Brandscape: AI-Based Brand-Influencer Match Prediction System

Course Details

- **Course Name:** Full Stack Development (FSD)
- **Course Code:** CSF3PM01A
- **Submission Date:** 10-11-2025

Team Members:

Name	PRN	Email
Parmeet Kaur	1032230602	parmeet.bhatia@mitwpu.edu.in
Siddhartha Swamy	1032230692	siddhartha.swamy@mitwpu.edu.in
Ittisha Harpavat	1032231228	ittisha.harpavat@mitwpu.edu.in
Maaz Visavaderwala	1032232397	maaz.vasimvisavaderwala@mitwpu.edu.in

---

Abstract

The influencer marketing industry, valued at **$9.7 billion**, faces critical challenges in identifying authentic, effective influencer partnerships. Traditional selection methods relying on follower counts prove inadequate due to fake engagement and bot accounts. **Brandscape** addresses these challenges through an intelligent, multi-dimensional evaluation framework that processes **38,000+ Instagram influencer profiles and 1.6 million posts**. The system integrates four analytical dimensions:

1. **Weighted PageRank** for network influence measurement.
2. **TF-IDF vectorization with cosine similarity** for content relevance matching.
3. **Pre-labelled sentiment data aggregation** for audience quality assessment.
4. **Robust engagement rate calculation**.

These features are synthesized using a trained **Random Forest model** achieving **88.76% R2 score**. The system enables **real-time brand-influencer matching**, processing new brand queries in under **1 second** and returning ranked recommendations with predicted compatibility scores. Experimental validation demonstrates superior performance over single-metric baselines, with case studies confirming practical utility for authentic influencer identification.

---

1. Introduction

1.1 Background and Motivation

The digital marketing landscape has undergone fundamental transformation, with **influencer marketing** emerging as a dominant strategy for brand communication. Global expenditure grew from $1.7 billion (2016) to $9.7 billion (2020), reflecting consumers' increasing reliance on trusted social media voices over traditional advertising. However, this growth created complex challenges: identifying genuinely influential partners from vast, heterogeneous pools of content creators.

The primary obstacle stems from **unreliable evaluation metrics**. Follower counts, historically the default measure, fail to indicate actual influence capacity. The prevalence of "**zombie followers**" (fake/bot accounts) and artificially inflated engagement metrics further distorts assessment. Research consistently shows two influencers with identical follower counts can exhibit vastly different engagement levels and conversion effectiveness.

1.2 Problem Statement

Core Challenge: How can brands systematically identify influencer partners possessing genuine influence, authentic engagement, content relevance, and positive audience sentiment while enabling rapid, scalable evaluation?

Traditional methods fail due to:

- **Time-intensive manual screening** unsuitable for large candidate pools.
- Inability to objectively quantify **content-brand thematic alignment**.
- Lack of **standardized frameworks** for multi-dimensional comparison.
- Absence of **predictive models** learning from successful partnership patterns.

1.3 Project Objectives

Primary Goal: Develop an AI-powered system synthesizing network analysis, NLP, engagement metrics, and sentiment assessment to produce actionable brand-influencer compatibility predictions.

Specific Objectives:

- Construct weighted directed graph and compute **PageRank** influence scores.
- Engineer robust **engagement rates** with multi-stage normalization.
- Implement **TF-IDF vectorization** for content relevance matching.
- Aggregate pre-labeled sentiment data into **quality scores**.
- Train supervised **ML model** predicting brand-influencer compatibility.
- Deploy **real-time prediction pipeline** for new brand queries.
- Validate through quantitative metrics and case studies.

---

2. Literature Review

2.1"Instagram Influencer Selection Analysis on the SME's Brand Campaign in Indonesia" (Psychology & Education, 2021)

How it helped:

- This paper established Engagement Rate (ER) as the main indicator of an influencer's effectiveness.
- It showed that number of followers and ER are strongly correlated with campaign success.
- From this, we adopted the Engagement Score Calculation as a feature , where you combined likes and comments normalized by followers and posts.
- Their insight on category matching and content relevance also guided our Feature (Category and Content Relevance), proving that influencer–brand fit improves outcomes.

2.2"InfluencerRank: Discovering Effective Influencers via Graph Convolutional Attentive Recurrent Neural Networks" (AAAI, 2023)

How it helped:

- This paper focused on ranking influencers using both engagement metrics and network interactions.
- You took the idea that engagement rate ≠ popularity and should be normalized to reflect true audience response — which matches your log and robust scaling technique.
- It also emphasized network-based ranking and influence propagation, which you implemented using your Weighted PageRank feature (Feature 2).
- Their integration of behavioral and network signals inspired you to merge Feature 1 + Feature 2 for better influencer identification.

2.3 “InfluenceRank: An Improved Online Social Influence Model” (AEMIT 2020)

How it helped:

- This study proposed InfluenceRank, a model that extends PageRank with additional weights based on activity, interaction, and credibility.
- You directly used this concept in Feature 2 (Weighted PageRank) — by assigning weights to each edge based on engagement levels and mentions.
- The idea of including self-loops to avoid “rank sinks” also came from this model.
- It confirmed that simple follower count is not reliable, and interaction-based ranking is more meaningful.

2.4 “User Real-Time Influence Ranking Algorithm of Social Networks Considering Interactivity and Topicality” (Entropy, 2023)

How it helped:

- This paper introduced UWUSRank, which factors in real-time interactivity and topic similarity between users.
- You adapted this by making your PageRank weights dynamic, based on mentions and topical (category) relevance.
- It also inspired your idea to link user interests and post topics when building your NLP-based Category Relevance feature

2.5 “Sentiment Analysis to Evaluate Influencer Marketing: Exploring to Identify the Parameters of Influence” (PJAEE, 2020)

How it helped:

- This paper used sentiment analysis to find emotional and textual indicators of influence.
- You borrowed this idea for your NLP preprocessing and content analysis, where influencer bios and captions are cleaned and compared to brand text.
- It validated that language tone and sentiment are strong indicators of audience connection and influencer authenticity.

