Università Politecnica delle Marche

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

---

CORSO DI DATA SCIENCE

Natural Language Processing per la classificazione di email di phishing ed estrazione delle informazioni

Professore

Prof. Domenico Ursino

Studenti

Matteo Risolo

Niccolò De Pascali

ANNO ACCADEMICO 2025/2026

Indice

Capitolo 1

Introduzione e descrizione del dataset

Il **Natural Language Processing (NLP)** è un ambito dell'intelligenza artificiale e del machine learning che studia metodologie e modelli computazionali per l'analisi, la comprensione e la generazione del linguaggio naturale in forma testuale o parlata. L'obiettivo principale del NLP è consentire ai sistemi automatici di elaborare testi non strutturati, trasformandoli in rappresentazioni formali utili all'estrazione di conoscenza e al supporto di decisioni automatiche.

Tra i principali task del NLP rientra la **Text Classification**, che consiste nell'assegnazione automatica di una o più etichette a un documento testuale sulla base del suo contenuto semantico. Tale task è ampiamente utilizzato in numerosi contesti applicativi, tra cui il filtraggio delle email, l'analisi del sentiment, la categorizzazione di documenti e il rilevamento di contenuti malevoli. Accanto alla classificazione, un ulteriore insieme di task fondamentali è rappresentato dalla **Information Extraction (IE)**, il cui obiettivo è estrarre informazioni strutturate a partire da testo non strutturato. Tra i task più rilevanti di IE si collocano la **Named Entity Recognition (NER)**, finalizzata all'identificazione e classificazione di entità semantiche (come persone, organizzazioni, indirizzi email o URL), e la **Keyphrase Extraction (KPE)**, che

mira a individuare le parole o espressioni più rappresentative del contenuto di un documento.

Il dataset impiegato nel presente progetto è il **Phishing Email Dataset**, disponibile pubblicamente sulla piattaforma Kaggle. Esso è costituito da una collezione di email etichettate come phishing o legittime, rendendolo direttamente utilizzabile per task di classificazione binaria supervisionata.

Ogni istanza del dataset include il contenuto testuale dell'email e una label numerica associata (0 indica un'email legittima, 1 invece indica un'email di phishing), consentendo l'addestramento e la valutazione di modelli di text classification finalizzati al rilevamento automatico di messaggi fraudolenti. Inoltre, la natura testuale dei messaggi, tipica delle comunicazioni di phishing reali, presenta numerosi elementi informativi rilevanti, quali riferimenti a servizi online, URL, indirizzi email, nomi di organizzazioni e frasi persuasive.

Queste caratteristiche rendono il dataset particolarmente adatto anche ai task di Information Extraction, in quanto il testo contiene entità e parole chiave di interesse per l'analisi semantica. Di conseguenza, il dataset si presta efficacemente a uno studio congiunto di Text Classification, Named Entity Recognition e Keyphrase Extraction nel contesto della sicurezza informatica.

Capitolo 2

Pre-processing

La fase di pre-processing costituisce un passaggio fondamentale in un progetto di NLP, poiché consente di trasformare testi grezzi (spesso rumorosi e non omogenei) in un formato più coerente e informativo per l'addestramento e la valutazione dei modelli. In questa relazione il pre-processing viene descritto in due livelli: il primo riguarda un insieme di operazioni comuni applicate all'intero dataset, finalizzate alla pulizia e normalizzazione del contenuto; mentre il secondo racchiude una serie di interventi specifici per ciascun task, distinguendo il flusso dedicato alla Text Classification da quello dedicato alla Information Extraction (KPE e NER), in modo da rispettare le diverse esigenze di rappresentazione e annotazione del testo. I due pre-processing specifici verranno poi approfonditi successivamente nelle relative sezioni riferite ai loro task di appartenenza.

2.1 Pre-processing generale

Il dataset utilizzato è distribuito in più file CSV distinti, ciascuno contenente una porzione delle email raccolte. Pertanto, il primo passo del pre-processing ha riguardato il caricamento sistematico e l'unificazione dei dati in un'unica struttura coerente, al fine di consentire un'analisi uniforme dell'intero dataset

È stato innanzitutto definito il percorso della directory contenente i file CSV del dataset e generato l'elenco ordinato dei file disponibili, al fine di garantire una lettura coerente e riproducibile. I file sono stati quindi caricati sequenzialmente e concatenati verticalmente in un unico DataFrame, ottenendo un corpus unitario di email. Tale operazione consente di gestire il dataset in modo omogeneo e semplifica le successive fasi di pre-processing, analisi ed elaborazione. Infine, è stato effettuato un controllo preliminare sulla struttura del dataset risultante, al fine di verificare la corretta integrazione dei dati e la presenza delle colonne attese. Tale verifica iniziale rappresenta una buona pratica nei flussi di data pre-processing, poiché consente di individuare tempestivamente eventuali anomalie o incongruenze nei dati caricati.

	sender	receiver	date	subject	body	label	urls	text_combined
0	Young Esposito <Young@iworld.de>	user4@gvc.ceas-challenge.cc	Tue, 05 Aug 2008 16:31:02 -0700	Never agree to be a loser	Buck up, your troubles caused by small dimensi...	1	1.0	NaN
1	Mok <ipline's1983@icable.ph>	user2.2@gvc.ceas-challenge.cc	Tue, 05 Aug 2008 18:31:03 -0500	Befriend Jenna Jameson	\nUpgrade your sex and pleasures with these te...	1	1.0	NaN
2	Daily Top 10 <Karmandeep-opengevl@universalnet...	user2.9@gvc.ceas-challenge.cc	Tue, 05 Aug 2008 20:28:00 -1200	CNN.com Daily Top 10	>+=+...	1	1.0	NaN
3	Michael Parker <ivqrnai@pobox.com>	SpamAssassin Dev <xrh@spamassassin.apache.org>	Tue, 05 Aug 2008 17:31:20 -0600	Re: svn commit: r619753 - in /spamassassin/tru...	Would anyone object to removing .so from this ...	0	1.0	NaN
4	Gretchen Suggs <externalsep1@loanofficertool.com>	user2.2@gvc.ceas-challenge.cc	Tue, 05 Aug 2008 19:31:21 -0400	SpecialPricesPharmMoreinfo	\nWelcomeFastShippingCustomerSupport\nhttp://7...	1	1.0	NaN

Figura 2.1: Visualizzazione delle prime 5 colonne del dataset

Successivamente è stata analizzata la distribuzione delle classi nel dataset, calcolando il numero di email associate a ciascuna etichetta. Tale analisi preliminare consente di valutare l'eventuale presenza di squilibri tra le classi, aspetto rilevante nella progettazione e nella valutazione dei modelli di text classification. Dal conteggio emerge una distribuzione complessivamente bilanciata tra email di phishing (label = 1) ed email legittime (label = 0), con una lieve prevalenza dei messaggi di phishing. La visualizzazione tramite grafico a barre permette di rappresentare in modo immediato tale distribuzione.

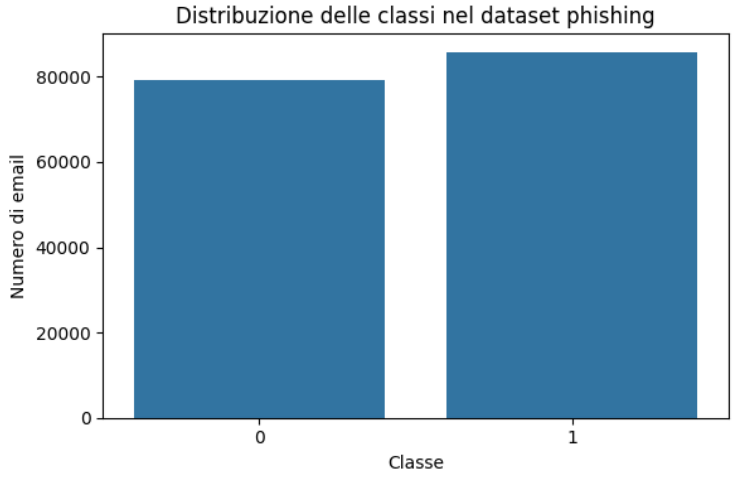

Figura 2.2: Distribuzione delle etichette

Successivamente è stata eseguita una fase di pulizia del dataset, che ha previsto la rimozione delle osservazioni duplicate, al fine di eliminare ridondanze informative che avrebbero potuto influenzare negativamente l'addestramento e la valutazione dei modelli. Inoltre, sono state eliminate le righe contenenti valori mancanti nel campo testuale dell'email, in quanto prive dell'informazione necessaria per l'applicazione delle tecniche di Natural Language Processing. Dopo tali operazioni di pulizia, è stata nuovamente analizzata la distribuzione delle classi.

