TASK 1:
Ob inem dimensiunile matricei photo i le atribuim variabilelor m i n.
Ini ializăm matricea finală new_X cu dimensiunile m x n, ini ializată cu elemente zero.
Convertim matricea photo la tipul de date double pentru a efectua calculele necesare.
Aplicăm algoritmul SVD (descompunerea în valori singulare) asupra matricei photo. Ob inem trei matrici: matricea de vectori singolari stânga U, matricea diagonală a valorilor singulare S i matricea de vectori sin golari dreapta V.
Reducem dimensiunea matricelor U, S i V pentru a ob ine matrici reduse U_reduced, S_reduced i V_redu ced, păstrând doar primele k coloane.
Calculăm noua matrice new_X ca produsul matricilor U_reduced, S_reduced i V_reduced' (transpusa mat ricei V_reduced), reprezentând aproximarea matricii ini iale photo.
Convertim matricea new_X la tipul de date uint8 pentru a ob ine o imagine validă (valori întregi în intervalu l 0-255).
Returnăm matricea new_X ca rezultat al func iei.

TASK 2:
Ob inem dimensiunile matricei photo i le atribuim variabilelor m i n.
Ini ializăm matricea finală new_X cu dimensiunile m x n, ini ializată cu elemente zero.
Convertim matricea photo la tipul de date double pentru a efectua calculele necesare.
Normalizăm matricea ini ială photo prin scăderea mediei fiecărui rând. Calculăm media fiecărui rând i o atr ibuim variabilei mu. Scădem această medie din fiecare rând al matricei photo.
Construim matricea Z prin transpunerea matricei photo normalizate.
Aplicăm descompunerea în valori singulare (SVD) asupra matricei Z. Ob inem trei matrici: matricea de vec tori singolari stânga U, matricea diagonală a valorilor singulare S i matricea de vectori singolari dreapta V.
Construim matricea W din primele pcs coloane ale matricei V.
Calculăm matricea Y prin înmul irea transpusa matricei W cu matricea photo normalizată.
Aproximăm matricea ini ială photo utilizând matricele W i Y, adăugând înapoi media fiecărui rând mu.
Convertim matricea new_X la tipul de date uint8 pentru a ob ine o imagine validă (valori întregi în intervalul 0-255).
Returnăm matricea new_X ca rezultat al func iei.

TASK 3:
Ob inem dimensiunile matricei photo i le atribuim variabilelor m i n.
Ini ializăm matricea finală new_X cu dimensiunile m x n, ini ializată cu elemente zero.
Convertim matricea photo la tipul de date double pentru a efectua calculele necesare.
Calculăm media fiecărui rând al matricei photo i o atribuim variabilei mu.
Normalizăm matricea ini ială photo prin scăderea mediei fiecărui rând din fiecare element corespunzător.
Calculăm matricea de covarian ă a matricei photo normalizate utilizând formula covarian ei sample-ului.
Ob inem vectorii i valorile proprii ale matricei de covarian ă folosind func ia eig.
Ordonăm valorile proprii în ordine descrescătoare i reordonăm corespunzător vectorii proprii în matricea V .
Păstrăm doar primele pcs coloane din matricea V, ob inând astfel matricea de transformare W.
Construim matricea Y prin înmul irea transpusa matricei W cu matricea photo normalizată.
Calculăm noua matrice new_X ca produsul matricelor W i Y, reprezentând aproximarea matricii ini iale pho to.
Adunăm înapoi media fiecărui rând mu la matricea new_X.
Convertim matricea new_X la tipul de date uint8 pentru a ob ine o imagine validă (valori întregi în intervalul 0-255).
Returnăm matricea new_X ca rezultat al func iei.

TASK 4:
-PREPARE_DATA:
Se define te variabila n cu valoarea 784, reprezentând numărul de pixeli ai unei imagini din setul de date MNIST.
Se ini ializează matricea train_mat cu dimensiunea no_train_images x n, ini ializată cu elemente zero. Ace

asta va stoca imaginile de antrenament.
Se ini ializează vectorul train_val cu dimensiunea 1 x no_train_images, ini ializat cu elemente zero. Acesta va stoca etichetele corespunzătoare imaginilor de antrenament.
Se încarcă datele din fi ierul specificat în variabila data, utilizând func ia load.
Se extrag imaginile de antrenament din datele încărcate i se atribuie matricei train_mat. Imaginile sunt pr eluate din câmpul trainX al variabilei data, limitându-se la primele no_train_images rânduri.
Se extrag etichetele corespunzătoare imaginilor de antrenament i se atribuie vectorului train_val. Etichetel e sunt preluate din câmpul trainY al variabilei data, limitându-se la primele no_train_images elemente.
Se returnează matricea train_mat i vectorul train_val ca rezultate ale func iei.

