibliografię z źródłami.

W przypadku aplikacji webowych/mobilnych, możesz dołączyć instrukcję dla użytkownika (readme).

Prezentacja

Przedstawienie wszystkich najważniejszych etapów tworzenia programu/poszczególnych funkcjonalności. Wskazanie ewentualnych błędów/niedoróbek wraz z wyjaśnieniem. Czytelność samej prezentacji, sposób prezentowania: wyraźny, skierowany w stronę widowni, bez czytania :)

Dokumentacja

Kod źródłowy aplikacji musi być dobrze udokumentowany, np. komentarze do kodu, instrukcja użytkowania, opis funkcjonalności.

Stabilność i wydajność

Aplikacja musi działać stabilnie i raczej bez błędów.

Aplikacje webowe

Funkcjonalność. Aplikacja musi poprawnie realizować swoje założone funkcje i spełniać oczekiwania użytkownika, np. możliwość logowania i rejestracji użytkowników, dodawania i edycji treści, wyszukiwania produktów.

Wygląd i użyteczność. Aplikacja musi mieć estetyczny i przejrzysty interfejs użytkownika, który jest łatwy w nawigacji i intuicyjny w obsłudze, np. spójny styl graficzny, czytelne teksty, logiczna struktura strony.

Aplikacje użytkowe

Funkcjonalność. Aplikacja musi poprawnie realizować swoje założone funkcje i spełniać oczekiwania użytkownika, np. możliwość zarządzania czasem, śledzenia wydatków, zapisywanie notatek/wydarzeń.

Wygląd i użyteczność. Aplikacja musi mieć estetyczny i przejrzysty interfejs użytkownika, który jest łatwy w nawigacji i intuicyjny w obsłudze, np. spójny styl graficzny, czytelne teksty, logiczna struktura aplikacji.

Gry

Gra ma zapewniać użytkownikowi przyjemne doznania :) Super będą zróżnicowane poziomy trudności, własne animacje. Nie musi być superzaawansowana, lecz powinna działać sprawnie i bez lagów. Ciekawy scenariusz i efekty dźwiękowe będą mile widziane.

Uczenie maszynowe

Baza danych

Czy baza danych jest ciekawa/oryginalna/słabo zbadana? Czy samodzielnie ją stworzę, czy jest ściągnięta? Większość z Państwa pewnie ściągnie gotowca. Warto wybrać bazy danych duże (>10 000 próbek), nietrywialne (dużo kolumn, niełatwe obrazki), może też z błędami wymagającymi naprawy.

Preprocessing

Bazę danych należy naprawić (usuwanie brakujących danych, błędów, wartości odstających). Należy sprawdzić balans klas i ewentualnie zbalansować dane (imputacja, downsampling, upsampling, augemntacja obrazów).

Klasyfikacja

Należy porównać kilka algorytmów klasyfikujących, w tym sieci neuronowe. Warto je przebadać różnymi miarami (accuracy, confusion matrix, learning curve, itp.). Eksperymenty należy powtórzyć wielokrotnie i uśrednić wynik. Można stosować cross-validation.

Aplikacja/serwis w przypadku uczenia maszynowego

Baza danych

Trochę jak po lewej, ale wymagania są nieco mniejsze. Jeśli aplikacja jest prototypem, to właściwie baza danych może być ręcznie stworzona i nie musi być duża. Warto jednak skolekcjonować przynajmniej 100 próbek.

Preprocessing

Warto rozpatrzyć kroki wymienione w sekcji raport badawczy.

Klasyfikacja & Optymalizacja

Właściwie wypada skupić się na klasyfikatorach, które najlepiej pasują do naszego zadania. Taki klasyfikator trzeba dobrze skonfigurować. powinien być nie tylko precyzyjny, ale również lekki. Czas obliczeń jest bardzo ważny, zwłaszcza na słabszych maszynach/aplikacjach mobilnych.

Dokumentacja/Sprawozdanie

Opisz wszystkie funkcjonalności aplikacji.
Opisz eksperymenty tak jak powiedziano po lewej, ale może nie tak dokładnie jak w przypadku raportu.
Możesz napisać instrukcję dla użytkownika aplikacji (readme).

Tematy

Aplikacje webowe

1. Klon popularnej strony. Wybierz popularną stronę internetową i stwórz jej uproszczoną wersję. Przykład. portal aukcyjny, serwis społecznościowy, blog.
2. Aplikacja webowa z funkcją logowania i rejestracji użytkowników.
3. System zarządzania treścią (CMS). Prosta platforma do tworzenia i edycji stron internetowych.
4. Sklep internetowy. Aplikacja do sprzedaży produktów online z koszykiem i płatnościami.