Tabella 2.1: Distribuzione delle Etichette nel Dataset

Label (Classe)	**Numero di Campioni**
1 (Phishing)	42.890
0 (Benign)	39.595
Totale	**82.485**

I risultati mostrano una distribuzione ancora sostanzialmente bilanciata tra email di phishing e email legittime. Alla luce di questo equilibrio, non è stato ritenuto opportuno applicare tecniche di undersampling o oversampling, poiché l'assenza di un marcato sbilanciamento consente di addestrare i modelli

di classificazione in modo affidabile senza introdurre ulteriori trasformazioni sui dati.

Capitolo 3

Text Classification

3.1 Pre-processing per il Text Classification

Una volta completata la fase di pre-processing comune, il dataset è stato ulteriormente elaborato con operazioni specifiche per il task di Text Classification. In questo contesto, l'obiettivo principale del pre-processing è quello di trasformare il testo grezzo delle email in una rappresentazione adatta all'apprendimento automatico, riducendo il rumore linguistico e preservando le informazioni semanticamente rilevanti per la distinzione tra email di phishing e legittime. Le scelte effettuate in questa fase sono finalizzate a migliorare la qualità delle feature testuali e, di conseguenza, le prestazioni dei modelli di classificazione.

```python
# Creazione della colonna di testo combinato subject + body
data["text_combined"] = (
    data["subject"].fillna("").astype(str) + " " + data["body"].fillna("").astype(str)
).str.strip()

# rimuove righe completamente vuote dopo la concatenazione
data = data[data["text_combined"].str.len() > 0].copy()


# Regex precompilate
URL_RE = re.compile(r"(https?://[^\s]+|www\.[^\s]+)")
EMAIL_RE = re.compile(r"\b[\w\.-]+@[\w\.-]+\.\w+\b")
NUM_RE = re.compile(r"\b\d+(\.\d+)?\b")
HTML_RE = re.compile(r"<[^>]+>")


# Funzione di pulizia del testo
def clean_email_text(text) -> str:
    if pd.isna(text):
        return ""
    text = str(text).lower()

    # Rimuove tag HTML
    text = HTML_RE.sub(" ", text)

    # Normalizza URL, Email e Numeri
    text = URL_RE.sub(" __URL__ ", text)
    text = EMAIL_RE.sub(" __EMAIL__ ", text)
    text = NUM_RE.sub(" __NUM__ ", text)

    # Pulizia spazi
    text = re.sub(r"\s+", " ", text).strip()
    return text
```

Figura 3.1: Codice di preprocessing per la Text Classification

Come si evince dall'immagine precedente, per il task di classificazione testuale è stato innanzitutto costruito un campo testuale unico, combinando oggetto *subject* e corpo del messaggio *body* in una nuova colonna text_combined. Tale scelta consente di sfruttare congiuntamente due porzioni informative dell'email: l'oggetto, spesso sintetico e persuasivo nelle campagne di phishing, e il corpo, che contiene il contenuto principale del messaggio. Eventuali valori mancanti sono stati gestiti sostituendoli con stringhe vuote, e sono state rimosse le osservazioni che risultavano completamente prive di testo anche dopo la concatenazione.

Successivamente è stata applicata una procedura di normalizzazione e pulizia del testo mediante una funzione dedicata *clean_ email_ text*. In particolare, il testo è stato convertito in minuscolo per ridurre la variabilità dovuta alla capitalizzazione e sono stati rimossi eventuali tag HTML, che possono introdurre rumore e non apportano informazione semantica utile alla classificazione. Inoltre, alcuni pattern ricorrenti e altamente variabili nel dominio email (URL, indirizzi email e numeri) sono stati ricondotti a token standardizzati

$\_\_URL\_\_$, $\_\_EMAIL\_\_$, $\_\_NUM\_\_$. Questa scelta consente di preservare l'informazione "di presenza" di tali elementi, riducendo al contempo la frammentazione del vocabolario causata da stringhe uniche (ad es. URL diversi tra email), con un potenziale beneficio in termini di generalizzazione dei modelli.

A completamento del pre-processing, è stata effettuata una normalizzazione degli spazi, collassando sequenze multiple e rimuovendo spazi iniziali/-finali. Parallelamente, sono state derivate due feature numeriche num_urls, num_emails che conteggiano rispettivamente il numero di URL e di indirizzi email presenti nel testo combinato, utilizzate a fini di ispezione e analisi descrittiva delle caratteristiche tipiche delle email di phishing rispetto a quelle legittime. Infine, è stata creata la colonna definitiva text_clean, utilizzata come input testuale pulito nelle fasi successive di modellazione.

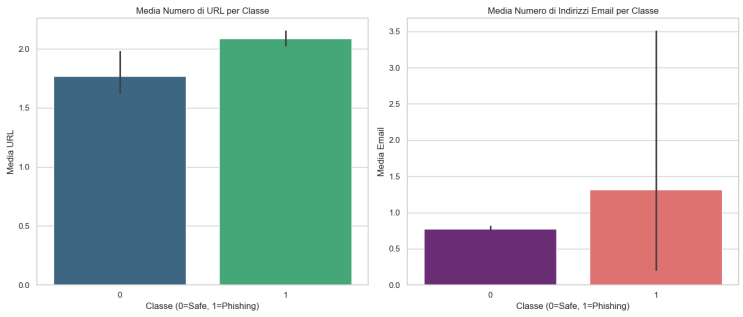

Figura 3.2: Media numero URL per classe e Media numero email per classe

Come detto in precedenza, le feature num_urls e num_emails sono state derivate esclusivamente a scopo esplorativo e descrittivo, al fine di analizzare alcune caratteristiche strutturali tipiche delle email e di supportare la visualizzazione dei dati. Dai risultati emerge che le email di phishing presentano in media un numero maggiore di URL rispetto alle email legittime, confermando una caratteristica tipica di questo tipo di messaggi, spesso finalizzati a reindirizzare l'utente verso siti malevoli. Analogamente, anche il numero medio di indirizzi email risulta più elevato nella classe phishing, sebbene con

una variabilità maggiore, indicata dall'ampiezza delle barre di errore. Queste feature non sono state direttamente utilizzate come input nei modelli di text classification, ma hanno fornito un utile riscontro empirico sulla coerenza del dataset e sulle caratteristiche distintive dei messaggi di phishing.

3.2 Introduzione

Completata la fase di pre-processing, l'analisi si è concentrata sul task di **Text Classification**, con l'obiettivo di sviluppare e confrontare diversi modelli in grado di distinguere automaticamente tra email di phishing ed email legittime. Nel presente lavoro sono stati adottati sia approcci tradizionali di machine learning, sia modelli basati su reti neurali profonde, al fine di confrontarne le prestazioni e analizzarne i punti di forza e di debolezza sul medesimo dataset. In particolare, sono stati impiegati modelli di **Naive Bayes**, **Logistic Regression** e **Support Vector Machine** (SVM), che rappresentano approcci consolidati nella classificazione testuale basata su rappresentazioni vettoriali del testo. Tali modelli sono ampiamente utilizzati in letteratura per la loro efficienza computazionale, interpretabilità e buone prestazioni in contesti di classificazione binaria su dati testuali. Accanto agli approcci di classificazione tradizionali, è stato impiegato un modello basato su rappresentazioni distribuzionali dei testi ottenute tramite **Word2Vec**, utilizzate come input per un classificatore di tipo Logistic Regression. Parallelamente, sono stati adottati modelli neurali più avanzati, ovvero **Long Short-Term Memory (LSTM)** e **Convolutional Neural Networks** (**CNN**), capaci di modellare rispettivamente le dipendenze sequenziali nel testo e i pattern locali rilevanti per la classificazione.