-VISUALISE_IMAGE:
Se ini ializează matricea finală im cu dimensiunea 28x28, ini ializată cu elemente zero. Aceasta va repreze nta imaginea vizualizată.
Se cite te rândul cu numărul specificat din matricea de antrenament train_mat i se atribuie variabilei image _row.
Se reorganizează rândul citit într-o matrice de dimensiune 28x28 utilizând func ia reshape, apoi se transpu ne matricea rezultată. Astfel, ob inem matricea im care reprezintă imaginea vizualizată.
Se converte te matricea im la tipul de date uint8 pentru a o transforma într-o imagine validă (valori întregi î n intervalul 0-255).
Se afi ează imaginea utilizând func ia imshow.
Func ia se încheie.

-MAGIC_WITH_PCA:
Se ob in dimensiunile matricei de antrenament train_mat i se atribuie variabilelor m i n.
Se ini ializează matricea train cu dimensiunea m x n, ini ializată cu elemente zero. Aceasta va stoca rezulta tul final al transformării PCA.
Se ini ializează vectorul miu cu dimensiunea 1 x n, ini ializat cu elemente zero. Acesta va reprezenta medi a coloanelor matricei de antrenament.
Se ini ializează matricea Y cu dimensiunea m x pcs, ini ializată cu elemente zero. Aceasta va reprezenta m atricea rezultată după schimbarea bazei matricei de antrenament.
Se ini ializează matricea Vk cu dimensiunea n x pcs, ini ializată cu elemente zero. Aceasta va reprezenta matricea de vectori proprii corespunzători componentelor principale.
Se convert e te matricea de antrenament train_mat la tipul de date double.
Se calculează media fiecărei coloane a matricei train_mat utilizând func ia mean i se atribuie vectorului mi u.
Se scade media miu din matricea de antrenament train_mat pentru a o centra în jurul originii.
Se calculează matricea de covarian ă cov_mat a matricei de antrenament train_mat utilizând func ia cov.
Se calculează vectorii i valorile proprii ale matricei de covarian ă cov_mat utilizând func ia eig.
Se sortează valorile proprii în ordine descrescătoare i se creează matricea Vk care con ine vectorii proprii corespunzători sorta i ca coloane.
Se păstrează doar primele pcs coloane din matricea Vk.
Se creează matricea Y prin schimbarea bazei matricei de antrenament utilizând matricea Vk.
Se calculează matricea train ca produsul matricial dintre matricele Y i Vk', reprezentând astfel o aproxima re a matricei de antrenament ini iale.
Func ia returnează matricea train, media miu, matricea Y i matricea Vk ca rezultate ale functiei.

-PREPARE_PHOTO:
Ini ializăm vectorul final sir cu dimensiunea 1 x 784, ini ializat cu elemente zero. Acesta va stoca rezultatul f inal, adică vectorul de pixeli al imaginii pregătite.
Inversăm pixelii imaginii im prin scăderea valorii fiecărui pixel din 255. Această opera ie inversează culorile imaginii.
Transpunem imaginea im pentru a ne asigura că este în formatul corect pentru reshape.
Realizăm opera ia de reshape, transformând matricea într-un vector de pixeli. În acest caz, transpunerea ne asigură că valorile pixelilor vor fi plasate corect în vector.
Atribuim imaginea reshaped vectorului final sir.

Func ia returnează vectorul sir ca rezultat al func iei.

-KNN:
Ini ializăm variabila prediction cu valoarea -1. Aceasta va reprezenta eticheta prezisă pentru datele de test are.
Ini ializăm matricea de distan e distance cu dimensiunea m x 1, unde m reprezintă numărul de exemple de antrenament.
Calculăm distan a Euclidiană între fiecare rând din matricea Y i vectorul de test test. Aceasta se realizeaz ă prin norma diferen ei dintre rândurile respective.
Sortăm distan ele în ordine crescătoare i păstrăm primele k valori. Sortarea este realizată folosind func ia s ort.
Ob inem etichetele corespunzătoare celor k vecini cei mai apropia i i le atribuim variabilei k_nearest_labels .
Calculăm eticheta prezisă (prediction) ca mediană a etichetelor celor k vecini cei mai apropia i. Aceasta se realizează folosind func ia median.
Func ia returnează eticheta prezisă (prediction) ca rezultat al func iei.

-CLASSIFYIMAGE:
Ini ializăm variabila prediction cu valoarea -1. Aceasta va reprezenta eticheta prezisă pentru imaginea de t estare.
Convertim imaginea im la tipul double.
Aplicăm func ia magic_with_pca asupra datelor de antrenament (train_mat), pentru a ob ine matricele train (antrenamentul transformate prin PCA), miu (media coloanelor din train_mat), Y (caracteristicile transform ate prin PCA) i Vk (matricea de transformare bazată pe PCA).
Scădem media coloanelor din train_mat din vectorul imaginii im. Aceasta are scopul de a centra imaginea în jurul originii.
Transformăm baza imaginii prin înmul irea cu matricea Vk. Aceasta realizează proiec ia imaginii într-un nou spa iu bazat pe componente principale.
Calculăm eticheta prezisă (prediction) utilizând metoda KNN cu k = 5. Apelăm func ia KNN i transmitem eti chetele de antrenament (train_val), caracteristicile transformate prin PCA (Y), imaginea de testare (im) i n umărul de vecini (k = 5).
Func ia returnează eticheta prezisă (prediction) ca rezultat al func iei.