Aplikacje użytkowe

5. Kalendarz. Aplikacja do zarządzania czasem z funkcją dodawania wydarzeń, notatek, przypomnień i synchronizacji z kalendarzem Google.
6. Lista rzeczy do zrobienia. Aplikacja do zarządzania zadaniami z priorytetami, kategoriami i powiadomieniami.
7. Kalkulator finansowy. Aplikacja do obliczania budżetu, śledzenia wydatków i oszczędzania pieniędzy.

8. Czytnik RSS. Aplikacja do pobierania i wyświetlania wiadomości z różnych źródeł.

Gry

9. Gra tekstowa. Klasyczna gra przygodowa oparta na tekście, z wyborem opcji i interakcją z otoczeniem.
10. Gra graficzna z interfejsem użytkownika.
11. Gra z wykorzystaniem biblioteki PyGame. Bardziej zaawansowana gra z grafiką i dźwiękiem.

Uczenie maszynowe

12. Klasyfikacja obrazów. Program do rozpoznawania obiektów na zdjęciach.
13. Predykcja pogody. Program do przewidywania pogody na podstawie danych historycznych.
14. Detektor emocji. Program do analizowania opinii i nastrojów w tekstach.
15. Czatbot. Prosty chatbot oparty na sztucznej inteligencji.
16. Aplikacja do grania ruchem. Program umożliwiający granie w grę nie za pomocą myszki, pada czy klawiatury, lecz wyginając śmiało ciało ;)
17. Klasyfikacja audio, np. po przekształceniu dźwięku w obraz (zob. np. https://towardsdatascience.com/getting-to-know-the-mel-spectrogram-31bca3e2d9d0). Polecam też https://towardsdatascience.com/audio-classification-with-deep-learning-in-python-cf752b22ba07

Inne

18. Program do analizy danych. Program do wczytywania, czyszczenia i analizowania danych z różnych źródeł.
19. Aplikacja mobilna. Prosta aplikacja mobilna z wykorzystaniem biblioteki Kivy.
20. Interfejs do sterowania robotem. Program do sterowania robotem za pomocą interfejsu graficznego.

Przydatne linki

Aplikacje webowe

Ogólne

https://fliplet.com/blog/how-to-create-a-web-app-in-15-steps/
https://medium.com/@dattu1993/creating-a-web-application-with-python-a-comprehensive-guide-for-beginners-db59df5867e4
https://www.browserstack.com/guide/web-development-in-python-guide

Flask

https://flask.palletsprojects.com/en/3.0.x/
https://learn.microsoft.com/en-us/visualstudio/ide/quickstart-python?view=vs-2022
https://www.freecodecamp.org/news/how-to-build-a-web-application-using-flask-and-deploy-it-to-the-cloud-3551c985e492/
https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3

Django
https://docs.djangoproject.com/en/5.0/intro/tutorial01/
https://www.geeksforgeeks.org/django-tutorial/
https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial_local_library_website
https://towardsdatascience.com/build-a-user-authentication-web-app-with-python-and-django-c60000148724

Przykłady świetnych stron internetowych
https://docs.google.com/document/u/0/
https://www.netflix.com
https://trello.com
https://www.office.com
https://allegro.pl
https://www.ebay.pl
https://open.spotify.com/
https://www.facebook.com
https://basecamp.com
https://www.pinterest.com
https://twitter.com

Gry
https://towardsdatascience.com/making-simple-games-in-python-f35f3ae6f31a
https://inventwithpython.com/pygame/
https://www.geeksforgeeks.org/snake-game-in-python-using-pygame-module/
https://opensource.com/article/20/10/learn-python-ebook

Aplikacje użytkowe
https://medium.com/nyu-ds-review/how-to-create-a-python-web-app-to-securely-collect-and-store-user-information-cb8f36921988
https://www.makeuseof.com/python-database-registration-app-how-create/
https://www.geeksforgeeks.org/create-first-gui-application-using-python-tkinter/