3.3 Naive Bayes

Il modello **Naive Bayes** è stato utilizzato come primo approccio di classificazione testuale, in quanto rappresenta una baseline consolidata nel Natural Language Processing per problemi di classificazione binaria su dati testuali. Il dataset è stato suddiviso in training set e test set mediante split stratificato, preservando la distribuzione originale delle classi. Il testo è stato rappresentato tramite **TF-IDF**, considerando unigrammi e bigrammi, con l'obiettivo di catturare sia informazione lessicale sia brevi pattern testuali rilevanti.

L'ottimizzazione del modello è stata effettuata mediante **Grid Search con validazione incrociata a 5 fold**, selezionando il parametro di smoothing α sulla base della metrica F1-score macro. Il modello ottimizzato è stato infine valutato sul test set, producendo le metriche di classificazione utilizzate come riferimento per il confronto con gli approcci successivi.

Tabella 3.1: Performance del Modello Naive Bayes

Classe	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Benign)	0.9808	0.9948	0.9878	7.919
1 (Spam)	0.9952	0.9820	0.9886	8.578
Accuracy		0.9882		16.497
Macro Avg	0.9880	0.9884	0.9882	16.497
Weighted Avg	0.9883	0.9882	0.9882	16.497

La valutazione sul test set conferma le buone prestazioni del classificatore. Il modello raggiunge un'accuratezza complessiva pari a 0.9882, con valori di F1-score molto elevati e comparabili per entrambe le classi. In particolare, le email legittime (classe 0) presentano un recall superiore, mentre le email di phishing (classe 1) mostrano una precisione leggermente più elevata, evidenziando un buon compromesso tra individuazione dei messaggi malevoli e contenimento dei falsi positivi.

Nel complesso, i risultati ottenuti dimostrano che il modello Naive Bayes, nonostante la sua semplicità, risulta altamente efficace nel contesto del rilevamento automatico del phishing, costituendo una solida baseline di riferimento per il confronto con i modelli più complessi analizzati successivamente.

3.4 Logistic Regression

Il modello di **Logistic Regression** è stato utilizzato come approccio di classificazione lineare supervisionata, ampiamente impiegato nella text classification per la sua capacità di modellare efficacemente problemi di classificazione binaria su dati ad alta dimensionalità. Anche in questo caso il testo è stato rappresentato tramite TF-IDF con unigrammi e bigrammi. Per tenere conto di eventuali residui squilibri tra le classi, è stato utilizzato il parametro *class_weight="balanced"*. L'ottimizzazione del modello è stata effettuata mediante Grid Search sul parametro di regolarizzazione C, utilizzando una validazione incrociata a 5 fold e la metrica F1-score macro. Il modello ottimizzato è stato quindi valutato sul test set, producendo le metriche di precision, recall e F1-score utilizzate per il confronto con gli altri approcci.

Tabella 3.2: Performance della Logistic Regression)

Classe	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Benign)	0.9909	0.9913	0.9911	7.919
1 (Spam)	0.9920	0.9916	0.9918	8.578
Accuracy		0.9915		16.497
Macro Avg	0.9914	0.9914	0.9914	16.497
Weighted Avg	0.9915	0.9915	0.9915	16.497

La valutazione sul test set conferma le ottime prestazioni del modello di Logistic Regression, che raggiunge un'accuratezza pari a 0.9915, accompagnata da valori di precision, recall e F1-score elevati e sostanzialmente bilanciati

tra le due classi. In particolare, il modello dimostra una capacità molto simmetrica di classificazione, riuscendo a individuare con elevata affidabilità sia le email legittime sia quelle di phishing. Questo comportamento indica una riduzione efficace sia dei falsi positivi (email legittime erroneamente classificate come phishing) sia dei falsi negativi (email di phishing non rilevate), aspetto particolarmente rilevante nel contesto della sicurezza informatica. Rispetto alla baseline fornita dal Naive Bayes, il modello mostra un miglioramento consistente e più stabile delle metriche di valutazione, confermandosi come un solido punto di riferimento per il confronto con gli approcci di deep learning presentati nelle sezioni successive.

3.5 SVM

Successivamente si è applicato il modello **Support Vector Machine**, il quale risulta essere particolarmente adatto alla classificazione di testi rappresentati in spazi vettoriali ad alta dimensionalità. Analogamente agli altri approcci, il testo è stato rappresentato tramite TF-IDF con unigrammi e bigrammi, ed il parametro di regolarizzazione C è stato selezionato tramite Grid Search utilizzando l'F1-score macro come metrica di riferimento. L'uso della pesatura delle classi consente inoltre di mantenere un comportamento bilanciato del classificatore rispetto alle due etichette.

Tabella 3.3: Performance del modello SVM

Classe	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Benign)	0.9932	0.9932	0.9932	7.919
1 (Spam)	0.9937	0.9937	0.9937	8.578
Accuracy		0.9935		16.497
Macro Avg	0.9934	0.9934	0.9934	16.497
Weighted Avg	0.9935	0.9935	0.9935	16.497

La valutazione sul test set conferma le ottime prestazioni del modello SVM, che raggiunge un'accuratezza complessiva pari a 0.9935, accompagnata da valori di precision, recall e F1-score estremamente elevati e sostanzialmente identici per entrambe le classi. Questo comportamento indica una classificazione molto equilibrata, con una riduzione ulteriore sia dei falsi positivi sia dei falsi negativi rispetto a Naive Bayes e Logistic Regression. Il confronto con i modelli precedentemente analizzati, in particolare con la Logistic Regression, evidenzia un miglioramento marginale ma costante delle prestazioni complessive. Tale risultato suggerisce una maggiore efficacia dell'SVM nel separare le due tipologie di email a partire dalla rappresentazione testuale TF-IDF. Alla luce di queste evidenze, l'SVM viene identificato come il miglior modello classico tra quelli considerati e selezionato per un'analisi più approfondita mediante matrice di confusione.

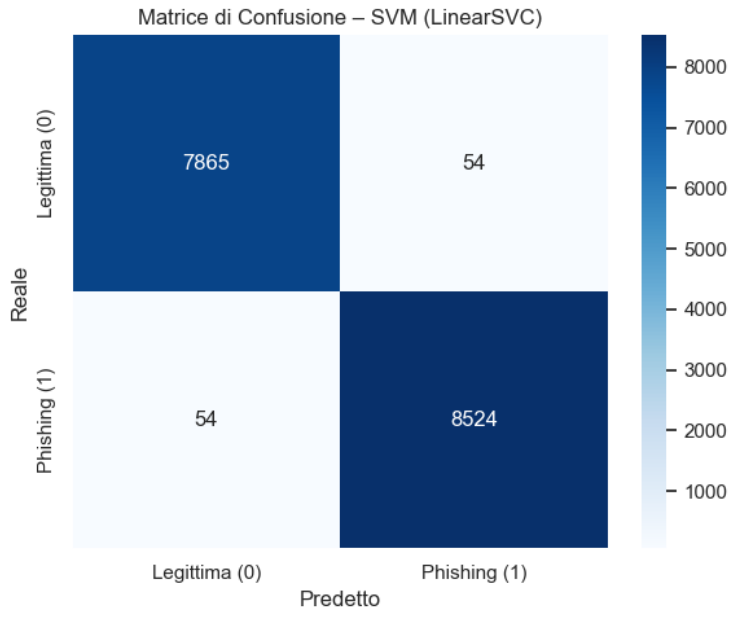

Figura 3.3: Matrice di confusione del modello SVM

Dalla matrice emerge che 7.865 email legittime e 8.524 email di phishing sono state correttamente classificate, a fronte di un numero molto contenuto di errori. In particolare, si osservano 54 falsi positivi (email legittime classi-

ficate come phishing) e 54 falsi negativi (email di phishing classificate come legittime).