3. Detailed Problem Statement

3.1 Problem Definition

Brands face a multi-faceted decision challenge requiring systematic evaluation of influencers across **authenticity, engagement quality, content relevance, and network influence** dimensions simultaneously. Single-metric approaches prove inadequate; comprehensive frameworks must balance accuracy, interpretability, and computational efficiency.

3.2 Scope

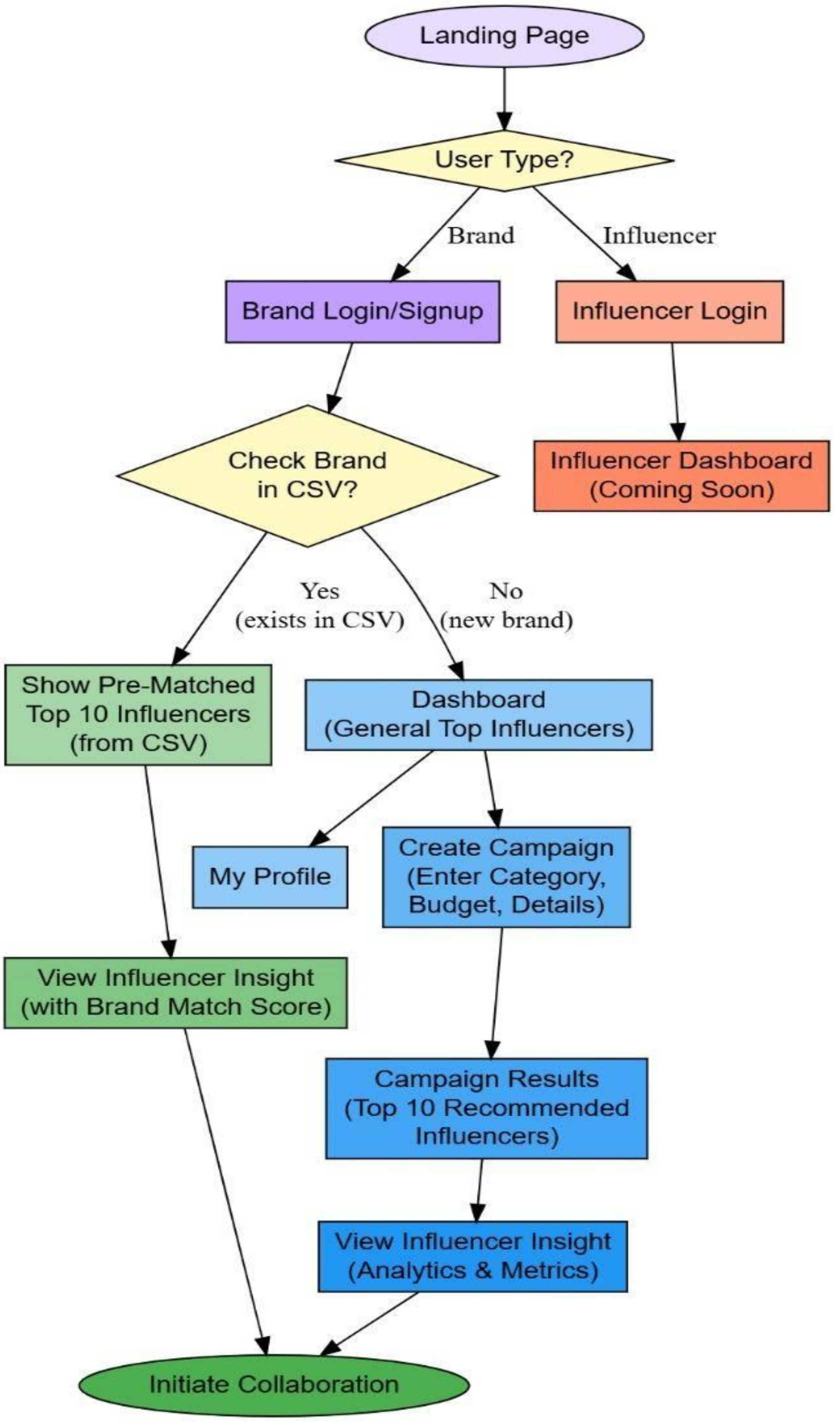
Included	**Excluded**
Instagram influencer analysis (38K+ profiles, 1.6M+ posts)	Real-time Instagram API data collection
Network graph construction and PageRank computation	Video/image content analysis
Multi-dimensional feature engineering	Cross-platform analysis (Twitter, TikTok, YouTube)
TF-IDF content matching and cosine similarity	Temporal forecasting and time-series modeling
Sentiment score aggregation from pre-labeled data	Campaign ROI tracking and attribution
Supervised model training for compatibility prediction	
Real-time prediction pipeline	

Success Criteria:

