

Wprowadzenie do Uczenia Maszynowego (ML)

Speaker: Ulvi Movsum-zada
Senior Software Engineer
IBM



Co to jest sztuczna inteligencja (AI)?

Sztuczna inteligencja (AI) to dział informatyki, który polega na tworzeniu programów komputerowych zdolnych do wykonywania zadań, które zwykle wymagają ludzkiej inteligencji, takich jak rozpoznawanie mowy, podejmowanie decyzji, tłumaczenie języków i wiele innych.

Przykłady AI w codziennym życiu:

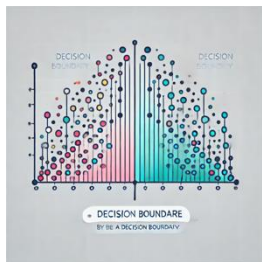
- **Siri i Google Assistant:** rozpoznawanie głosu i odpowiedzi na zapytania użytkownika.
- **Rekomendacje filmów na Netflixie, YouTube:** systemy rekomendacji, które uczą się twoich preferencji.
- **Samochody autonomiczne:** samochody, które uczą się nawigować bez kierowcy.
- ...



Problemy do rozwiązania

Klasyfikacja

Zadaniem klasyfikacji jest przypisanie danych wejściowych do jednej z kilku kategorii



Regresja (Regression)

Regresja polega na przewidywaniu wartości liczbowych na podstawie danych wejściowych.



Analiza szeregów czasowych

Analiza szeregów czasowych zajmuje się danymi zależnymi od czasu, aby przewidywać przyszłe wartości na podstawie przeszłych danych.



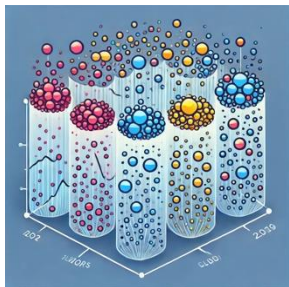
Przetwarzanie języka naturalnego (NLP)

AI rozumie i analizuje ludzką mowę oraz tekst, co umożliwia interakcje między człowiekiem a maszyną.



Klasteryzacja (Clustering)

Klasteryzacja polega na grupowaniu danych w zbiory (klastry), gdzie elementy w tym samym klastrze są do siebie bardziej podobne niż elementy z innych klastrów.



Rozpoznawanie obrazów i wideo

AI potrafi analizować obrazy i filmy, identyfikując w nich obiekty, twarze, ruchy itp.



Czym jest uczenie maszynowe (ML)?

Uczenie maszynowe (ang. Machine Learning, ML) to poddziedzina sztucznej inteligencji, która koncentruje się na tworzeniu algorytmów i modeli, które mogą uczyć się na podstawie danych. Zamiast programować komputer krok po kroku, w ML algorytmy odkrywają wzorce w danych i uczą się wykonywać zadania na podstawie tych wzorców.

Rodzaje uczenia maszynowego:

1. Uczenie nadzorowane (*Supervised Learning*):

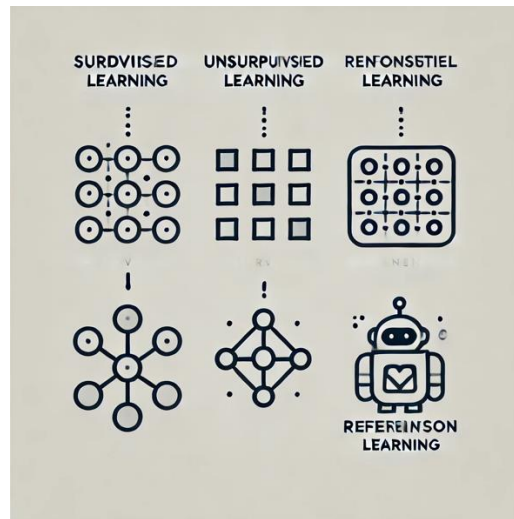
- Model uczy się na podstawie danych z etykietami, czyli przykładów, gdzie wynik (odpowiedź) jest znany.

2. Uczenie nienadzorowane (*Unsupervised Learning*):

- Model uczy się na podstawie danych bez znanych wyników. Algorytm próbuje odkryć ukryte struktury w danych.