La simmetria tra le due tipologie di errore evidenzia un comportamento estremamente bilanciato del classificatore, coerente con i valori elevati e uniformi di precision e recall osservati nel classification report. Questo aspetto è particolarmente rilevante nel dominio della sicurezza informatica, in quanto indica che il modello riesce a contenere sia il rischio di bloccare comunicazioni legittime sia quello, più critico, di non individuare messaggi di phishing.

Nel complesso, l'analisi della matrice di confusione conferma la solidità delle prestazioni dell'SVM e supporta la sua identificazione come miglior modello classico tra quelli analizzati.

3.6 Word2vec

Word2Vec è una tecnica di rappresentazione distribuzionale del linguaggio che consente di mappare le parole in vettori numerici densi, detti word embeddings, appresi automaticamente a partire dal contesto di utilizzo delle parole all'interno di un corpus testuale. A differenza delle rappresentazioni come TF-IDF, gli embedding catturano relazioni semantiche e sintattiche latenti tra i termini, consentendo di rappresentare parole semanticamente simili con vettori vicini nello spazio vettoriale. L'utilizzo di Word2Vec nel presente lavoro è motivato dalla volontà di valutare un approccio alternativo alla rappresentazione testuale, in grado di andare oltre la mera frequenza dei termini e di incorporare informazioni semantiche apprese dal contesto.

3.6.1 Pre-processing

Per prima cosa il testo pulito (*text_ clean*) è stato inizialmente trasformato in sequenze di token mediante una tokenizzazione lessicale. Sul corpus così ottenuto è stato quindi addestrato un modello Word2Vec, finalizzato all'apprendi-

mento di embedding delle parole che rappresentano ciascun termine come un vettore numerico denso di dimensione 200. Tale scelta consente di catturare in modo più ricco le relazioni semantiche tra i termini rispetto alle rappresentazioni basate esclusivamente sulla frequenza. Il modello è stato addestrato in modalità *skip-gram*, che apprende le rappresentazioni delle parole a partire dai contesti in cui compaiono, risultando particolarmente efficace su testi eterogenei come le email. Poiché inoltre Word2Vec fornisce rappresentazioni a livello di parola, per ottenere una rappresentazione a livello di documento è stato costruito un embedding dell'email calcolando la media dei vettori delle parole presenti nel testo. Nel caso in cui un'email non contenga termini presenti nel vocabolario del modello, viene assegnato un vettore nullo, garantendo una rappresentazione numerica consistente per tutte le osservazioni. I vettori così ottenuti costituiscono la matrice di input utilizzata nella fase di classificazione. A valle della trasformazione in embedding, i dati sono stati suddivisi in training e test set mediante split stratificato. La fase di classificazione è stata effettuata utilizzando un modello di Logistic Regression, ottimizzato tramite Grid Search sul parametro di regolarizzazione C e valutato mediante F1-score macro, in modo coerente con le configurazioni adottate nei modelli precedenti. La scelta del modello è ricaduta sul Logistic Regression in quanto particolarmente adatto alla gestione di feature dense e in grado di apprendere efficacemente un confine decisionale lineare nello spazio semantico degli embedding. Gli embedding Word2Vec producono infatti rappresentazioni vettoriali dense e continue delle parole, che, una volta aggregate a livello di documento, generano feature a dimensionalità relativamente contenuta e prive di sparsità. Al contrario, sebbene le SVM siano eccellenti per gestire l'alta dimensionalità e la sparsità tipiche del TF-IDF, nel contesto degli embedding densi tendono a offrire prestazioni analoghe alla Logistic Regression a fronte di una maggiore complessità computazionale. Per questo alla fine si è optato per procedere con il modello di Logistic Regression

3.6.2 Risultati ottenuti

Tabella 3.4: Performance del modello Word2Vec + Logistic Regression

Classe	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Benign)	0.9690	0.9686	0.9688	7.919
1 (Spam)	0.9710	0.9714	0.9712	8.578
Accuracy		0.9701		16.497
Macro Avg	0.9700	0.9700	0.9700	16.497
Weighted Avg	0.9701	0.9701	0.9701	16.497

La valutazione sul test set evidenzia un'accuratezza complessiva pari a 0.9701, con valori di precision, recall e F1-score elevati e ben bilanciati tra le due classi. Il modello mostra quindi una buona capacità di discriminazione tra email di phishing ed email legittime, confermando che le rappresentazioni distribuzionali basate su embedding di parole contengono informazione utile per il task di classificazione.

Tuttavia, il confronto con i modelli basati su TF-IDF evidenzia prestazioni inferiori rispetto ai migliori approcci classici (in particolare SVM e Logistic Regression). Tale risultato è riconducibile al fatto che la rappresentazione a livello di documento, ottenuta come media degli embedding delle parole, tende a perdere informazione strutturale e discriminativa presente nelle sequenze testuali, risultando meno efficace nel catturare pattern specifici del phishing. Nonostante ciò, l'approccio Word2Vec rappresenta un valido punto di confronto per valutare l'impatto di rappresentazioni semantiche dense rispetto a quelle basate su frequenza.

3.7 LSTM

Per valutare un approccio di deep learning alla classificazione delle email, è stato implementato un modello basato su **Long Short-Term Memory (LSTM)**, architettura particolarmente indicata per l'elaborazione di testi. Per poter procedere il dataset è stato suddiviso in training e test set mediante split stratificato, e successivamente dal training è stato ricavato un validation set (10%) per monitorare l'andamento dell'apprendimento e supportare la selezione del modello migliore. La trasformazione del testo in sequenze è stata effettuata tramite il layer TextVectorization, configurato con un vocabolario massimo di 50.000 token e una lunghezza massima delle sequenze pari a 300. L'architettura del modello prevede un layer di Embedding con dimensione 128, che apprende automaticamente una rappresentazione vettoriale dei token durante la fase di addestramento. Questa scelta consente al modello di imparare embedding ottimizzati specificamente per il task di classificazione delle email di phishing, a differenza di Word2Vec, che produce rappresentazioni statiche delle parole apprese indipendentemente dall'obiettivo finale. L'embedding integrato nel modello permette quindi di adattare le rappresentazioni delle parole al contesto e alle esigenze del classificatore. A valle dell'embedding, viene utilizzato un layer LSTM con 64 unità, in grado di modellare l'informazione sequenziale presente nel testo. Sono stati inoltre introdotti layer di dropout per ridurre il rischio di overfitting. Inoltre, è stata adottata una strategia di Early Stopping, basata sull'andamento della loss di validazione, al fine di interrompere l'addestramento quando non si osservano ulteriori miglioramenti e prevenire anche in questo caso fenomeni di overfitting.

Tabella 3.5: Performance del modello Deep Learning (LSTM)

Classe	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Benign)	0.9881	0.9830	0.9855	7.919
1 (Spam)	0.9843	0.9890	0.9867	8.578
Accuracy		0.9861		16.497
Macro Avg	0.9862	0.9860	0.9861	16.497
Weighted Avg	0.9861	0.9861	0.9861	16.497

La valutazione del modello LSTM sul test set evidenzia prestazioni complessivamente elevate, con un'accuratezza pari a 0.9861 e valori di precision, recall e F1-score ben bilanciati tra le due classi. In particolare, il modello mostra una buona capacità di individuare le email di phishing, mantenendo al contempo un numero contenuto di errori sulle email legittime.

Il confronto con i modelli classici basati su TF-IDF evidenzia come l'approccio LSTM non superi le migliori prestazioni ottenute con SVM, ma fornisca comunque risultati competitivi. Tale comportamento suggerisce che, nel contesto specifico del dataset analizzato, le architetture di deep learning basate su LSTM riescono a catturare efficacemente l'informazione sequenziale del testo, pur risultando più sensibili alla complessità del modello e alla quantità di dati disponibili.