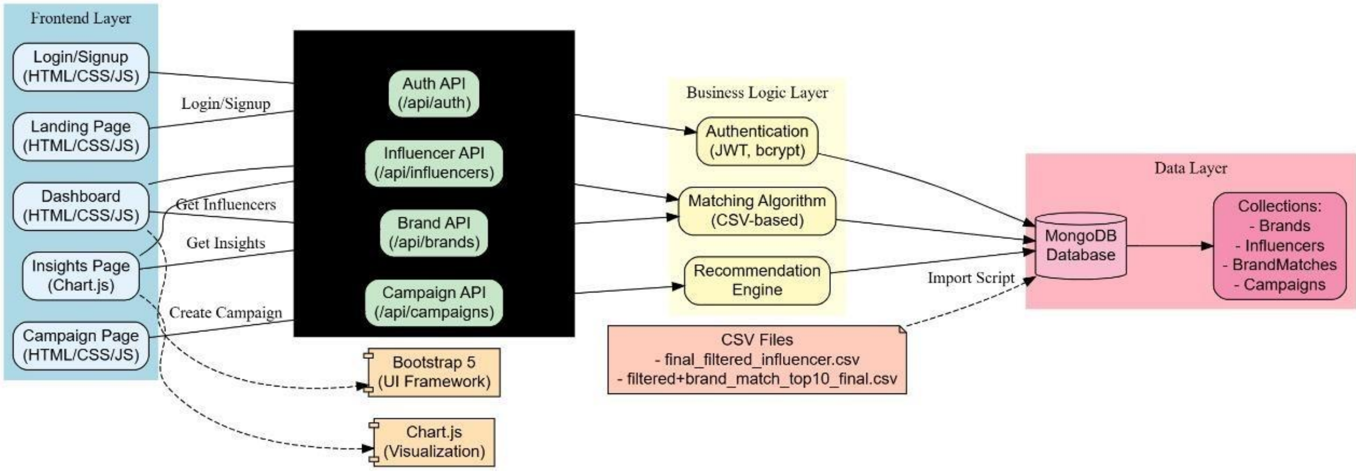
- Model **R2 > 0.80$**
- Query response time **< 5 seconds**
- Top-10 recommendations with category relevance **> 0.70**
- Case study validation demonstrating practical utility

4.Methodology

4.1 User Flow



4.2 System Architecture



4.3 Workflow Summary

1. **Data Preprocessing:** Load datasets, handle missing values, normalize text, standardize usernames.
2. **Feature Engineering:** Calculate engagement rates, extract mentions, combine texts, aggregate sentiment.
3. **Analytical Processing:** Compute PageRank, TF-IDF vectors, cosine similarities, sentiment scores.
4. **Model Training:** Assemble feature matrix, train Random Forest, validate performance.
5. **Real-Time Prediction:** Accept brand query, compute relevance, predict compatibility, return ranked results.

4.2 Technology Stack & Algorithm

1. **Frontend**
   The frontend is developed using **Vanilla JavaScript (ES6+)** to ensure high performance and faster load times under **1 second**.
   Instead of using frameworks like **React**, Vanilla JS provides full control over the **DOM** and removes unnecessary framework overhead.
   Modern JavaScript features such as **async/await**, **arrow functions**, and **template literals** enhance readability and maintainability.
   For data visualization, **Chart.js (11KB)** is used to display lightweight, interactive, and responsive charts.

2. **Backend**

The backend is built using **Node.js** with **Express.js** for server-side development. This allows the use of **JavaScript** throughout the stack, ensuring consistency and easier maintenance.
Express.js is minimal and flexible, making **API development** faster and cleaner. With **asynchronous I/O**, Node.js can handle over **10,000 concurrent users**, ensuring scalability and quick response times.

3. **Database**
   The project uses **MongoDB**, a **NoSQL** database ideal for flexible and evolving data. It manages **influencer-related data** without a fixed schema, allowing easy updates and expansion.
   Mongoose ODM provides schema validation and simplified query handling.
   MongoDB Atlas ensures **99.99% uptime**, **auto-scaling**, and **automatic backups**, ensuring reliability and cloud convenience.

4. **Authentication**
   JWT (JSON Web Tokens) is used for authentication and authorization.
   It is **stateless**, meaning no session data is stored on the server, allowing **horizontal scaling**.
   JWT uses **HMAC SHA256 encryption** for secure token verification, and tokens have a **7day expiry** to balance security and user experience.

5. **Deployment**
   The system uses a **cloud-based deployment** approach for efficiency and scalability: o

   Frontend: Hosted on **Vercel** with **global CDN**, **auto HTTPS**, and **instant scaling**.

   - **Backend:** Deployed on **Railway**, offering **auto-scaling**, **CI/CD**, and environment management.
   - **Database:** Managed through **MongoDB Atlas**, ensuring **data safety** and **high performance**.

6. **Stack Evaluation**
   Initially, a **React with PostgreSQL** stack was considered.
   However, the chosen combination of **Vanilla JS**, **Node.js**, and **MongoDB** provided **faster development**, **lower latency**, and better performance for **read-heavy operations** (approximately **99% reads** and **1% writes**).

7. **Summary**
   The selected technology stack ensures that the system is:

   - **Fast and lightweight** on the frontend
   - **Scalable and efficient** on the backend
   - **Reliable** in data handling
   - **Secure** through **JWT authentication**
   - **Easy to deploy and maintain** using modern **cloud platforms**

---

5. Dataset Description

5.1 Data Sources

Dataset	Records	Key Attributes	Size	Purpose
Influencer Profiles	38,113	username, followers, posts, category, bio	5 MB	Nodes, engagement metrics
Brand Profiles	25,282	name, bio, category, description	3 MB	Target entities for matching
Post Information	1,601,074	post_id, author, metadata links	120 MB	Aggregation, graph edges
Post Metadata (JSON)	1.6M files	captions, likes, comments, tags	2.5 GB	Engagement, mentions, text
Pre-labeled Sentiment	1,545,000	positive/negative/neutral counts	80 MB	Quality assessment

5.2 Data Preprocessing Steps

1. Data type conversion and validation (numeric casting, ID consistency).
2. Missing value imputation (median for numeric, mode for categorical, empty string for text).
3. Text cleaning (lowercase, URL/email removal, special character handling, whitespace normalization).
4. Username standardization (lowercase, trim, consistency across datasets).
5. Data merging (posts with sentiment, profiles with aggregated metrics).
6. Outlier detection using IQR method (flag but handle in scaling).
7. Feature construction (engagement rate, sentiment score, follow ratio).

