Progettazione algoritmica:

Illustrare il processo di progettazione dell'algoritmo per l'analisi dei dati. Descrivere le metodologie, le librerie scientifiche utilizzate e le fasi dell'elaborazione dati, compresa l'identificazione di pattern, l'estrazione di informazioni rilevanti e la generazione di risultati significativi.

Questa applicazione Python utilizza varie librerie per la manipolazione dei dati, l'analisi dei dati e la visualizzazione dei dati, come Pandas, Matplotlib, scikit-learn e PySimpleGUI. Si tratta di un'applicazione completa che include l'accesso e la registrazione degli utenti, la gestione dei dati e l'analisi dei dati.
Qui sotto è spiegato il processo generale della progettazione dell'algoritmo:

Acquisizione dei dati: I dati vengono caricati nel database SQLite dall'utente attraverso un'interfaccia grafica. Questa operazione viene gestita dalla funzione load_csv.
Preprocessing dei dati: Il file CSV caricato viene preprocessato utilizzando la tecnica di one-hot encoding con la funzione pd.get_dummies. Questa tecnica viene utilizzata per convertire le variabili categoriche in un formato che può essere fornito agli algoritmi di Machine Learning per migliorare la previsione.
Suddivisione del set di dati: Il set di dati viene diviso in set di addestramento e test utilizzando la funzione train test split della libreria scikit-learn. Il 80% dei dati viene utilizzato per l'addestramento e il 20% per il test.
Addestramento del modello: Il modello viene addestrato sulla base del set di addestramento utilizzando l'algoritmo Gaussian Naive Bayes (o un RandomForest Classifier, a seconda del codice commentato/non commentato). Questa operazione è gestita dalla funzione train_model.
Valutazione del modello: Il modello addestrato viene valutato utilizzando il set di test e la precisione del modello viene calcolata. Questa operazione è gestita dalla funzione test model.
Predizione: Il modello addestrato viene poi utilizzato per fare previsioni su nuovi dati caricati dall'utente. Questa operazione è gestita dalla funzione open file dialog prediction.
Esportazione dei dati di previsione: I dati di previsione vengono esportati in un file CSV.
Visualizzazione dei dati: Viene creata una visualizzazione sotto forma di istogrammi per confrontare le distribuzioni dei dati originali e dei dati previsti.
Gestione delle query: L'utente può eseguire delle query SQL per ottenere informazioni specifiche dai dati.

La parte di analisi dei dati del codice utilizza la libreria di apprendimento automatico scikit-learn per addestrare e testare i modelli. La parte di visualizzazione dei dati utilizza la libreria matplotlib per creare istogrammi dei dati. Infine, l'intera applicazione è costruita con una GUI utilizzando la libreria Py SimpleGUI.

Progettazione della dashboard:

Descrivere il processo di progettazione della dashboard, inclusa la selezione del template commerciale utilizzato come base. Spiegare come i dati analizzati dall'algoritmo vengono visualizzati nella dashboard e come gli utenti possono interagire con essa.

codice Python completo per progettare una dashboard per gestire l'accesso degli utenti, il caricamento dei dati, l'analisi dei dati e la visualizzazione utilizzando un classificatore di Naive Bayes ottimo per dataset medio piccoli e con variabili non strettamente dipendenti .Un classificatore Gaussiano Naive Bayes (GNB) è un algoritmo di apprendimento automatico supervisionato utilizzato per la classificazione. Si basa sull'assunzione che le caratteristiche (features) dei dati siano distribuite secondo una distribuzione gaussiana (normale) e che queste caratteristiche siano non estremamente legate tra loro. L'utente potrà anche scegliere un classificatore di tipo random forest che è caratterizzato da un insieme di alberi decisionali, che è molto adatto per il caso preso in esame a livello di variabili che entrano in gioco ma avrebbe bisogno di un dataset con un grande numero di record per renderlo apprezzabilmente funzionale.
Ecco come funziona il processo di classificazione con un classificatore GNB
Preparazione dei dati:

- Raccolta del dataset di addestramento che consiste in un insieme di esempi etichettati con le classi di appartenenza.
- Suddivisione del dataset in due parti: dati di addestramento e dati di test. I dati di addestramento verranno utilizzati per addestrare il classificatore, mentre i dati di test verranno utilizzati per valutarne le prestazioni.

