Reddit Depression Analytics

Studente: Vito Simone Goffredo 277318

Anno accademico 2025/2026

1 Introduzione e obiettivi del progetto

Il progetto "Reddit Depression Analytics" nasce con l'obiettivo di sviluppare una pipeline di Big Data Analytics in grado di analizzare grandi moli di dati testuali non strutturati (post social) per identificare marker linguistici e comportamentali associati al rischio di depressione.

Il sistema non si limita alla semplice classificazione binaria (Depresso/Non Depresso), ma fornisce una dashboard interattiva per l'esplorazione dei dati (EDA) e la validazione di teorie psicologiche note, sfruttando architetture scalabili basate su Apache Spark.

1.1 Il Dataset

Il dataset utilizzato è *eRisk* (Early Risk Prediction on the Internet). I dati sono organizzati in 10 chunk temporali incrementali, simulando l'accumulo storico di attività online per permettere una valutazione predittiva precoce.

La struttura è basata su file XML gerarchici, dove ogni file rappresenta la cronologia completa di un singolo utente anonimo (identificato dal tag `<ID>`, es. "subject1"). Ogni singola interazione dell'utente è racchiusa nel tag `<WRITING>` e contiene i seguenti campi informativi:

- **DATE:** Il timestamp preciso della pubblicazione (es. *2016-02-22 12:16:32*), fondamentale per l'analisi dei ritmi circadiani.
- **TITLE:** Il titolo del thread o della notizia condivisa (es. *"Samsung Galaxy S7 edge vs..."*).
- **TEXT:** Il contenuto testuale vero e proprio del commento o del post personale.
- **INFO:** Metadato sulla fonte (standardizzato come "reddit post").

È importante notare che la struttura eterogenea (alcuni post hanno solo il titolo, altri solo il testo) ha reso necessaria una fase di *Flattening* e pulizia per unificare i campi testuali.

- **Volume Dati:** 544.447 post analizzati.

- **Sbilanciamento Classi:** Forte prevalenza del gruppo di controllo ($\sim$92.5%) rispetto al gruppo a rischio ($\sim$7.5%).

2 Framework Teorico e Letteratura Scientifica

Lo sviluppo del progetto è stato guidato da tre pilastri della letteratura scientifica nel campo del *Digital Phenotyping*, che sono stati utilizzati per validare i risultati visivi della dashboard:

- **Ritmi Circadiani e Insonnia:** La letteratura clinica associa la depressione a disturbi del sonno e a un'inversione del ritmo circadiano (fenomeno noto come "Blue Light Insomnia"). Ci si aspetta un'attività social notturna significativamente più alta nei soggetti a rischio rispetto al gruppo di controllo[1].
- **Marker Linguistici Comportamentali (Ruminazione vs Apatia):** La letteratura scientifica presenta evidenze contrastanti. Mentre alcuni studi (es. *Al-Mosaiwi & Johnstone*) associano la depressione alla "ruminazione" (testi lunghi e ripetitivi), altri evidenziano l'impatto dell'apatia e della letargia, che portano a una riduzione della produzione verbale. Il progetto mira a verificare quale di questi due pattern sia prevalente nel dataset eRisk[2].
- **Semantica Latente (Word Embeddings):** L'approccio moderno all'NLP suggerisce che la frequenza delle parole (*Bag-of-Words*) è insufficiente per catturare il disagio mentale. È necessario analizzare lo spazio semantico (*Word Embeddings*) per associare termini apparentemente neutri a concetti clinici (es. "notte" $\rightarrow$ "tristezza")[3].

3 Architettura del Sistema

Per garantire scalabilità, riproducibilità e rispetto dei requisiti Big Data, è stata adottata un'architettura disaccoppiata ispirata al *Batch Layer* della Lambda Architecture, separando nettamente la fase di elaborazione pesante (Backend) da quella di presentazione (Frontend).

3.1 Backend (ETL, Processing & ML Layer)

Il backend è implementato in Python (**PySpark**) ed è costituito da una pipeline *end-to-end* contenuta nello script `ml_pipeline_semantic.py`, supportata dal

[1] De Choudhury, M., et al. (2013). "Predicting Depression via Social Media". `https://www.researchgate.net/publication/259948193_Predicting_Depression_via_Social_Media`

[2] Al-Mosaiwi, M., & Johnstone, T. (2018). "In an Absolute State". `https://journals.sagepub.com/doi/full/10.1177/2167702617747074`

[3] Resnik, P., et al. (2015). "Beyond LDA: Exploring Supervised Topic Modeling for Depression Detection in Web Logs". `https://aclanthology.org/W15-1212.pdf`

modulo `utils.py` per la configurazione dell'ambiente (gestione delle variabili `HADOOP_HOME` e `JAVA_HOME` su Windows).