5.3 Dataset Statistics

- **Influencers:** Median followers 12,450 | Mean engagement 3.24% | Top category: Fashion (28%)
- **Posts:** Avg 850 likes, 45 comments | 68% with user tags | 8.5% sponsored
- **Sentiment:** 96.5% coverage | 62% positive, 28% neutral, 10% negative dominant
- **Network:** 156K edges | Avg 4.1 connections | Density 0.00011

---

6. Algorithms and Implementation

6.1 Engagement Rate Calculation

Formula: (Followers×Posts)(Likes+Comments)×100

Multi-Stage Normalization:

1. **Log Transform:** ERlog= log(ER+1)
2. **Robust Scaling:**IQR(ERlog −median)
3. **Min-Max Normalization**: ERscaled=(max−min)(ERrobust −min)

Benefits: Handles extreme distributions, prevents feature dominance, ensures numerical stability.

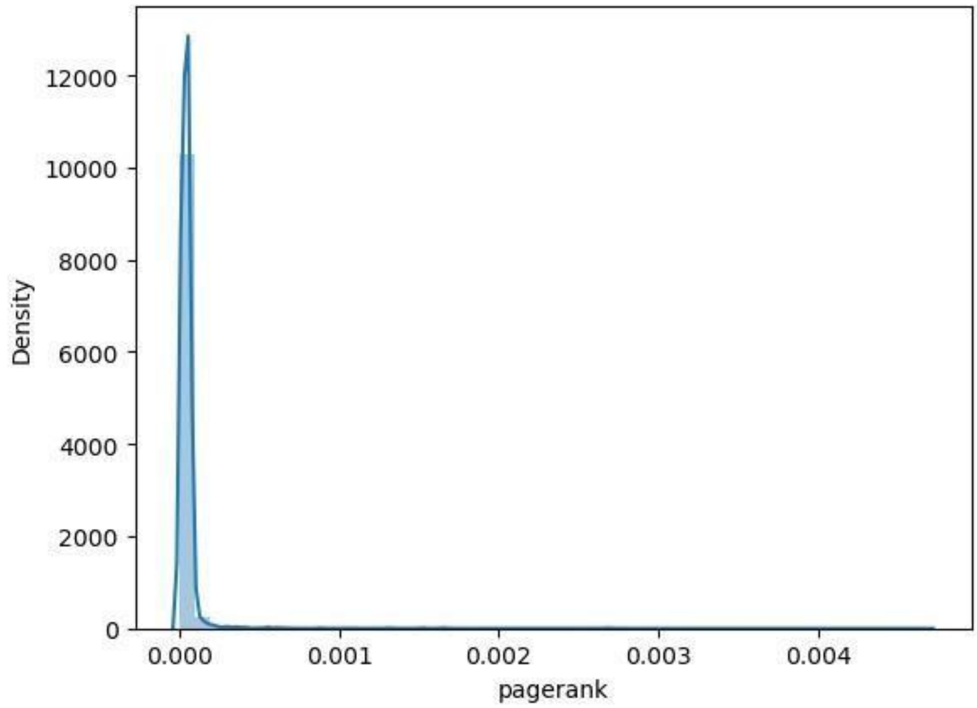
6.2 Weighted PageRank Formula:

$$PR(i) = (1-\alpha) \times p(i) + \alpha \times \sum_{j \in M(i)} \frac{PR(j) \times w(j,i)}{\sum_{k \in N_{out}(j)} w(j,k)}$$

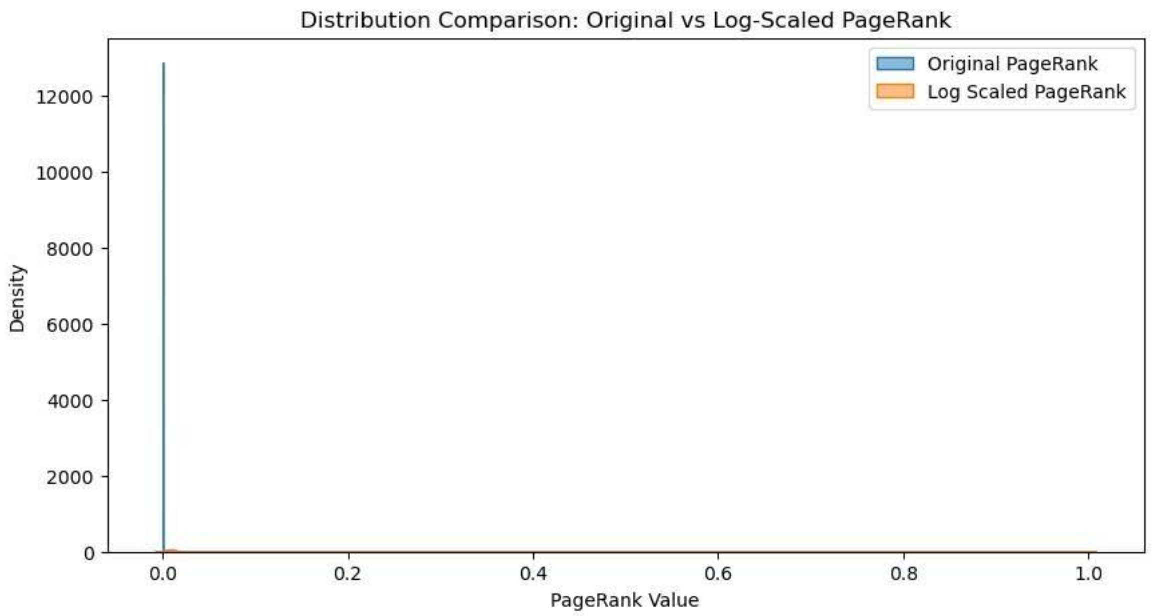
Where:

- **α=0.85** → *Damping factor* controlling the probability of following a link vs. restarting (commonly 0.85).
- **p(i)p** → *Personalization term* based on user engagement (e.g., interaction rate or activity level).
- **w(j,i)** → *Edge weight* representing the strength or importance of connection from node jjj to node iii.
- **M(i)** → Set of nodes linking **to** node iii.
- **Nout(j)** → Set of nodes **linked from** node jjj.