Calcolo delle statistiche:

- Per ogni classe nel dataset di addestramento, vengono calcolati i valori medi (mean) e le deviazioni standard (standard deviation) per ogni caratteristica del dataset. Queste statistiche vengono utilizzate per modellare la distribuzione gaussiana di ogni classe.

Apprendimento:

- Durante la fase di addestramento, il classificatore GNB calcola le probabilità a priori di ogni classe nel dataset di addestramento. Queste probabilità a priori vengono calcolate come la frequenza di ogni classe rispetto al numero totale di esempi di addestramento.
- Successivamente, per ogni esempio di addestramento, il classificatore calcola le probabilità condizionate delle caratteristiche dato un'etichetta di classe utilizzando la distribuzione gaussiana.
- Il classificatore GNB utilizza le probabilità a priori e le probabilità condizionate per stimare la probabilità che un dato esempio appartenga a una determinata classe utilizzando il teorema di Bayes.

Classificazione:

- Dopo l'addestramento, il classificatore GNB può essere utilizzato per classificare nuovi esempi.
- Per ogni nuovo esempio, il classificatore calcola le probabilità condizionate delle caratteristiche dato ogni classe utilizzando la distribuzione gaussiana.
- Quindi, utilizzando il teorema di Bayes, il classificatore calcola la probabilità a posteriori che l'esempio appartenga a ogni classe.
- Infine, il classificatore assegna l'esempio alla classe con la probabilità a posteriori più alta.

ecco invece come funziona un classificatore random forest.

è un algoritmo di apprendimento automatico basato su alberi decisionali che combina le previsioni di più alberi per ottenere una classificazione finale. Ecco come funziona il processo di addestramento e la classificazione nel Random Forest:

Creazione degli alberi decisionali: Il Random Forest costruisce un insieme di alberi decisionali. Ogni albero viene creato utilizzando una tecnica chiamata "bootstrapped aggregating" o "bagging". In pratica, vengono selezionati casualmente dei sottoinsiemi con sostituzione dal dataset di addestramento, e su ciascun sottoinsieme viene costruito un albero decisionale.

Suddivisione dei nodi dell'albero: Ogni albero decisionale nel Random Forest viene costruito utilizzando una suddivisione ricorsiva dei nodi. Durante la suddivisione, viene selezionata una variabile predittiva e un punto di suddivisione in base a un criterio,raggiunto un criterio di arresto, ad esempio quando il numero di osservazioni in un nodo è inferiore a una soglia predefinita o quando viene raggiunta una profondità massima.

Voto delle previsioni degli alberi: Una volta che tutti gli alberi sono stati costruiti, per classificare un'istanza di test, ognuno degli alberi emette una previsione. Nel caso di una classificazione, ogni albero "vota" per la sua

previsione di classe. La classe che riceve il maggior numero di voti è la classificazione finale del Random Forest.

Classificazione finale: La classificazione finale nel Random Forest può essere ottenuta semplicemente scegliendo la classe con il maggior numero di voti. In alcuni casi, si possono utilizzare misure di probabilità o punteggi di confidenza per la classificazione.

Random Forest offre diversi vantaggi, tra cui la robustezza rispetto al rumore e ai valori mancanti, la capacità di gestire set di dati di grandi dimensioni con molte variabili e la possibilità di calcolare l'importanza delle variabili.

Il codice include la pre elaborazione dei dati, la codifica dei dati (One-hot Encoding), la suddivisione dei dati in set di addestramento e test e l'implementazione di un modello di classificazione. Inoltre, gestisce il processo di registrazione e accesso degli utenti con SQLite3, incluso l'hashing delle password per aumentare la sicurezza.
L'interfaccia utente grafica (GUI) per questa applicazione è realizzata utilizzando Py SimpleGUI, che rende molto facile e intuitiva la creazione di una GUI in Python. È inoltre possibile esportare i risultati delle previsioni come file CSV e visualizzare i dati con istogrammi.

Integrazione algoritmo e dashboard:

Mostrare come l'algoritmo sviluppato viene collegato alla dashboard, consentendo agli utenti di accedere ai risultati dell'analisi e utilizzare i dati filtrati nel database.

