Università degli Studi di Cagliari
Facoltà di Scienze Economiche Giuridiche e Politiche
Corso di Laurea in Data Science, Business Analytics e Innovazione
2023-2024

PROGETTO LABORATORIO DEI BIG DATA

REPORT: SPOTIFY DATABASE ANALYSIS

ELENA VALDES 11/82/00354

INTRODUZIONE

Il presente progetto si propone di analizzare l'efficacia di diversi modelli di classificazione su un dataset di 112,6MB contenente 170634 osservazioni e 143 features. Il seguente dataset è stato scaricato dal sito Kaggle[1] e contiene tutte le canzoni delle classifiche Daily Top 200 di Spotify in 35+1 paesi (globali) per un periodo di oltre tre anni (2017-2020), e per ogni canzone contiene dati aggregati quali titolo, artista, paese, genere, popolarità, top50, top10 etc.
Il seguente progetto si concentra sul valutare l'accuratezza e la prestazione di tre modelli: regressione logistica, support vector machine (SVM) e rete neurale utilizzando PySpark, un framework per il calcolo distribuito e il trattamento di grandi set di dati.
L'obiettivo è valutare quale modello fornisce le migliori performance nella classificazione dei brani musicali, nello specifico la presenza di un brano nella classifica Top50 nel caso della regressione logistica e della rete neurale, mentre nel caso della SVM se la canzone contiene linguaggio esplicito.

SVOLGIMENTO

Nella classe SpotifyDataProcessor, dopo aver inizializzato la sessione di Spark è stato caricato il dataset nel metodo load_data ed è stato fatto il preprocessamento dei dati col metodo preprocess_data. Considerato che il dataset contiene i dati già puliti e pronti all'utilizzo per l'applicazione dei modelli, è stato sufficiente indicizzare le variabili di tipo stringa convertendole in numeri e vettorizzare i features combinando le colonne indicizzate. Nella prima classe è stato quindi restituito il dataframe pre-elaborato contenente le colonne indicizzate e la colonna "features" con appunto il vettore dei features.

Nella seconda classe "ModelTrainer", si entra nell'addestramento dei modelli, dopo aver creato l'oggetto della classe prendendo come parametro data_processor, necessario per prendere i dati pre-elaborati nella classe precedente, è stato creato il metodo model_coef che addestra il modello usando la tecnica di gridsearch per trovare i migliori parametri con il metodo successivamente definito gparam_grid dove per ogni modello genera la griglia dei parametri appropriata.
In seguito è stato crea l'oggetto per valutare le prestazioni del modello BinaryClassificationEvaluator che considera la colonna delle predizioni grezze in base alla variabile risposta (colonna contenente le etichette reali 1.0 e 0.0) usando come metrica di valutazione l'area sotto la curva ROC la quale misura la capacità del modello di distinguere tra classi positive e negative, e più il valore dell'area è elevato più è corretta la previsione del modello; è stata poi eseguita la validazione incrociata del modello (cross validation) addestrando più modelli su diverse partizioni dei dati (3 folds assegnati). In seguito, è stato applicato il modello migliore trovato dalla Cross Validation per fare la previsione sui dati di input. Sono stati poi estratti i migliori parametri del modello, attraverso il richiamo al metodo gbest_params, in seguito definito, e infine sono stati estratti anche i migliori coefficienti del modello per ogni feature del dataset in base alla variabile risposta, che nel caso della regressione logistica è la variabile "Top_50dummy", nel caso della SVM è "Explicit_true". Nel codice, le variabili risposta sono state logicamente escluse dal calcolo, e i coefficienti dei features sono stati riordinati secondo valore assoluto.

[1] https://www.kaggle.com/datasets/pepepython/spotify-huge-database-daily-charts-over-3-years?select=Final+database.csv

Nella classe ModelEvaluator invece sono state valutate le prestazioni dei tre modelli, prendendo come parametro predictions che, come detto in precedenza, rappresenta le previsioni del modello sull'insieme dei dati. Dopo aver costruito la classe è stato creato il metodo evaluate_model che calcola le metriche di valutazione per ogni modello: accuracy, F1score, precision e recall rispetto al valore reale dei dati di input. Per valutare ulteriormente le prestazioni del modello è stato creato il metodo plot_confusmx per visualizzare la matrice di confusione per il modello specificato; questa mette in relazione le previsioni del modello e i valori effettivi, restituendo per ogni incrocio delle previsioni con la variabile risposta il numero di osservazioni vere positive e vere negative; false positive e false negative.