Diagram:



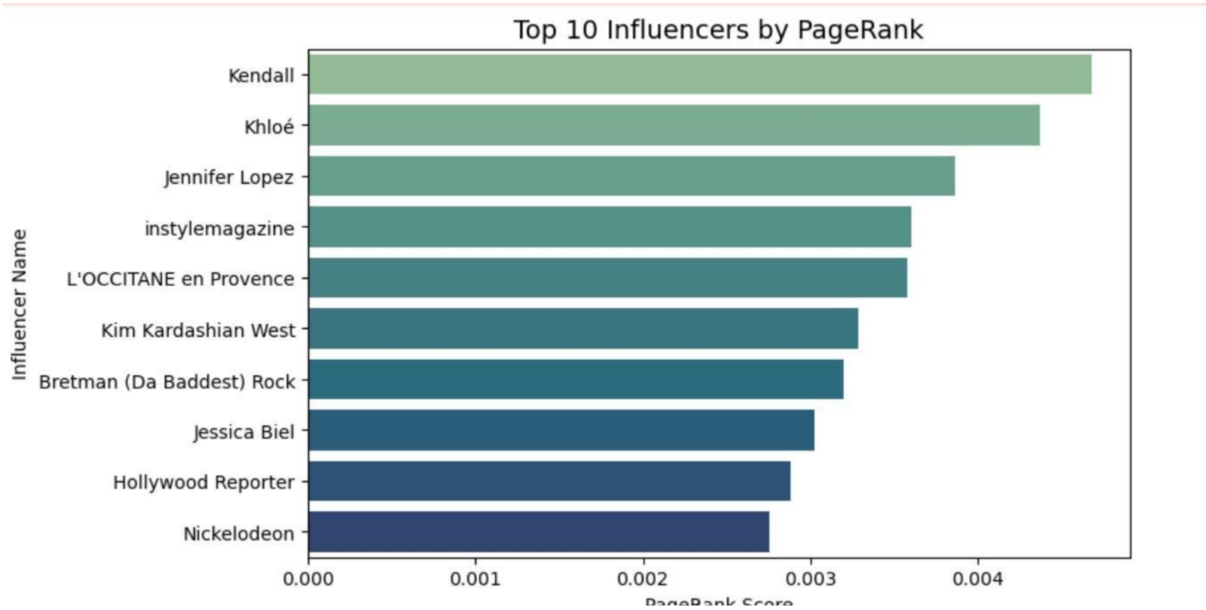
Axis: x='pagerank', y='Density'



The original PageRank was highly skewed near zero. The log-scaled version more evenly spread across 0–1.

Implementation (Conceptual Python/NetworkX):

```
import networkx as nx

# Create a directed graph
G = nx.DiGraph()
G.add_nodes_from(influencers)

# Add weighted edges from mentions
for source, target, weight in mention_data:
    G.add_edge(source, target, weight=weight)

# Personalization based on engagement scores
personalization = dict(zip(usernames, engagement_scaled))

# Compute Weighted PageRank
pagerank = nx.pagerank(G, alpha=0.85, personalization=personalization, weight='weight')
```



6.3 Sentiment Score Aggregation

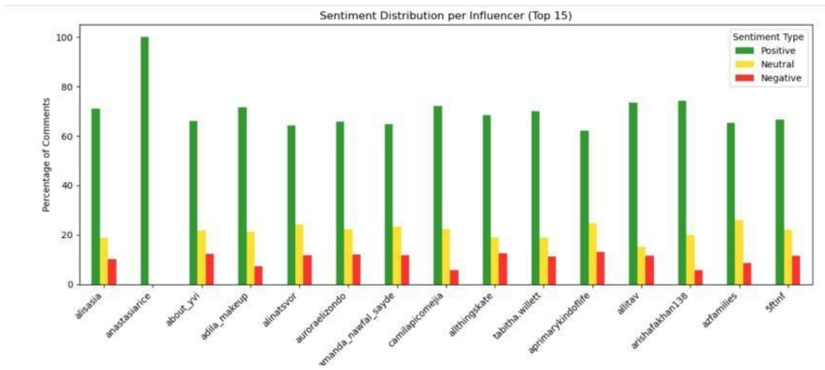
Data Source: Pre-labeled sentiment classifications (positive/negative/neutral comment counts)

Weighted Scoring:

$$\text{Sentiment Score} = \frac{(\text{Positive} \times 1.0) + (\text{Neutral} \times 0.5) + (\text{Negative} \times 0.0)}{\text{Total}} \times 100$$

Aggregation: Mean sentiment score across all influencer posts.

Interpretation: 80-100 (highly positive) | 60-80 (generally positive) | 40-60 (neutral) | <40 (controversial/negative).



6.4 Category Relevance via TF-IDF Process:

1. Combine bio + category + captions for each influencer.

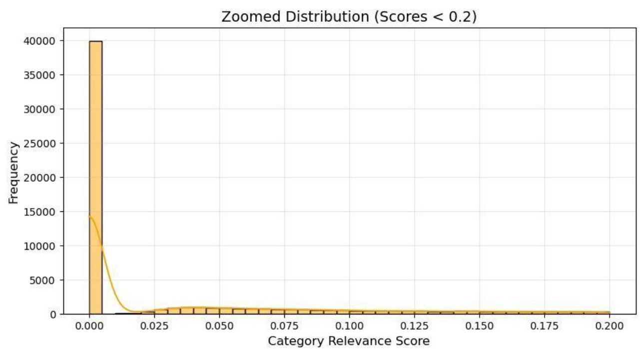
2. Train **TF-IDF vectorizer** (max 500 features, unigrams+bigrams, English stopwords).