Uczenie maszynowe
Przykładowe bazy danych
https://www.kaggle.com/datasets/uciml/adult-census-income Rozpoznawanie, ile zarabia osoba
https://www.kaggle.com/datasets/blastchar/telco-customer-churn Klasyfikacja klientów telefonii komórkowej: czy przedłużą umowę, czy zrezygnują?
https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009 Jaką jakość ma czerwone wino? (można zmienić liczbę klas z 1-10 na good/bad).
https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset Diagnoza choroby serca
https://www.kaggle.com/datasets/emmanuelfwerr/thyroid-disease-data Diagnoza choroby tarczycy

https://www.kaggle.com/datasets/uciml/mushroom-classification Klasyfikacja grzybów jadalnych i niejadalnych

Obrazkowe bazy danych
https://www.kaggle.com/datasets/gpiosenka/100-bird-species Klasyfikacja gatunków ptaków
https://www.kaggle.com/datasets/phucthaiv02/butterfly-image-classification Klasyfikacja gatunków motyli
https://www.kaggle.com/datasets/alexattia/the-simpsons-characters-dataset
https://www.kaggle.com/datasets/kostastokis/simpsons-faces Klasyfikacja postaci z Simpsonów
https://www.kaggle.com/datasets/andyczhao/covidx-cxr2 Diagnoza COVID na podstawie zdjęcia RTG płuc
https://www.kaggle.com/datasets/csafrit2/plant-leaves-for-image-classification Klasyfikacja roślin po obrazie liścia
https://www.kaggle.com/search?q=image+classification+in%3Adatasets wiele innych przykładów – polecam przejrzeć.

Bazę danych można stworzyć też samemu_ej. Przykładowo mogą to być Wasze zdjęcia, które chcecie oznaczyć tagiem *pies*/*kot*, jeżeli na zdjęciu znajduje się Wasz zwierzak.
Można też skorzystać z innej, znalezionej przez siebie, bazy danych.

Granie ruchem

https://www.youtube.com/watch?v=Vi3Li3TkUVY&ab_channel=EverythingIsHacked
Chodzi o napisanie apki, a nawet jej prototypu, gdyż wystarczy zgrubnie napisany program umożliwiający sterowanie ruchem naszego ciała/rąk/dłońmi obiektu w grze. Gra może być napisana ręcznie lub ściągnięta. Dla uproszczenia można założyć, że np. grająca osoba stoi w odpowiedniej odległości. Jeśli nie uda się wytrenować naszego modelu, można zastosować transfer learning. Generalnie dużo ciekawych możliwości :)

https://huggingface.co/datasets/sayakpaul/poses-controlnet-dataset
https://huggingface.co/spaces/hysts/mediapipe-pose-estimation
https://developers.google.com/mediapipe + https://github.com/google/mediapipe

Licznik
Można napisać aplikację, która będzie zliczała obiekty na zdjęciu. Przykładowo:

- Jabłek na drzewie, truskawek na krzaku https://www.youtube.com/watch?v=5pSemkmVMRc&ab_channel=VijayKumar
- Ludzi na sali wykładowej bądź na ulicy https://www.youtube.com/watch?v=Po8ozbsNvV4&ab_channel=RugvedChavan

Detektor emocji
Aplikacja, która będzie rozpoznawała emocje osoby na zdjęciu. Następnie aplikacja może wykonywać jakąś akcję – różną, zależnie od emocji:

- Puścić relaksującą muzykę.
- Zaproponować gimnastykę, kawę? :P
- Puścić głośny sygnał, żeby ocucić przysypiającą osobę.

https://www.youtube.com/watch?v=UHdrxHPRBng&ab_channel=DataMagic%28bySunnyKusawa%29

Ogólne/inne

https://trio.dev/python-applications/
https://www.netguru.com/blog/image-recognition-apps
https://www.interviewquery.com/p/image-recognition-machine-learning-projects
https://neptune.ai/blog/15-computer-visions-projects

Estetyka i kolory

Proszę zadbać o to, by paleta kolorów była odpowiednia dla osób z niewidzeniem barw. Przykłady możecie znaleźć np. tutaj:
https://www.simplifiedsciencepublishing.com/resources/best-color-palettes-for-scientific-figures-and-data-visualizations
https://www.moleculareco logist.com/2020/04/23/simple-tools-for-mastering-color-in-scientific-figures/
https://www.animateyour.science/post/how-to-select-a-great-colour-scheme-for-your-scientific-poster
https://www.simplifiedsciencepublishing.com/resources/best-color-palettes-for-scientific-figures-and-data-visualizations
Proszę również zadbać o odpowiedni kontrast, wielkość liter itd. Opisy wykresów i grafik powinien znajdować się *pod nimi*, natomiast tabel, jeśli są – *powyżej*.