Il blocco di esecuzione if___name__==”__main__” esegue il codice; crea la sessione di Spark con SparkSession.builder, richiama i metodi della classe SpotifyDataProcessor con il caricamento del file csv e preprocessa i dati. Vengono in seguito addestrati i modelli con il metodo model_coef, richiamando i features e le variabili risposta per ottenerne le predizioni, parametri e coefficienti e calcolandone il tempo di elaborazione per ogni modello.
Viene creato un dizionario “models” che contiene la stringa del nome del modello come chiave e come valori le previsioni, i migliori parametri, i coefficienti e la variabile risposta per ciascun modello. Per la rete neurale non sono stati valutati parametri né coefficienti. Viene quindi iterato sul dizionario “models” con un ciclo for e per ciascun modello sono state stampate le metriche di valutazione, i migliori parametri, i coefficienti e la visualizzazione della matrice di confusione. Infine, viene arrestata la sessione Spark.

RISULTATI

I risultati ottenuti sono riassunti di seguito per ogni modello. Le variabili risposta “Top50_dummy” e “Explicit_true” sono binarie, e nello specifico presentano due etichette: 0.0 (negativa) e 1.0 (positiva).

1. Modello Regressione Lineare

Accuracy	*0.7060533425539023*
F1 Score	*0.586141320246044*
Weighted Precision	*0.7129765678260117*
Weighted Recall	*0.7060533425539023*
regParam	*0.1*
elasticNetParam	*0.0*

La variabile risposta è “Top50_dummy”
Il tempo di elaborazione del modello è stato di 8 minuti e 7 secondi.
Il modello, quindi, ha riportato un'accuratezza pari al 70.6% delle previsioni rispetto ai valori effettivi, l'F1score è stato 58.6% indicando quanto bene il modello si comporta nel bilanciare precision e recall. La precision calcolata è pesata, ossia è stata calcolata considerando il peso delle due classi di risposta del modello, ed è risultata pari al 71.2%: ciò indica la proporzione delle classificazioni reputate effettivamente positive del modello; il recall pesato invece risulta essere pari al 70.6%, in questo caso il recall misura la proporzione delle previsioni corrette del modello rispetto ai valori

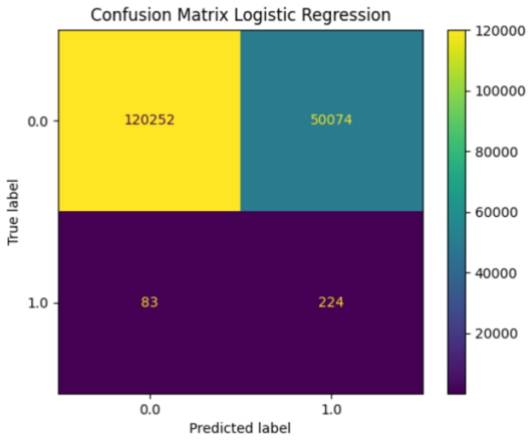
effettivi positivi. Queste ultime due misure verranno analizzate e visualizzate nella matrice di confusione.

Per quanto riguarda i parametri calcolati, il “regParam” che controlla la forza della regolarizzazione L2 (ridge regression) applicata al modello ha riportato valore pari a 0.1, ciò indica che il modello ha avuto una penalizzazione moderata sui coefficienti durante l'addestramento per evitare l'overfitting, cercando di bilanciare quindi tra la complessità del modello e la capacità di generalizzazione dei dati non visti.
Il parametro “elasticNetParam” che invece controlla il mix tra la regolarizzazione L1 (lasso) e L2 sul modello, ha riportato il valore di 0.0, ciò significa che è stata utilizzata solo la regolarizzazione L2, escludendo completamente la regolarizzazione L1, in questo caso il modello tende ad avere una distribuzione uniforme dei coefficienti, dando meno importanza alla selezione delle feature e più importanza alla precisione generale.
Questa combo di parametri riporta quindi un modello bilanciato, che cerca di ottenere una buona precisione generale senza concentrarsi troppo sulla selezione dei features.