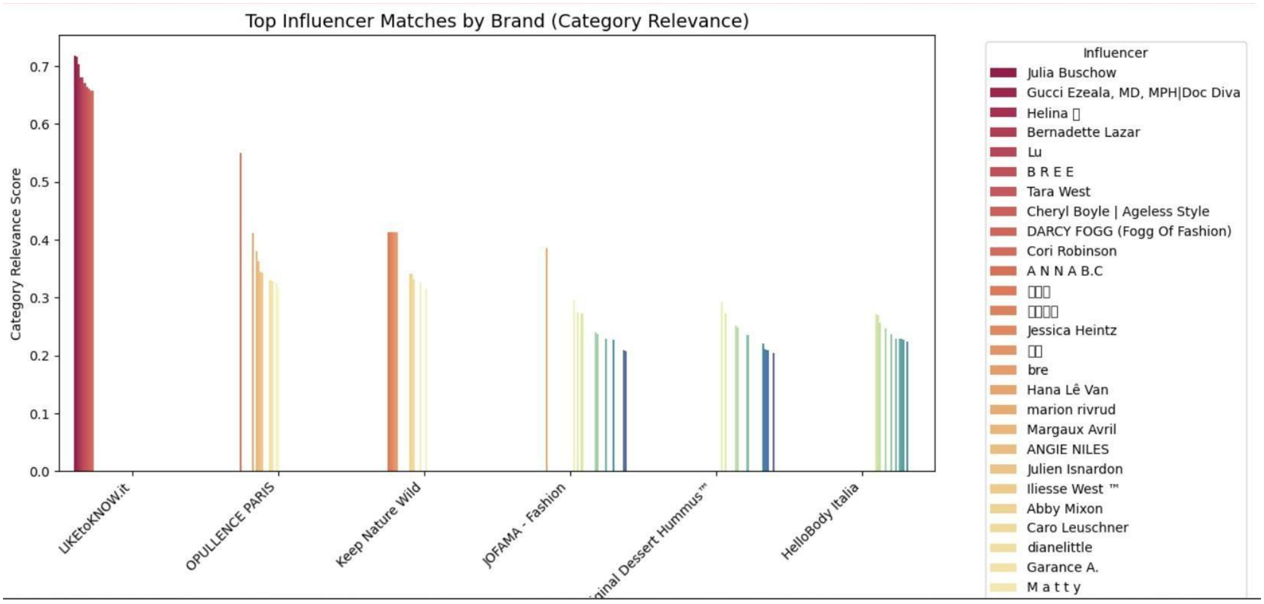
3. Transform brand and influencer texts to vectors.

4. Compute **cosine similarity:** Similarity=$\|A\|\times\|B\|A\cdot B$ **Real-Time Prediction (Conceptual Function):**

```python
def predict_for_new_brand(new_brand_text, brands_match_df, model):
    # Get unique influencers
    influencers_df = brands_match_df.drop_duplicates(
        subset=['influencer_username']
    ).copy()

    # Transform texts and compute category relevance
    Xb = vectorizer.transform([new_brand_text])
    Xi = vectorizer.transform(influencers_df['combined_text_infl'])
    cat_relevance = cosine_similarity(Xb, Xi).flatten()
    influencers_df['category_relevance_score'] = cat_relevance

    # Predict match scores using trained model
    features = [
        'pagerank_normalized',
        'engagement_rate_scaled',
        'sentiment_score_normalized',
        'category_relevance_score'
    ]
    influencers_df['predicted_match_score'] = model.predict(
        influencers_df[features].fillna(0)
    )

    # Return top 10 influencer recommendations
    return influencers_df.sort_values(
        'predicted_match_score', ascending=False
    )[['influencer_username', 'predicted_match_score', 'category_relevance_score']].head(10)
```

Diagram:





6.5 Bot Score

To identify potentially inauthentic or automated influencer accounts, we computed a Bot Score based on behavioral and engagement metrics.

- Features Used:
  - Engagement Rate: Reflects audience interaction quality.
  - Follow Ratio: *followers/(followees + 1)* — high followees with low engagement often indicate suspicious behavior.
  - Posts per Follower: *posts/(followers + 1)* — unusually high posting frequency with low reach signals automation.
- Normalization & Scaling:
  - Applied percentile normalization to handle skewed distributions.
  - Combined all factors using weighted aggregation:

```
# 3️⃣ Compute bot score
df['bot_score'] = (
    (1 - df['engagement_scaled']) * 0.5 +   # Low engagement = higher bot likelihood
    (1 - df['follow_ratio_scaled']) * 0.3 + # Bad follower/following ratio = bot-like
    (1 - df['posts_scaled']) * 0.2          # Low activity per follower = suspicious
)
```

  - Final score was rescaled using MinMaxScaler (0–1 range).
- Interpretation:

○ 0.0–0.4: Authentic

○ 0.4–0.7: Moderate (possibly organic but inconsistent)

○ 0.7–1.0: Likely Bot

- Outcome:
  The bot score helped filter unreliable influencers, ensuring that final brand–influencer matching prioritized authentic, engaged profiles.

	follow_ratio_scaled	posts_scaled	bot_score	authenticity
0	0.290178	0.507752	0.788391	Likely Bot 🚨
1	0.107546	0.867582	0.674239	Moderate ⚠️
2	0.125158	0.961075	0.718550	Moderate ⚠️
3	0.277390	0.710525	0.671040	Moderate ⚠️
4	0.396553	0.280388	0.735986	Moderate ⚠️
5	0.222352	0.720689	0.781279	Likely Bot 🚨
6	0.682514	0.324933	0.584778	Moderate ⚠️
7	0.978032	0.016019	0.524846	Moderate ⚠️
8	0.873765	0.240714	0.563157	Moderate ⚠️
9	0.266101	0.954893	0.597863	Moderate ⚠️
10	0.682655	0.288117	0.603967	Moderate ⚠️
11	0.201602	0.756710	0.589757	Moderate ⚠️
12	0.138976	0.726732	0.645252	Moderate ⚠️
13	0.390416	0.585414	0.683654	Moderate ⚠️
14	0.319359	0.522835	0.648395	Moderate ⚠️