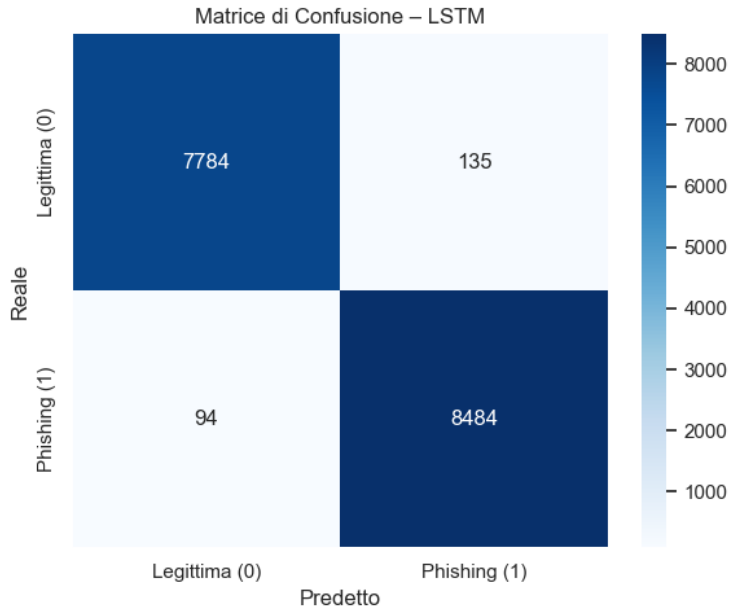

Figura 3.4: Matrice di confusione del modello LSTM

La matrice di confusione del modello LSTM mostra un'elevata capacità di classificazione su entrambe le classi. In particolare, 7.784 email legittime e 8.484 email di phishing sono state correttamente classificate. Gli errori risultano contenuti, con 135 falsi positivi (email legittime classificate come phishing) e 94 falsi negativi (email di phishing classificate come legittime).

La distribuzione degli errori evidenzia un comportamento complessivamente bilanciato del modello, sebbene il numero di falsi positivi risulti leggermente superiore rispetto a quello dei falsi negativi. Questo aspetto è coerente con i valori di recall osservati nel classification report e indica una maggiore cautela del modello nel classificare i messaggi come legittimi, aspetto rilevante nel contesto del rilevamento del phishing.

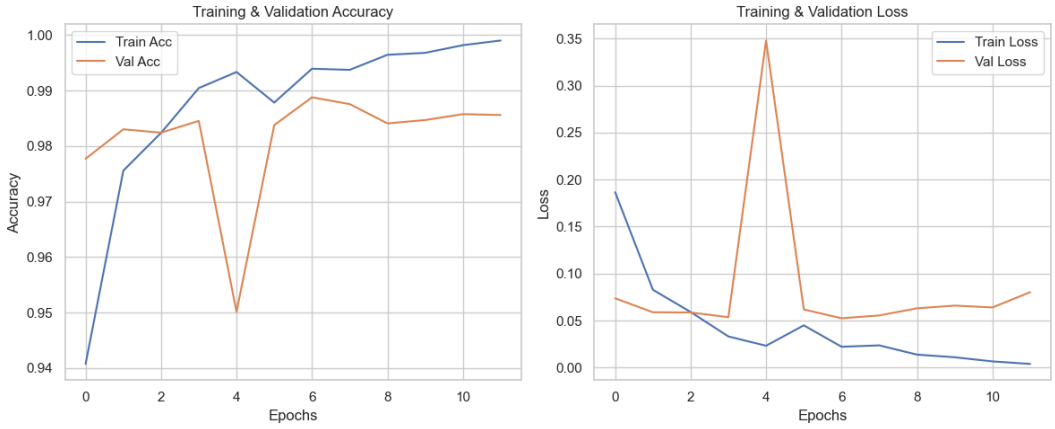

Figura 3.5: Curve di Accuracy e Loss del modello LSTM

L'analisi delle curve di accuracy e loss durante le epoche di addestramento mostra una rapida crescita delle prestazioni sul training set, accompagnata da un andamento più stabile sul validation set. In particolare, l'accuracy di training aumenta progressivamente fino a valori prossimi all'unità, mentre l'accuracy di validazione si stabilizza su valori elevati, senza mostrare un degrado sistematico.

La curva della loss evidenzia una diminuzione costante sul training set, mentre sul validation set si osserva un andamento più irregolare, con un picco isolato seguito da una stabilizzazione. Tale comportamento suggerisce la presenza di una moderata variabilità durante l'addestramento, efficacemente controllata dall'utilizzo di Early Stopping, che ha consentito di selezionare un modello in grado di generalizzare in modo adeguato senza incorrere in fenomeni di overfitting marcato.

3.8 CNN

Come ulteriore approccio di deep learning per la text classification è stato implementato un modello basato su **Convolutional Neural Network (CNN)**, architettura frequentemente impiegata in ambito NLP per individuare pattern

locali nel testo (sequenze brevi di token assimilabili a n-gram informativi). Anche in questo caso, il testo viene trasformato in sequenze di interi tramite TextVectorization, con vocabolario massimo pari a 50.000 token e lunghezza fissata a 300; il vocabolario viene costruito esclusivamente sui dati di training.

Il modello utilizza un layer di Embedding (dimensione 128) appreso direttamente durante l'addestramento, per ottenere rappresentazioni dense dei token ottimizzate per il compito di classificazione. La componente convoluzionale è composta da due blocchi *Conv1D + MaxPooling1D*. I livelli convoluzionali *Conv1D* hanno il compito di individuare brevi sequenze di parole ricorrenti nel testo, tipiche delle email di phishing, mentre il *MaxPooling1D* riduce la quantità di informazione da elaborare, mantenendo solo le caratteristiche più significative individuate localmente, eliminando ridondanze e rumore. L'aggregazione finale delle feature avviene tramite *GlobalMaxPooling1D*, il quale riassume infine l'intera email in un unico vettore, conservando per ciascuna caratteristica il valore massimo rilevato, così da indicare se una determinata sequenza rilevante è presente nel messaggio, indipendentemente dalla sua posizione. Questo vettore compatto viene poi utilizzato per la classificazione finale, inoltre anche in questo caso, è stata applicata una strategia di Early Stopping basata sulla loss di validazione, al fine di selezionare la configurazione con migliore capacità di generalizzazione.

La valutazione del modello CNN sul test set evidenzia prestazioni elevate, con un'accuratezza pari a 0.9868 e valori di precision, recall e F1-score complessivamente bilanciati tra le due classi. In particolare, il modello mostra un recall molto elevato sulla classe phishing, indicando una forte capacità di individuare correttamente i messaggi malevoli, a fronte di una leggera riduzione del recall sulla classe legittima.

Il confronto con il modello LSTM mostra prestazioni complessive simili, con una lieve differenza nel profilo degli errori: la CNN tende a ridurre i falsi negativi (phishing non rilevato), privilegiando la sensibilità verso la classe phishing,

Tabella 3.6: Performance del modello Deep Learning (CNN)

Classe	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Benign)	0.9937	0.9788	0.9862	7.919
1 (Spam)	0.9807	0.9943	0.9874	8.578
Accuracy		0.9868		16.497
Macro Avg	0.9872	0.9865	0.9868	16.497
Weighted Avg	0.9869	0.9868	0.9868	16.497

mentre l'LSTM presenta un comportamento più bilanciato tra le due classi. Tuttavia, entrambi i modelli di deep learning risultano inferiori, in termini di accuratezza complessiva, rispetto al miglior modello classico individuato (SVM).