3. Uczenie ze wzmocnieniem (*Reinforcement Learning*):

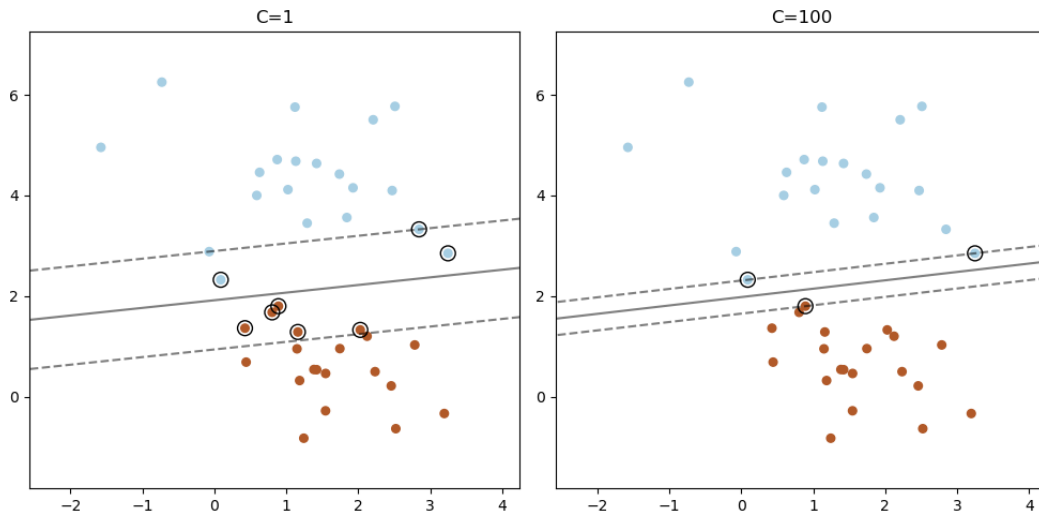
- Model uczy się poprzez interakcję z otoczeniem i otrzymywanie nagród lub kar w zależności od tego, jak dobrze radzi sobie z zadaniem.



Przykład algorytmu klasyfikacji – Support Vector Classifier (SVC)

SVC to algorytm, który pomaga komputerom podejmować decyzje o tym, do której grupy (klasy) coś należy.

Zadaniem SVC jest narysowanie **linii** (w rzeczywistości może to być również bardziej skomplikowana powierzchnia), która najlepiej oddziela te dwie grupy, tak aby można było przewidzieć, do której grupy należą nowe dane.



Jak działa SVC?

1. Hiperpłaszczyzna:

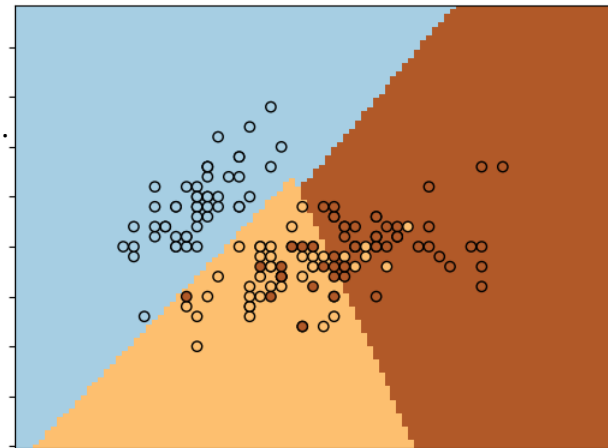
- SVC stara się znaleźć hiperpłaszczyznę, która najlepiej oddzieli różne klasy w danych. W dwuwymiarowych danych będzie to linia, w trójwymiarowych — płaszczyzna, a w przypadku danych o wyższych wymiarach — hiperpowierzchnia.
- Najlepsza hiperpłaszczyzna to ta, która maksymalizuje odległość (margines) między najbliższymi punktami danych z każdej klasy.

2. Support Vectors (Wektory nośne):

- Są to punkty danych, które znajdują się najbliżej hiperpłaszczyzny. To one decydują o tym, gdzie zostanie umieszczona linia podziału. Pozostałe dane nie wpływają na pozycję tej linii.