Le fasi principali del processo sono:

- **Ingestione:** Parsing distribuito dei file XML gerarchici utilizzando la libreria esterna `com.databricks:spark-xml`. La pipeline gestisce il caricamento massivo dai vari chunk e l'appiattimento (*Flattening*) della struttura nidificata `<WRITING>` in DataFrame tabulari.
- **Data Cleaning & Bilanciamento:** Pulizia del testo tramite rimozione di stop-words custom e normalizzazione. Fondamentale è lo step di bilanciamento delle classi: dato il forte sbilanciamento (92.5% vs 7.5%), viene applicato un *Undersampling* casuale sul gruppo di controllo prima del training per evitare bias nel modello.
- **Machine Learning:** Addestramento di un modello **Word2Vec** (per l'estrazione di feature semantiche) seguito da un **Random Forest Classifier**. Il modello addestrato e le metriche di validazione (F1-Score) vengono persistiti su disco.
- **Aggregazione (Viste Materializzate):** Calcolo distribuito di metriche complesse non gestibili in tempo reale (Heatmap temporali differenziali, statistiche utente aggregate). I risultati vengono salvati in formato **Parquet** e **CSV** ottimizzato per essere consumati dal frontend.

Script di Supporto e Testing: Durante lo sviluppo della pipeline sono stati implementati script ausiliari per validare l'architettura:

- `etl_job_test.py`: Script di testing che verifica il corretto parsing XML del primo chunk (`chunk1/`), utilizzato per validare la configurazione Spark e la corretta estrazione dei campi gerarchici prima di processare l'intero dataset.
- `etl_job.py`: Pipeline ETL completa che implementa l'intero flusso di ingestione, flattening e data cleaning con tokenizzazione NLP (RegexTokenizer + StopWordsRemover). Questo script è stato utilizzato nelle fasi preliminari di sviluppo per esplorare la struttura dei dati e calcolare statistiche globali.

3.2 Frontend (Serving & Visualization Layer)

L'interfaccia utente è realizzata con **Streamlit**, una libreria Python open-source per la creazione rapida di data app.

La dashboard (`app.py`) segue il paradigma *Lazy Loading*: non esegue calcoli pesanti sui dati grezzi, ma "consuma" esclusivamente le viste materializzate generate dal backend. Questo approccio garantisce:

1. **Latenza Zero:** I grafici interattivi (realizzati con **Plotly**) vengono renderizzati istantaneamente indipendentemente dal volume dei dati sottostanti (500k+ post).
2. **Scalabilità:** L'interfaccia rimane reattiva anche se il dataset di training cresce di ordini di grandezza.

4 Evoluzione Metodologica: Dal Sintattico al Semantico

Uno degli aspetti chiave del progetto è stata l'evoluzione della pipeline di Machine Learning, passata attraverso due iterazioni principali per migliorare la capacità di generalizzazione e la robustezza clinica del modello.

4.1 Fase 1: Approccio Baseline (TF-IDF)

Inizialmente, è stato implementato un approccio puramente sintattico basato sulla frequenza dei termini. La pipeline (script `ml_pipeline.py`) utilizzava la tecnica **TF-IDF** (*Term Frequency-Inverse Document Frequency*) per la vettorizzazione, combinata con una **Logistic Regression**.

- **Configurazione:** Vettorizzazione delle top 5.000 parole più frequenti, con filtri statistici per rimuovere termini troppo rari (*minDF=5.0*) o troppo comuni (*maxDF=0.75*).
- **Risultati:** F1-Score estremamente elevato ($\sim$93%).
- **Criticità:** Un'analisi qualitativa ha rivelato un forte *Overfitting* su keyword specifiche. Il modello tendeva a "memorizzare" parole chiave senza comprenderne il contesto (es. classificava correttamente solo se presenti termini espliciti come "depressed", fallendo su descrizioni indirette dei sintomi).

4.2 Fase 2: Approccio Avanzato (Word2Vec)

Per superare i limiti sintattici, l'architettura è stata migrata verso un approccio semantico (script `ml_pipeline_semantic.py`), utilizzando **Word2Vec** (*Distributed Representations of Words*) abbinato a un classificatore **Random Forest**.