I coefficienti del modello indicano l'importanza di ciascuna feature nel modello di regressione logistica. Nello specifico, i primi 10 presi secondo valore assoluto sono:

Album/Single_index	*0.11162132891553857*
Explicit_index	*0.06542688779800432*
Cluster_index:	*-0.05247541120001101*
mode_index	*0.047345553574151725*
Celebrate_index	*0.030628994967937553*
LDA_Topic_index	*0.018135803551197968*
Genre_new_index	*-0.017659935609363544*
Track_number_index	*-0.012396713467940199*
time_signature_index	*-0.011488209056342444*
Tracks_in_album_index	*-0.006995225606904615*

Il coefficiente positivo indica che un aumento nella feature tende ad aumentare la probabilità della classe di interesse; al contrario quando negativo, indica che un aumento nel feature tende a diminuire la probabilità della classe di interesse.
Per quanto riguarda la matrice di confusione in basso:



120252 osservazioni sono state catalogate correttamente come vere negative (la canzone non rientra nella top50 0.0), mentre quelle catalogate vere positive sono 224 (la canzone rientra nella top50 1.0). I falsi positivi risultano essere 50074 osservazioni in cui la classe reale è negativa ma quella predetta è erroneamente positiva (errore di tipo I); i falsi negativi invece risultano essere 83 osservazioni in cui la classe reale è positiva ma quella predetta è negativa (errore di tipo II).

2. *Modello Support Vector Machine*

Accuracy	*0.9997479971635029*
F1 Score	*0.9997480134085053*
Weighted Precision	*0.9997481119724063*
Weighted Recall	*0.9997479971635029*
regParam	*1.0*

La variabile risposta è “Explicit_true”.
Il tempo di elaborazione è stato di 40 minuti e 51 secondi.
Il dataset contenendo 143 features presenta 2 colonne binarie simili a Explicit_true, la feature “Explicit_false” che risulta essere l’opposta a Explicit_true, in quanto per ogni valore positivo di quella true (1.0) presenta il valore opposto negativo (0.0) e viceversa; e la feature “Explicit” anche questa binaria categorica, che presenta i valori “False” e “True”.
Nel testare il modello della SVM inizialmente le due colonne sono state eliminate dal dataset nella riga 50 del codice, nel metodo model_coef:

```
column_names = [col for col in self.data_processor.data.columns
If col.endswith("_index") and col != f"{label_col}_index"
and col != f"Explicit_false_index" and col != f"Explicit"]]
```

Tuttavia le metriche di valutazione del modello SVM non sono cambiate, riportando misure identiche alle attuali, perciò è stato optato di considerare tutte le features ed escludere solamente la variabile risposta.
Tutte le metriche restituiscono il valore vicino al 100%.

Il parametro “regParam” ha riportato un valore pari a 1.0 indicando l'applicazione di una regolarizzazione forte, che può aiutare a prevenire l'overfitting del modello ai dati di addestramento e migliorando la sua capacità di generalizzazione ai dati non visti. Tuttavia, dall’altra parte potrebbe anche ridurre la flessibilità del modello e la sua capacità di adattarsi ai dati di addestramento in modo ottimale.

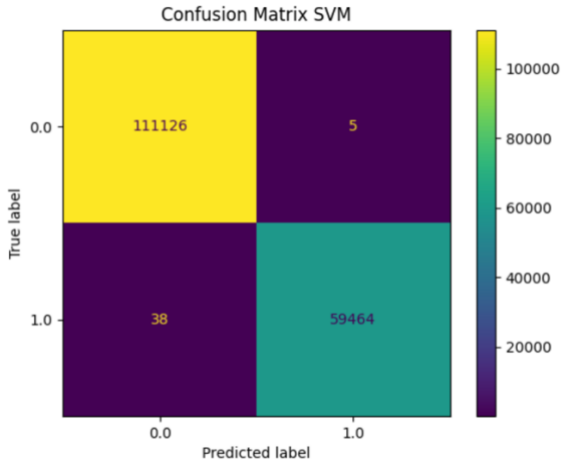
I primi 10 coefficienti riportati per importanza e valore assoluto sono:

Explicit_index	*1.532861595423319*
Album/Single_index	*-0.007310094148246977*
mode_index	*0.006249578560206523*
Celebrate_index	*-0.0038912176237244074*
LDA_Topic_index	*0.0020471902018289905*

Genre_new_index	*-0.001229910517111018*
time_signature_index	*0.0011970418599278867*
Track_number_index	*0.0011175865379366907*
Country0_index	*-0.0007275157292894498*
Cluster_index:	*-0.00039071383002767154*

Si può notare come i primi tre siano stati rilevati come primi più importanti anche nel modello di regressione logistica precedentemente analizzato, sebbene la variabile di risposta sia diversa.