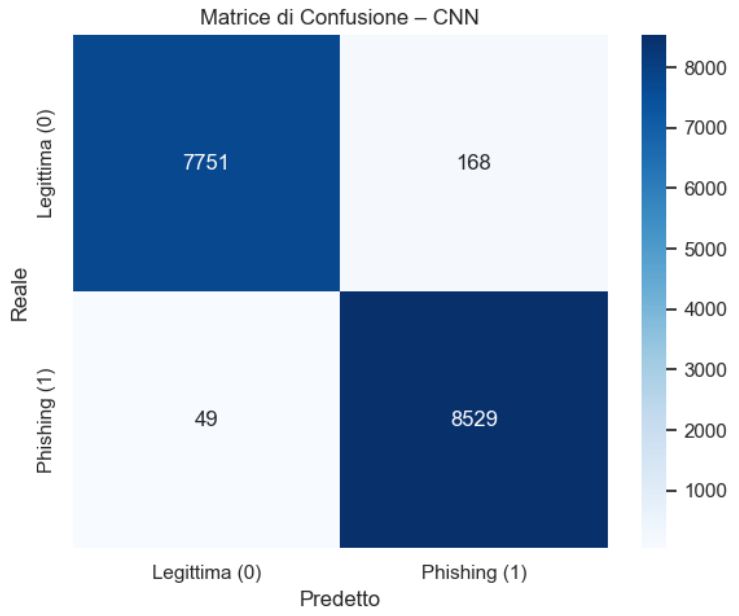

Figura 3.6: Matrice di confusione del modello CNN

La matrice di confusione del modello CNN evidenzia una buona capacità di individuazione delle email di phishing. In particolare, 8.529 email di phishing e 7.751 email legittime sono state correttamente classificate. Gli errori risultano complessivamente contenuti, con 49 falsi negativi (email di phishing classificate

come legittime) e 168 falsi positivi (email legittime classificate come phishing).

La distribuzione degli errori mostra che il modello privilegia la sensibilità verso la classe phishing, riducendo il numero di falsi negativi rispetto al modello LSTM, a fronte di un incremento dei falsi positivi. Tale comportamento è coerente con le metriche osservate nel classification report e risulta particolarmente rilevante nel contesto del rilevamento del phishing, dove la mancata individuazione di messaggi malevoli rappresenta l'errore più critico.

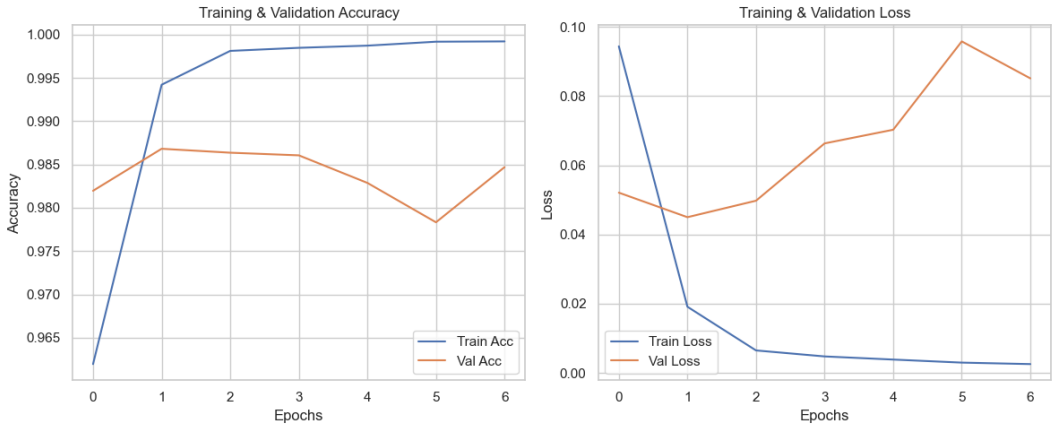

Figura 3.7: Curve di Accuracy e Loss del modello CNN

L'analisi delle curve di accuracy mostra una rapida convergenza del modello sul training set, con valori prossimi all'unità già dopo poche epoche. L'accuracy di validazione si mantiene su livelli elevati ma presenta una leggera flessione nelle epoche successive, suggerendo una tendenza all'overfitting.

Analogamente, la curva della loss evidenzia una riduzione marcata sul training set, mentre sul validation set si osserva un progressivo aumento dopo le prime epoche. Questo comportamento conferma la presenza di overfitting, efficacemente contenuto grazie all'adozione della strategia di Early Stopping, che ha consentito di selezionare un modello con un buon compromesso tra capacità di apprendimento e generalizzazione.

Capitolo 4

Information Extraction

4.1 Pre-processing per l'Information Extraction

Per quanto riguarda il task di Information Extraction invece, il pre-processing assume un ruolo particolarmente importante, poiché i task di information extraction richiedono una gestione del testo differente rispetto alla text classification, con una maggiore attenzione alla struttura linguistica e alla preservazione delle entità informative. Per questo motivo, il pre-processing è stato suddiviso in due flussi distinti: uno specifico per la Keyphrase Extraction e uno dedicato alla Named Entity Recognition. Tale distinzione consente di applicare trasformazioni mirate in funzione degli obiettivi dei singoli task, evitando operazioni che potrebbero risultare utili per un task ma dannose per l'altro.

4.1.1 Pre-processing per il KPE

Figura 4.1: Pre-processing del KPE

Per il task di Keyphrase Extraction è stato adottato un pre-processing conservativo, finalizzato a ridurre rumore e artefatti tipici delle email senza alterare in modo significativo il contenuto informativo. L'obiettivo è preservare quanto più possibile la struttura linguistica e i termini potenzialmente rilevanti come keyphrase, evitando trasformazioni aggressive che potrebbero eliminare informazioni utili all'estrazione. In primo luogo, il testo è stato normalizzato gestendo le terminazioni di riga e rimuovendo eventuali caratteri di controllo, frequentemente presenti nei corpora derivati da email. Successivamente, sono stati eliminati eventuali tag HTML grezzi, mantenendo il testo leggibile e più omogeneo. Un passaggio centrale riguarda la gestione degli URL e degli indirizzi email. Tali elementi possono essere molto frequenti e altamente variabili; per questo motivo sono stati sostituiti con token standardizzati $<URL>$, $<EMAIL>$. In questo modo si preserva l'informazione della loro presenza, evitando che stringhe uniche e non generalizzabili interferiscano con l'estrazione delle keyphrase. Infine, è stata effettuata una pulizia leggera delle righe, rimuovendo linee vuote o costituite quasi esclusivamente da simboli ripetuti

(tipiche di separatori, quoting o accumuli di formattazione), senza però eliminare contenuto testuale significativo. Il testo è stato poi ricondotto a una forma lineare tramite normalizzazione degli spazi. Il risultato finale è stato salvato nella colonna body_kpe, utilizzata come input per le successive fasi del task KPE.

4.1.2 Pre-processing del NER

Figura 4.2: Pre-processing del NER

Per il task di Named Entity Recognition è stato adottato un pre-processing distinto rispetto a quello utilizzato per la Keyphrase Extraction, in quanto l'obiettivo principale del NER è l'individuazione e la classificazione di entità specifiche direttamente nel testo. Di conseguenza, il pre-processing è stato progettato per preservare integralmente le entità di interesse, evitando operazioni di mascheramento che potrebbero compromettere il riconoscimento corretto delle stesse. Analogamente al flusso KPE, il testo è stato inizialmente normalizzato gestendo le terminazioni di riga e rimuovendo caratteri di controllo e tag HTML grezzi, al fine di ridurre il rumore tipico delle email. A differenza del pre-processing per KPE, URL e indirizzi email non sono stati sostituiti con

token generici, ma mantenuti nel testo originale. Tuttavia, essi sono stati separati da spazi, in modo da facilitarne la successiva tokenizzazione e il corretto allineamento con le entità riconosciute dal modello NER.

Anche in questo caso è stata inoltre applicata una pulizia leggera delle righe, eliminando linee vuote o composte quasi esclusivamente da simboli ripetuti, che non apportano informazione semantica rilevante e possono interferire con il riconoscimento delle entità. Infine, il testo è stato ricondotto a una forma lineare mediante normalizzazione degli spazi.

Il risultato del pre-processing è stato memorizzato nella colonna body_ner, utilizzata come input per le successive fasi di Named Entity Recognition.

4.2 Introduzione

Una volta completate le operazioni di pre-processing specifiche, il progetto prosegue con la fase di **Information Extraction**, il cui obiettivo è estrarre automaticamente informazioni strutturate a partire dal testo non strutturato delle email. A differenza della text classification, che fornisce una valutazione globale del messaggio, l'information extraction consente di analizzarne il contenuto in modo più dettagliato, individuando elementi informativi rilevanti direttamente all'interno del testo.