- **Logica:** Word2Vec mappa le parole in uno spazio vettoriale denso a bassa dimensionalità (100 dimensioni), dove termini con significati simili sono geometricamente vicini.
- **Risultati:** F1-Score dell'81.88%.

- **Valutazione:** Sebbene numericamente inferiore al baseline, questo modello è stato giudicato qualitativamente superiore per l'applicazione finale. Ha dimostrato una reale capacità di astrazione, collegando autonomamente termini come "sadness" a concetti clinici correlati come "antidepressant", "crushed" e "deprived" (come evidenziato dall'analisi dei sinonimi), garantendo una maggiore robustezza su dati reali non visti.

5 Analisi dei Risultati e Validazione Scientifica

5.1 Analisi Esplorativa Preliminare (WordCloud)

Prima dell'analisi semantica avanzata, è stata condotta un'analisi esplorativa tradizionale tramite WordCloud (script `eda_viz.py`). Questo approccio, basato sulla frequenza delle parole dopo rimozione di una lista custom di stop-words estesa (oltre 100 termini includenti articoli, preposizioni, verbi comuni e residui tecnici come "http", "reddit", "deleted"), ha generato visualizzazioni comparative per i due gruppi.

Le WordCloud risultanti non hanno evidenziato differenze qualitative marcate tra gruppo di controllo e gruppo a rischio, confermando la necessità di un approccio semantico più sofisticato (Word Embeddings) per catturare il disagio mentale oltre la semplice frequenza lessicale. Questa limitazione del metodo bag-of-words ha motivato la scelta dell'architettura Word2Vec per l'analisi finale. La dashboard finale permette di validare visivamente le ipotesi teoriche discusse nella Sezione 2, fornendo riscontri empirici basati sui dati processati.

5.2 Validazione Ritmi Circadiani (Heatmap Differenziale)

L'implementazione di una doppia Heatmap ha permesso di confrontare i pattern temporali di pubblicazione tra i due gruppi.

- **Gruppo di Controllo:** L'attività è concentrata nelle ore diurne e serali, con un netto calo fisiologico nelle ore notturne (02:00 - 06:00), indicando un ciclo sonno-veglia regolare.
- **Soggetti a Rischio:** Si rileva una presenza significativa di attività anche in orari notturni profondi. Questo dato visivo conferma l'ipotesi clinica dell'insonnia e dell'alterazione dei ritmi circadiani come marker comportamentale della depressione.

5.3 Estrazione di Conoscenza Semantica

L'utilizzo di algoritmi avanzati (*Word Embeddings*) ha permesso di estrarre "Concetti Latenti" anziché semplici frequenze di parole. Come visibile nella visualizzazione "Semantic Network Discovery", il modello ha appreso autonomamente associazioni complesse:

- Il termine `anxiety` è stato associato vettorialmente a `ptsd` (comorbilità).

- Sono emersi termini legati alla farmacologia (`adderall`, `medication`) e alla gravità dei sintomi (`severe`, `crippling`).

Questo dimostra che la rete neurale non si è limitata a memorizzare keyword, ma ha appreso il contesto clinico della patologia.

5.4 Analisi Comportamentale: Verbosità vs Apatia

Un risultato significativo emerge dall'analisi dello *Scatter Plot* (Lunghezza Media vs Classificazione).

Contrariamente all'ipotesi della "ruminazione" (verbosità eccessiva) suggerita da parte della letteratura, i dati evidenziano un pattern opposto: i soggetti classificati a rischio si concentrano prevalentemente nella fascia di **bassa lunghezza media** (post brevi), mentre i post molto lunghi appartengono quasi esclusivamente al gruppo di controllo.

Interpretazione: Questo risultato suggerisce che, nel contesto di questo specifico dataset, prevale il sintomo dell'**apatia** o dell'affaticamento cognitivo (Psychomotor Retardation): gli utenti a rischio tendono a interazioni brevi e sforzi comunicativi ridotti, piuttosto che a lunghi sfoghi articolati.

6 Conclusioni

Il progetto ha dimostrato con successo come le tecnologie Big Data (Apache Spark) e le tecniche avanzate di NLP (Word Embeddings) possano essere integrate per il *Digital Phenotyping*.

L'architettura sviluppata ha permesso di processare oltre mezzo milione di post, gestendo efficacemente lo sbilanciamento delle classi e fornendo, attraverso la dashboard, prove visive che correlano i dati digitali con teorie psicologiche consolidate. Il passaggio da un approccio sintattico a uno semantico ha garantito la creazione di un modello non solo accurato, ma anche interpretabile e capace di generalizzazione clinica.