Per quanto riguarda la matrice di confusione:

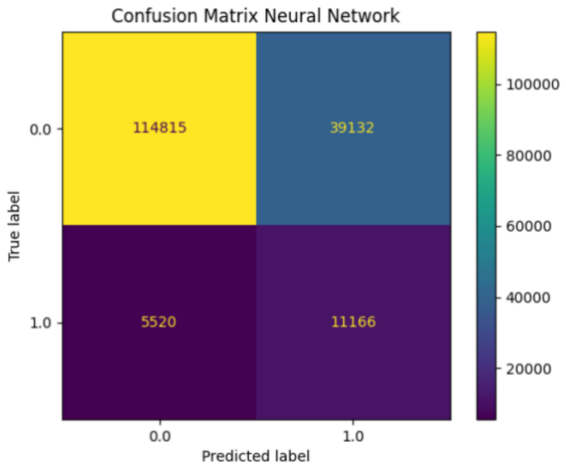


Si può notare come la matrice di confusione confermi le metriche calcolate in precedenza, di fatto i veri negativi sono 111126 i casi in cui la classe reale è negativa (la canzone non contiene linguaggio esplicito 0.0); i veri positivi sono 59464 nei casi in cui la classe reale è positiva (la canzone contiene linguaggio esplicito 1.0).
I falsi positivi risultano essere 5 casi in cui la classe reale è negativa ma quella predetta è erroneamente positiva (errore di tipo I); mentre i falsi negativi risultano essere 38 casi in cui la classe reale è positiva ma quella predetta è negativa (errore di tipo II)

3. *Modello Rete Neurale*

Accuracy	*0.7383155661566051*
F1 Score	*0.6886941704989773*
Weighted Precision	*0.723221678582239*
Weighted Recall	*0.7383155661566052*

La variabile risposta è “Top50_dummy”.
Il tempo di elaborazione è stato di 17 minuti e 3 secondi.
Il modello ha riportato quindi un'accuratezza pari al 73.8% nelle previsioni rispetto ai valori effettivi nel dataset, l’F1score del modello è pari 68.8%, la precision pesata del modello è 72.3%, mentre il recall pesata del modello è pari al 73.8%.
Per quanto riguarda la matrice di confusione in basso:



114815 osservazioni sono state correttamente catalogate come vere negative (la canzone non è presente nella top50 0.0); 11166 sono state catalogate come vere positive (la canzone è presente nella top50 1.0). Le osservazioni false positive risultano essere 39132, qui la classe reale è negativa ma quella predetta è erroneamente positiva (errore di tipo I); le false negative invece risultano essere 5520, qui la classe reale è positiva ma quella predetta è negativa (errore di tipo II).

CONCLUSIONI

Dai risultati ottenuti, emerge che la SVM ha ottenuto le performance migliori, con un'accuracy e un F1 score vicini al 100%. Tuttavia, anche le performance della regressione logistica e della rete neurale sono state comunque molto buone, con accuracy superiori al 70%. La regressione logistica infatti ha dimostrato una buona capacità di generalizzazione, una regolarizzazione moderata con dei parametri che hanno permesso al modello di mantenere un buon equilibrio tra la complessità e la capacità di generalizzazione, contribuendo a migliorare le prestazioni complessive; la SVM invece applicando una regolarizzazione più forte ha ridotto l'overfitting e migliorato la capacità di generalizzazione del modello, approcciando differentemente la gestione della complessità del modello rispetto alla logistica.
I risultati delle metriche sono stati confermati dalle matrici di confusione, la rete neurale e la regressione hanno predetto correttamente un buon numero di osservazioni, mentre la SVM ne ha predetto la quasi totalità.
Osservando i coefficienti delle feature per la regressione logistica e la SVM, possiamo notare che la tipologia della canzone (single, album, compilation) e la presenza di linguaggio esplicito siano state i features che hanno avuto un impatto significativo sulle previsioni dei modelli, così come anche il genere, il “Cluster” di paesi che nel dataset riguardava un raggruppamento in 4 aree geografiche e il numero della traccia. Da notare anche come la presenza di linguaggio esplicito usata come variabile risposta nella SVM sia una feature rilevante nella determinazione delle previsioni sulla top50 della regressione logistica.
Ulteriori ottimizzazioni o modifiche nel tuning dei parametri dei modelli potrebbero portare a miglioramenti delle prestazioni; l’aumento a 5 folds per la cross validation tuttavia non ha prodotto miglioramenti delle metriche.