Come è stato già detto in precedenza. in questo lavoro l'Information Extraction è articolata in due task distinti e complementari: **Keyphrase Extraction (KPE)**, finalizzata all'individuazione delle parole o espressioni più rappresentative del contenuto delle email, e **Named Entity Recognition (NER)**, volta al riconoscimento e alla classificazione di entità specifiche quali URL, indirizzi email e altri elementi di interesse nel contesto del phishing. Tale approccio consente di arricchire l'analisi testuale, fornendo una comprensione più approfondita delle caratteristiche informative e semantiche dei messaggi analizzati.

4.3 KPE

La fase di Keyphrase Extraction è stata realizzata applicando un metodo non supervisionato basato su *TextRank*, con l'obiettivo di estrarre, per ciascuna email, un insieme ristretto di parole o espressioni rappresentative del contenuto.

In primo luogo è stato introdotto un filtro sugli outlier di lunghezza, escludendo i testi con dimensione superiore a 50.000 caratteri. Questa scelta consente di evitare casi anomali (ad esempio email molto lunghe o contenenti porzioni di testo non informative) e, soprattutto, di mantenere sostenibili i tempi e i consumi di memoria durante l'analisi linguistica. Successivamente è stata costruita una pipeline NLP mediante *spaCy* (modello *en_core_web_sm*), utilizzata per trasformare i testi in oggetti linguistici (Doc) contenenti tokenizzazione e informazioni morfosintattiche. Su ciascun documento è stata quindi applicata l'estrazione delle keyphrase tramite *textacy.extract.keyterms.textrank*, impostando la normalizzazione a livello di lemma e selezionando le top-5 keyphrase per email. Infine, per garantire coerenza con il pre-processing, è stato applicato un filtro sulle keyphrase estratte, escludendo eventuali placeholder introdotti in precedenza (<URL>, <EMAIL>), che potrebbero comparire tra i termini più frequenti ma non rappresentano contenuto semantico utile ai fini dell'interpretazione.

	label	kpe_textrank_top5
0	1	[small dimension, Nazi tank, trouble, lover]
1	1	[sex, technique, pleasure]
2	1	[= =...
3	0	[= =...
4	1	[]
5	1	[C mo l cfd IC eym K, Yo wu urS mo, rc ebo eFo...
6	1	[fake swiss Men, Popular Panerai, Replica Watc...
7	1	[= =...
8	0	[assignee, otherbugsdependingo|, nthis|, |Remo...
9	1	[= =...

Figura 4.3: Visualizzazione delle prime 5 righe

La visualizzazione riporta, per un sottoinsieme di email, l'etichetta di classe e le keyphrase estratte tramite TextRank. L'analisi qualitativa degli esempi evidenzia come le keyphrase individuate riflettano in modo coerente il contenuto semantico dei messaggi.

Nel caso delle email etichettate come phishing (label = 1), le keyphrase estratte includono frequentemente termini riconducibili a contenuti persuasivi, commerciali o potenzialmente ingannevoli, come riferimenti a prodotti, marchi noti, offerte o tematiche sensibili. Tali elementi risultano tipici delle campagne di phishing, che mirano a catturare l'attenzione dell'utente sfruttando leve emotive o di interesse immediato.

Per le email legittime (label = 0), le keyphrase appaiono invece più descrittive e legate a contesti informativi o operativi, riflettendo un uso meno aggressivo del linguaggio e una maggiore neutralità semantica. In alcuni casi, la lista delle keyphrase risulta ridotta o poco informativa, a conferma del fatto che non tutti i messaggi contengono termini fortemente caratterizzanti.

Si osserva inoltre che, per alcuni messaggi, l'estrazione produce liste vuote o composte da token non significativi (ad esempio sequenze di simboli). Questo comportamento è riconducibile a email con contenuto estremamente rumoroso

o poco linguistico e rappresenta un limite noto dei metodi non supervisionati basati su grafi. Tuttavia, tali casi risultano minoritari e non compromettono la validità complessiva dell'analisi.

Successivamente, è stato introdotto un filtro di validazione sulle keyphrase estratte, con l'obiettivo di eliminare termini non informativi o rumorosi. In particolare, sono state scartate keyphrase contenenti placeholder, sequenze di simboli, stringhe troppo brevi o eccessivamente lunghe, e termini con una prevalenza elevata di caratteri non alfabetici. Una volta applicato il filtro poi è stata effettuata una visualizzazione qualitativa dei risultati della Keyphrase Extraction, campionando un numero limitato di email per ciascuna classe.

	label	body_kpe	kpe_textrank_top5
24834	1	>+=+...	[= =...
24191	1	Not all p pfx en hqf is en mh larg wf ing met ...	[lar ws ge p unm eni, lar cv ge onl, mb lar zt...
5144	1	Buy the Patek Philippe watch and know everythi...	[luxurious costume replica watch, Patek Philip...
72806	1	Your address is up-to-date ACCOUNT ACTIVITY: D...	[American Express ® account, American Express ...
60200	1	hello , young lovers ! :) sinalyomm ' e '	[young lover, sinalyomm]

	label	body_kpe	kpe_textrank_top5
37664	0	<URL> ------- Additional Comments From user7@ ...	[additional comment, assignee, user7@, 07:55]
36008	0	>> - for experienced users (Barry, skip, etc) ...	[visible ~/local symlink, dependent ~/local, S...
19101	0	Nick Coghlan wrote: > I believe the list of in...	[Dev mailing list, api design, Python version,...
50624	0	should i cancel my trip ? i don ' t mind at al...	[louise kitchen chief operating officer enron ...
61175	0	i think it is time that you have someone inven...	[critical one, business unit, project, daily o...

Figura 4.4: Visualizzazione di 5 righe per ogni classe

Come è stato già analizzato in precedenza, dall'analisi dei campioni emerge che, nelle email di phishing, le keyphrase sono spesso associate a marchi noti, prodotti, account o messaggi di allerta, mentre nelle email legittime risultano più frequenti termini legati a contesti informativi, tecnici o comunicazioni operative. Questa visualizzazione fornisce un riscontro qualitativo sull'efficacia del metodo di estrazione adottato e supporta l'interpretazione dei risultati ottenuti nella fase di KPE. Tuttavia in alcuni casi compaiono ancora sequenze di simboli tra le keyphrase estratte. Ciò è dovuto al carattere non supervisionato di TextRank e alla presenza di testi particolarmente rumorosi, in cui tali sequenze possono risultare centrali nel grafo di co-occorrenza. Il filtro applicato, volutamente non troppo aggressivo, riduce la maggior parte dei casi ma non

elimina completamente tutte le occorrenze residue, che risultano comunque limitate e non influenti sull'analisi complessiva.

	keyphrase	log_odds_phish_vs_legit	count_phish	count_legit
111577	cable news network lp	7.988061	2930	0
131538	+ = +	7.988061	2930	0
97265	custom alert alert	7.254177	1406	0
22644	personal life	6.935457	1022	0
143050	republican presidential candidate john mccain	6.900645	987	0
124287	n. dakota oil boom	6.900645	987	0
64308	plane evacuates passenger	6.894554	981	0
119709	smoke forces emergency landing	6.894554	981	0
44413	anthrax suspect apparent suicide	6.875016	962	0
46575	defendant fakes heart attack	6.875016	962	0
65697	= ...	6.196302	2930	5
91950	replica classics	6.131832	457	0
9137	short week	6.049968	421	0
23051	inch	5.915759	368	0
154317	patek philippe	5.614434	272	0
49361	revolutionary e	5.561790	258	0
97708	foreign account	5.489759	240	0
152536	right time	5.460283	233	0
64093	pill	5.456001	232	0
155160	expense drugstore	5.451700	231	0
103708	herbal breakthrough	5.434308	227	0
103733	free bottle today	5.434308	227	0
5972	short month	5.416608	223	0
63557	large manhood	5.412134	222	0

Figura 4.5: Visualizzazione delle keyphrase maggiormente presenti nelle email di phishing

Come analisi conclusiva della Keyphrase Extraction, è stata condotta una valutazione comparativa per classe delle keyphrase estratte, stimandone il potere discriminante tra email di phishing e email legittime. Per ciascuna keyphrase sono stati calcolati: la frequenza nella classe phishing, la frequenza nella classe legittima e il log-odds ratio (phish vs legit), il quale serve a misurare quanto una keyphrase sia caratteristica del phishing rispetto alle email legittime (valore del ratio positivo). I risultati mostrano che alcune keyphrase presentano un elevato log-odds a favore del phishing, comparendo quasi esclusivamente in email malevole e risultando assenti o rare nelle email legittime. Tali termini sono spesso riconducibili a contenuti sensazionalistici, commerciali o di allerta, coerenti con le strategie linguistiche tipiche del phishing. Al contrario, le keyphrase più bilanciate o associate alle email legittime risultano meno marcate dal punto di vista discriminativo.