3. Margines:

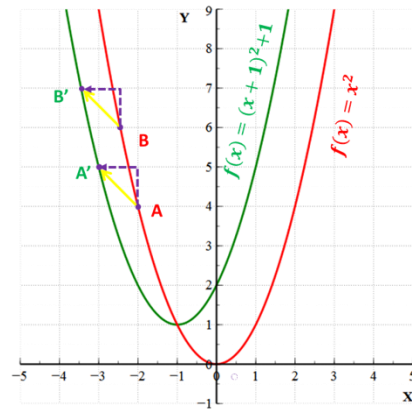
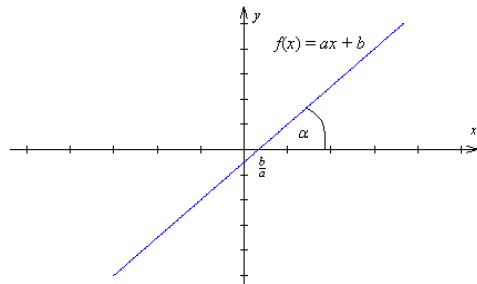
- Algorytm SVC dąży do maksymalizacji marginesu, czyli odległości pomiędzy support vectors a hiperpłaszczyzną. Im większy margines, tym lepsza klasyfikacja, ponieważ model będzie bardziej odporny na błędy i szumy w danych.



Rodzaje jąder w SVC:

SVC może korzystać z różnych jąder (kernel), które przekształcają dane do wyższej przestrzeni wymiarowej, aby lepiej oddzielić klasy, zwłaszcza w przypadkach, gdy klasy nie są liniowo separowalne (czyli nie można ich oddzielić prostą linią)

- **Liniowe jądro:** Używane, gdy dane są liniowo separowalne (można je oddzielić prostą lub płaszczyzną).
- **Wielomianowe (polynomial kernel):** Pozwala oddzielać dane, które mają bardziej złożoną strukturę.
- **Jądro RBF (Radial Basis Function):** Bardzo popularne, pozwala na oddzielenie bardziej złożonych nieliniowych danych.



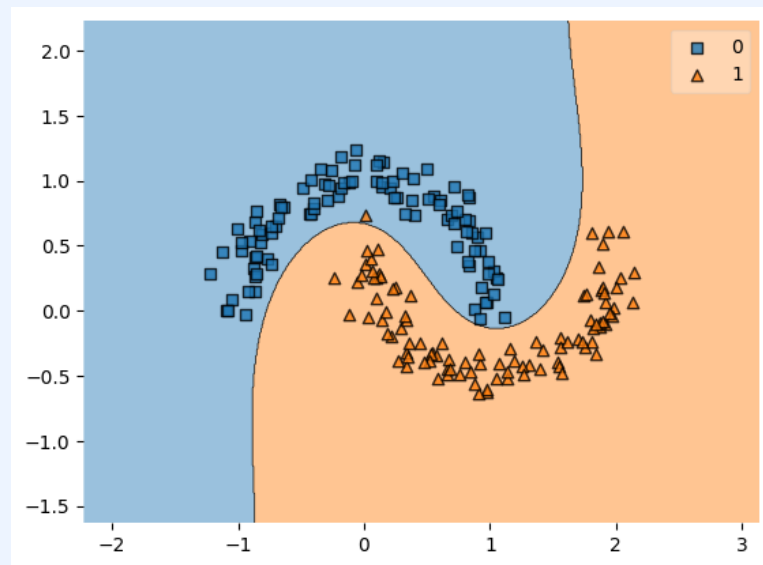
Praktyka – Implementacja SVC w Pythonie

GitHub: https://github.com/tstopa/P-TECH_QC.git

Notebook 1: classic_linear_SVC.ipynb

wspólna praca

Notebook 2: classic_polinomina_SVM_zadanie.ipynb **praca indywidualna**



QUIZ

Podstawowe pojęcia kwantowych obliczeń

1. Kubit (qubit):

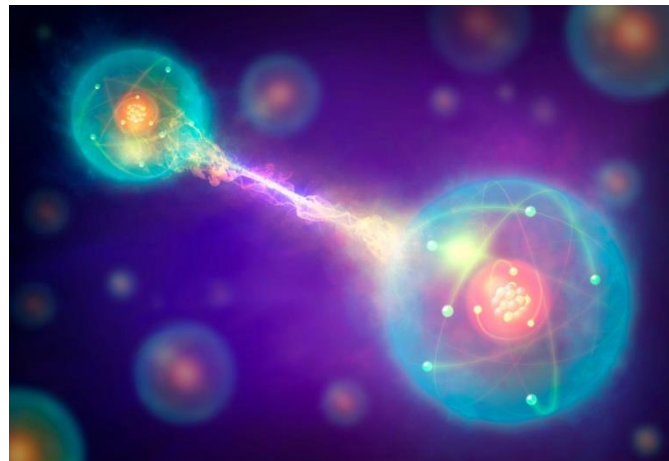
- Podstawowa jednostka informacji w komputerze kwantowym. W odróżnieniu od klasycznego bitu, który może być 0 lub 1, kubit może istnieć w stanie superpozycji, gdzie jest jednocześnie 0 i 1 z pewnym prawdopodobieństwem.