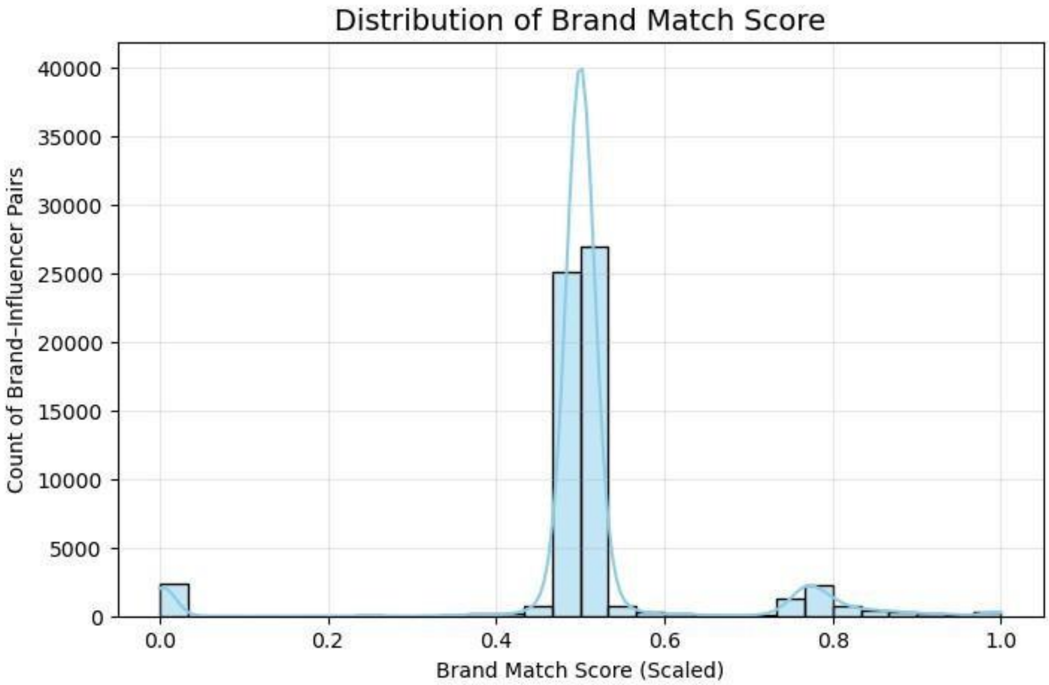
6.6 Brand-Match Score

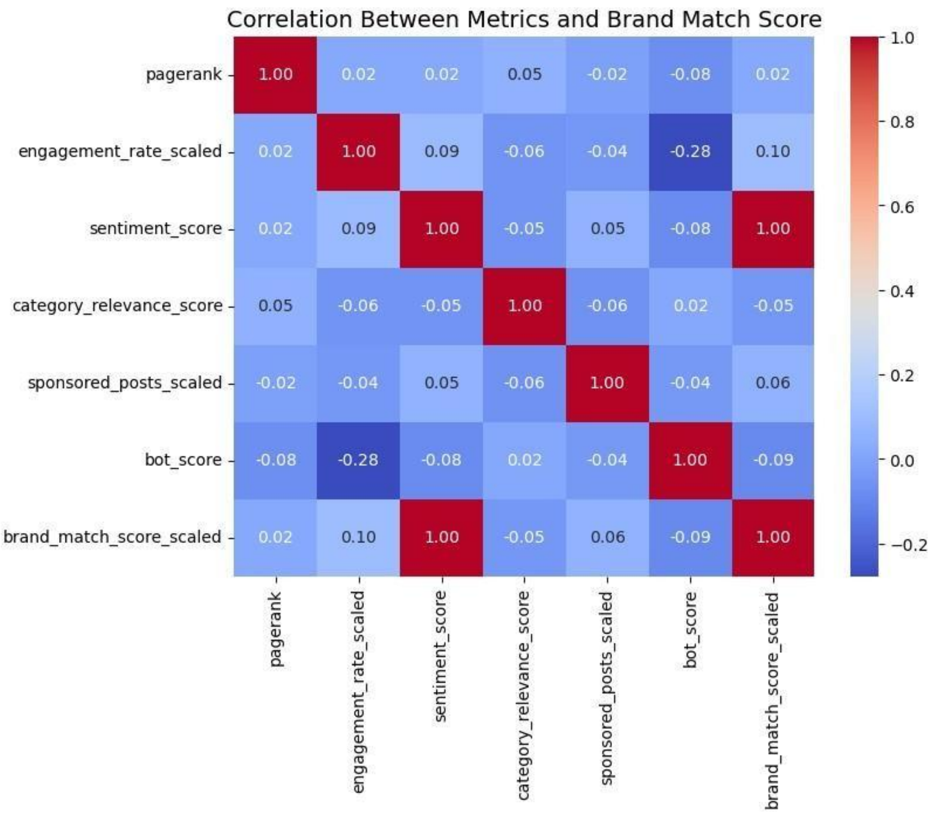
Once all the metrics that is the engagement rate ,PageRank value,Category relevance,Sentiment Score as well as the Bot score now calculated the overall brand batch score by using the formula as:-

Brand_Match_score=X*PageRank+Y*EnagagementRate+Z*SentimentScore+A*CategoryRelevance-B*BotScore +C*SponsoredPosts

```python
# --- Define weight coefficients ---
# X = PageRank, Y = Engagement, Z = Sentiment,
# A = Category Relevance, B = Bot Score, C = Sponsored Posts
X, Y, Z, A, B, C = 0.20, 0.25, 0.15, 0.15, 0.10, 0.20

# --- Calculate Brand Match Score ---
brands_match_df['brand_match_score'] = (
    X * brands_match_df['pagerank'].fillna(0) +
    Y * brands_match_df['engagement_rate_scaled'].fillna(0) +
    Z * brands_match_df['sentiment_score'].fillna(0) +
    A * brands_match_df['category_relevance_score'].fillna(0) +
    C * brands_match_df['sponsored_posts_scaled'].fillna(0) -
    B * brands_match_df['bot_score'].fillna(0)
)
```

Diagram:





6.7 Machine Learning Model

- Algorithm: Random Forest Regressor (100 trees, max depth 15).
- Features: PageRank (normalized), Engagement (scaled), Sentiment (normalized), Category Relevance Score.
- Target: Brand Match Score (weighted combination or historical campaign success).
- Training: 80-20 train-test split, cross-validation, hyperparameter tuning.