4.4 NER

Per il task di **Named Entity Recognition** è stata adottata una strategia ibrida, combinando un modello NER generale con un riconoscimento mirato di entità tipiche del dominio email/phishing.

Come per il KPE, in primo luogo è stato applicato un filtro sugli outlier di lunghezza, escludendo i testi con più di 50.000 caratteri, al fine di evitare casi anomali e mantenere sostenibile l'elaborazione. Successivamente, è stato utilizzato il modello *spaCy en_core_web_sm* per estrarre le entità nominate presenti nel testo , restituendo per ciascuna entità la stringa testuale e la relativa label.

Parallelamente, poiché URL e indirizzi email rappresentano entità di particolare interesse nel contesto del phishing e non sempre vengono riconosciuti in modo affidabile dai modelli NER generici, è stato introdotto un secondo estrattore basato su *regex* dedicato specificamente a URL ed EMAIL.

Infine, le entità estratte dai due metodi sono state aggregate in un'unica lista, ottenendo per ciascuna email un insieme complessivo di entità etichettate, utilizzato nelle successive analisi e visualizzazioni.

	label	ner_all
0	1	[(Buck, PERSON), (Nazi, NORP), (http://whitedo...
1	1	[(http://www.brightmade.com, URL)]
2	1	[(DAILY, ORG), (10, CARDINAL), (1, 2008, DATE)...
3	0	[(TLD, ORG), (http://en.wikipedia.org/wiki/.so...
4	1	[(WelcomeFastShippingCustomerSupport, GPE), (h...
5	1	[(Yo wu urS mo, PERSON)]
6	1	[(Swiss Men's, ORG), (Ladie, PERSON), (Replica...
7	1	[(DAILY, ORG), (10, CARDINAL), (1, 2008, DATE)...
8	0	[(http://issues.apache.org/SpamAssassin/show_b...
9	1	[(DAILY, ORG), (10, CARDINAL), (1, 2008, DATE)...

Figura 4.6: Visualizzazione delle entità riconosciute

La visualizzazione mostra, per un sottoinsieme di email, l'etichetta di classe e l'elenco delle entità riconosciute, ottenute combinando l'output del modello NER di spaCy con l'estrazione mirata di URL ed email tramite regex.

Dagli esempi emerge che il sistema è in grado di individuare correttamente diverse tipologie di entità, tra cui PERSON, ORG, GPE, DATE, CARDINAL, oltre a URL ed EMAIL, particolarmente rilevanti nel contesto del phishing. Nelle email etichettate come phishing compaiono frequentemente entità riconducibili a organizzazioni, persone o riferimenti temporali utilizzati per rendere il messaggio più credibile, mentre nelle email legittime le entità risultano spesso legate a contesti informativi o tecnici.

Successivamente è stata applicata una fase di normalizzazione delle entità estratte, con l'obiettivo di rendere l'output del NER più coerente e confrontabile. In particolare, le entità sono state ripulite da spazi superflui e deduplicate a livello di singola email, ovvero si fa in modo che all'interno di uno stesso messaggio, una determinata entità viene conteggiata una sola volta, anche se compare più volte nel testo. Questa operazione consente di evitare ridondanze dovute a ripetizioni della stessa entità all'interno del messaggio e di facilitare le successive analisi statistiche e di visualizzazione.

	PERSON	NORP	URL	ORG	CARDINAL	DATE	TIME	LOC	ORDINAL	GPE	EMAIL	MONEY	EVENT	FAC	QUANTITY	PERCENT	PRODUCT	WORK_OF_ART	LAW	LANGUAGE
label																				
0	4.508438	0.351036	1.398148	3.648154	4.027503	3.297978	0.749842	0.111226	0.241556	1.160413	0.036485	0.500367	0.048782	0.075779	0.120208	0.150773	0.172026	0.177593	0.038206	0.036510
1	1.932575	0.269235	2.043007	2.368286	2.539730	1.297945	0.246356	0.102922	0.136064	1.019171	0.074236	0.651212	0.071530	0.030109	0.094829	0.333652	0.068055	0.043193	0.039695	0.005457

Figura 4.7: Analisi quantitativa delle entità NER per classe

Dopo la fase di normalizzazione è stata condotta un'analisi quantitativa delle entità riconosciute, calcolando la media del numero di entità per email separatamente per ciascuna classe e per ciascun tipo di entità NER.

I risultati mostrano differenze significative nella distribuzione delle entità. In particolare, le email legittime presentano mediamente un numero più elevato di entità di tipo PERSON, ORG, CARDINAL e DATE, tipicamente associate

a comunicazioni informative, tecniche o contestualizzate. Al contrario, le email di phishing mostrano una maggiore presenza relativa di URL ed EMAIL, entità strettamente collegate a tentativi di reindirizzamento e raccolta di informazioni sensibili.

Questa analisi conferma che, una volta normalizzate e deduplicate, le entità estratte risultano comparabili a livello statistico e consentono di evidenziare pattern distintivi tra le due classi, difficilmente osservabili tramite una semplice ispezione qualitativa dei singoli esempi.

	entity	entity_type	count
0	one	CARDINAL	7824
1	2	CARDINAL	6062
2	1	CARDINAL	5802
3	3	CARDINAL	5557
4	8	CARDINAL	5167
5	5	CARDINAL	5084
6	4	CARDINAL	4641
7	cnn	ORG	4543
8	georgia	GPE	4420
9	atlanta	GPE	4393
10	center	GPE	4336
11	cnn.com	ORG	4336
12	6	CARDINAL	4258
13	10	CARDINAL	4026
14	9	CARDINAL	3592
15	today	DATE	3258
16	7	CARDINAL	3222
17	daily	ORG	2936
18	1, 2008	DATE	2930
19	3:58 pm edt >+	TIME	2930
20	the most trusted name in news > cable news net...	ORG	2930
21	lllp	PERSON	2930
22	cable news network lp	ORG	2930
23	http://www.cnn.com/feedback	URL	2930

Figura 4.8: Entità NER più frequenti nelle email di phishing

La tabella riporta le entità più frequenti estratte dalle email di phishing, insieme al relativo tipo di entità e al numero totale di occorrenze nel corpus. Questa analisi consente di individuare quali elementi informativi risultano

maggiormente ricorrenti e caratterizzanti nei messaggi malevoli. Dai risultati emerge una forte presenza di entità di tipo CARDINAL e DATE, spesso utilizzate per conferire urgenza o specificità temporale ai messaggi (ad esempio numeri, scadenze, riferimenti a date). Sono inoltre frequenti entità di tipo ORG e GPE, che includono nomi di organizzazioni, domini o luoghi noti, sfruttati per aumentare la credibilità del messaggio. La presenza di URL tra le entità più frequenti conferma il ruolo centrale dei collegamenti ipertestuali nelle campagne di phishing, finalizzate al reindirizzamento dell'utente verso risorse esterne. Nel complesso, questa visualizzazione fornisce una sintesi efficace delle strategie linguistiche e informative adottate nelle email di phishing e dimostra come il NER possa contribuire a individuare pattern ricorrenti e indicatori semantici rilevanti, integrando l'analisi svolta nella fase di classificazione testuale.