2. Superpozycja:

- Dzięki superpozycji, komputer kwantowy może jednocześnie wykonywać wiele operacji na różnych stanach, co pozwala na dużo szybsze przetwarzanie informacji.

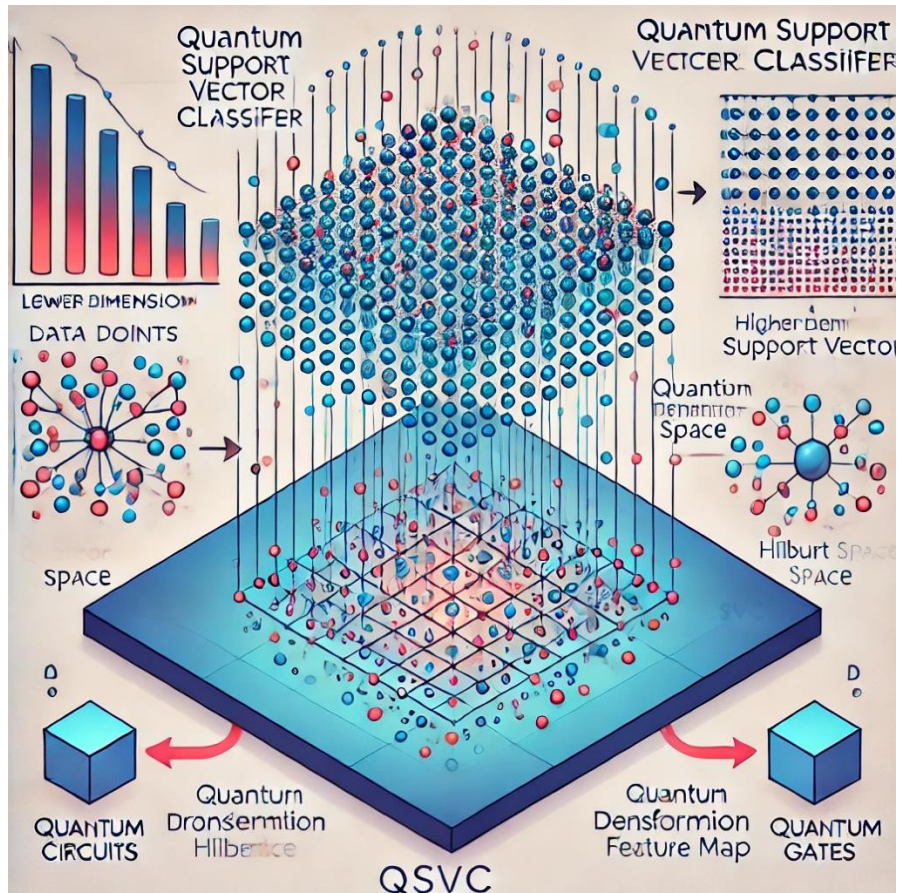
3. Splątanie (entanglement):

- Splątanie to kwantowe zjawisko, w którym dwa kubity są ze sobą związane, nawet jeśli są fizycznie oddalone. Stan jednego kubita automatycznie wpływa na stan drugiego. Dzięki splątaniu kwantowe komputery mogą przeprowadzać bardziej złożone obliczenia.



QSVC (Quantum Support Vector Classifier)

QSV jest kwantową wersją klasycznego **SV**, która wykorzystuje kwantowe obliczenia do tworzenia bardziej złożonych funkcji jądra (kernel), co może poprawić dokładność klasyfikacji dla trudnych, nieliniowych problemów.



Jak działa QSVC?

1.: Przekształcenie danych do kwantowego stanu

1. W klasycznym SVC dane są analizowane w zwykłym, niskowymiarowym przestrzeni, co czasem nie pozwala na łatwe oddzielenie klas. W **QSVC** dane są najpierw przekształcane w **kwantowe stany** za pomocą specjalnych operacji kwantowych, które rozszerzają nasze dane do **wysokowymiarowej przestrzeni Hilberta**.