Configurazione del database SQLite: Lo script inizia impostando un database SQLite per archiviare i dettagli degli utenti e i dati. È leggero e non richiede un processo server separato, i programmi eseguono nello stesso spazio di memoria del programma, rendendo SQLite una scelta popolare per l'integrazione nelle applicazioni.
Hashing delle password: Le password non vengono archiviate in chiaro per motivi di sicurezza, ma vengono invece sottoposte a hashing utilizzando l'algoritmo SHA-256.

Registrazione e accesso degli utenti: Gli utenti possono registrarsi con un nome utente e una password, e gli utenti esistenti possono effettuare l'accesso.

Caricamento dei dati: Gli utenti possono caricare un file CSV che contiene i dati da analizzare. I dati vengono quindi elaborati (compresa la codifica one-hot per le variabili categoriche), divisi in set di addestramento e test e archiviati nel database.

Addestramento e test del modello: l'utente sceglie tra un classificatore di Naive Bayes gaussiano oppure un classificatore RandomForest che viene addestrato sui dati caricati. Viene testata la performance del modello e viene visualizzato all'utente il punteggio di accuratezza ottenuto.

Previsione dei dati: Gli utenti possono caricare un altro file CSV per effettuare previsioni utilizzando il modello addestrato. I risultati delle previsioni vengono salvati nel database.

Esportazione dei dati: I dati previsti sono esportati come file in formato CSV.

Visualizzazione dei dati: I dati sono visualizzati utilizzando istogrammi su combinazioni utili.

Ricerca dei dati: Gli utenti possono eseguire una ricerca nel database in base a un nome di colonna specificato e a un termine di ricerca. La ricerca come specificato nella dashboard la ricerca è case sensitive e deve inoltre avere lo stesso nome delle colonne e degli elementi ricercati. L'applicazione risponde con un messaggio di errore dedicato.

Si tratta di un setup relativamente completo e l'applicazione consente l'interazione dell'utente, l'elaborazione dei dati, l'applicazione del modello di machine learning e la visualizzazione dei dati.

ANALISI DATI:

I dataset proposti sono molto esigui il che ha portato potenzialmente un problema di overfitting nel nostro caso specifico, in quanto l'algoritmo predittivo è stato addestrato e testato con un numero molto limitato di record per poter garantire un funzionamento corretto. Infatti l'accuratezza è sospettosamente elevata Nonostante questo problema ,se venisse addestrato successivamente con dataset più consistenti il risultato potrebbe essere apprezzabilmente superiore.
Dai grafici e dalle tabelle è possibile estrapolare informazioni utili a migliorare l'efficienza.

Dall'analisi emerge che i problemi maggiori e più assidui si hanno con i marchi Prada e Gucci.
E' altresi evidente che il quality test fallisce piu spesso in proporzione(pur essendo uno degli elementi meno usati) quando il materiale usato è il metallo , grossi problemi di questo tipo si evidenziano soprattutto con l'acetato e quando questi è colorato di nero il che potrebbe presagire problemi su questa combinazione che andrebbe approfondita ed indagata. Anche se è più fallimentare anche vista la proporzione di elementi prodotti in quantita sensibilmente maggiore(acetato di colore nero) rispetto ad altri materiali e colori. La plastica nei dati presentati e conosciuti non fallisce mai il Test sulla qualità. Per quanto riguarda tipo e dimensione non abbiamo dati sufficienti per esprimere giudizi certi.
In conclusione possiamo con un margine di dubbio associato alla esigua quantità di record di approfondire con controlli mirati le produzioni riguardanti i marchi Prada e Gucci (o almeno i lotti presentati) e indagare su quali problemi possono essere associati al materiale Metallo, in special modo, e alla produzione dei suddetti , cercando di capire se il problema risiede nello stampaggio degli stessi o in altra fase operativa ad esempio stesura del colore. Inoltre avere uno sguardo approfondito su che tipi di metallo vengano utilizzati se alluminio o altre leghe ad esempio. Occorre avviare un indagine anche per i prodotti in acetato di colore nero, al fine di valutare se il fallimento del controllo sulla qualità è imputabile alla maggiore popolazione di esemplari di questo tipo per marca in particolare e le loro singole produzioni.