2.: Funkcja jądra kwantowego (Quantum Kernel)

1. W przestrzeni Hilberta QSVC stosuje **funkcję jądra kwantowego** (quantum kernel), która przekształca dane, uwzględniając ich złożone wzajemne zależności. Dzięki kwantowej naturze algorytmu, te relacje mogą być uchwycone bardziej precyzyjnie niż w klasycznych algorytmach.
2. W klasycznym SVC stosuje się funkcje jądra (kernel functions), które pomagają w przekształceniu danych. W QSVC to przekształcenie jest wykonywane przez **kwantowe bramki**, które działają w przestrzeni Hilberta, co pozwala na bardziej złożone i dokładne separowanie klas.

3.: Wyznaczanie granicy decyzyjnej

1. Podobnie jak w klasycznym SVC, QSVC stara się maksymalizować margines między klasami, jednak dzięki pracy w przestrzeni Hilberta, granica decyzyjna może mieć bardziej złożony kształt, który lepiej odzwierciedla relacje między danymi.

4.: Klasyfikacja nowych danych

Praktyka – Implementacja QSVC w Pythonie

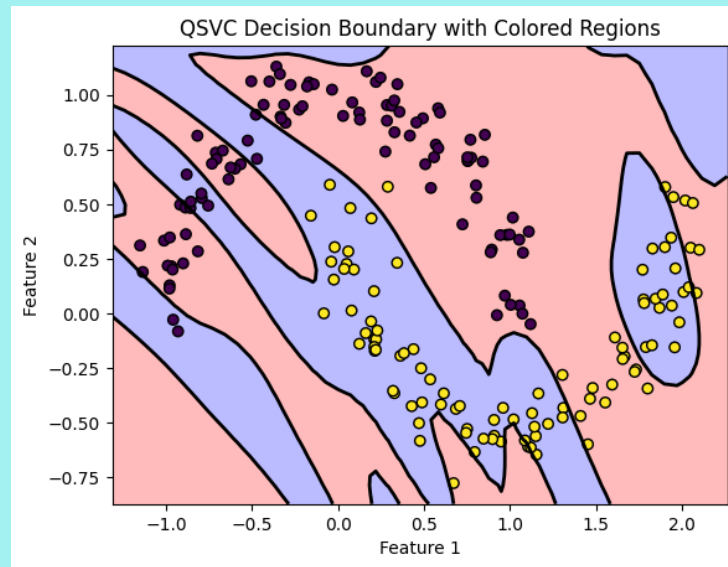
GitHub: https://github.com/tstopa/P-TECH_QC.git

Notebook 1: linear_QSVC_zadanie.ipynb

wspólna praca

Notebook 2: polinomial_QSVC_zadanie.ipynb

praca indywidualna



Podsumowanie i kierunki rozwoju AI i kwantowego uczenia maszynowego

1. Sztuczna inteligencja (AI):

- AI to szeroka dziedzina, która obejmuje różne techniki i metody, pozwalające komputerom wykonywać zadania, które normalnie wymagałyby ludzkiej inteligencji.
- **Uczenie maszynowe** jest kluczową częścią AI, gdzie systemy uczą się na podstawie danych, a nie są ręcznie programowane.

2. Support Vector Classifier (SVC):

- Jest to jeden z popularniejszych algorytmów klasyfikacyjnych, który wykorzystuje hiperpłaszczyznę do oddzielania klas. SVC jest elastyczny i można go dostosowywać poprzez wybór różnych jąder.

3. Kwantowe uczenie maszynowe (Quantum Machine Learning):

- Kwantowe obliczenia mogą przynieść przełom w dziedzinie AI, umożliwiając szybsze przetwarzanie bardziej złożonych problemów.
- QSVC to przykład, jak kwantowe funkcje jądra mogą poprawić klasyfikację nieliniowych problemów.

Kierunki rozwoju AI i kwantowego uczenia maszynowego:

1.AI:

- W przyszłości AI będzie coraz bardziej zintegrowana z naszym codziennym życiem — od inteligentnych asystentów po zaawansowane systemy medyczne.
- Rozwój **samo-uczących się systemów** (self-learning systems), które będą mogły poprawiać swoje działanie bez ingerencji człowieka.

2. Kwantowe obliczenia:

- Obecnie kwantowe komputery są w fazie rozwoju, ale mają potencjał, aby zrewolucjonizować obliczenia w wielu dziedzinach, takich jak chemia, fizyka, uczenie maszynowe i optymalizacja.
- **Praktyczne zastosowanie** kwantowych komputerów w AI może stać się rzeczywistością w ciągu najbliższych 10-20 lat, kiedy technologia stanie się bardziej dostępna.

Q&A