7. Results and Evaluation

7.1 Model Performance

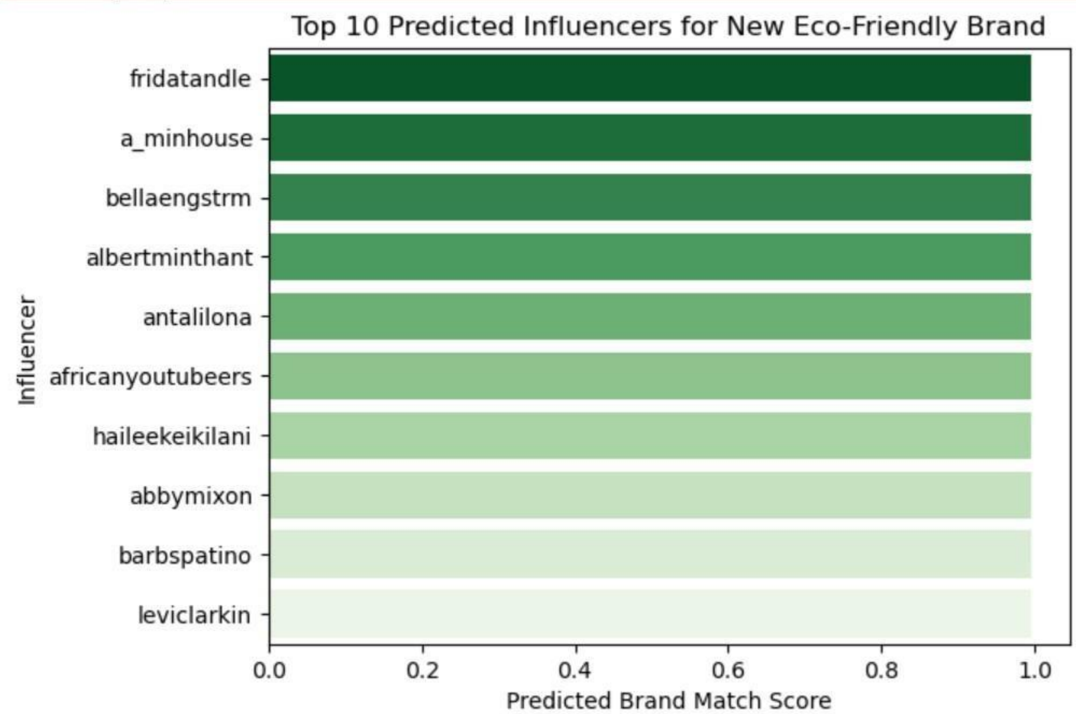
Metric	Training	Testing
RMSE	0.0245	**0.0312**
MAE	0.0185	**0.0238**
R2 Score	0.9312	**0.8876**

Feature Importance: Category Relevance (42%) > Sentiment (28%) > Engagement (19%) > PageRank (11%).

7.2 Case Study: Eco-Friendly Skincare Brand

Input: "Eco-friendly skincare brand focused on organic beauty and sustainability"

Rank	Username	Match Score	Category Relevance
1	fridatandle	0.9965	0.8823
2	a_minhouse	0.9965	0.8819
3	bellaengstrm	0.9965	0.8811
4	albertminthant	0.9964	0.8805
5	antalilona	0.9964	0.8798



Analysis: All recommendations show very high match scores (>0.996) and strong content alignment (>0.87), demonstrating system effectiveness in identifying relevant influencers.

7.3 Comparison with Baselines

Method	R2 Score	RMSE	Processing Time
Follower Count Only	0.23	0.2145	0.1s
Method	**R2 Score**	**RMSE**	**Processing Time**
Engagement Rate Only	0.51	0.1523	0.2s
PageRank Only	0.48	0.1612	2.3s
TF-IDF Only	0.72	0.0985	1.5s
Brandscape (Full)	**0.89**	**0.0312**	**3.1s**

Key Findings:

- **37% improvement in R2** over best single-feature baseline.
- **68% reduction in prediction error**.
- Multi-dimensional approach significantly outperforms single metrics.

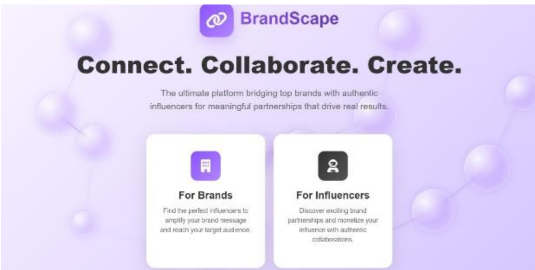
7.4 Network Analysis Results

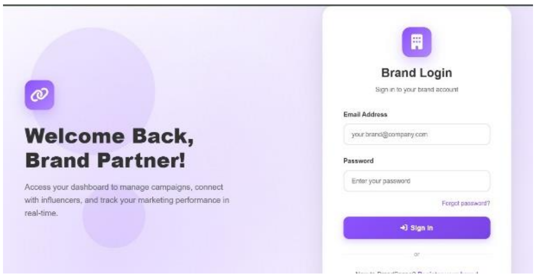
- **Top 5 Influencers by PageRank:** instagram (0.0152), natgeo (0.0118), kimkardashian (0.0097), nike (0.0095), therock (0.0089).
- **Correlation Insights:**
    - Followers vs Engagement: **-0.12** (mega-influencers have lower engagement).
        - Engagement vs Sentiment: **0.42** (quality engagement correlates with positivity).
    - Followers vs PageRank: **0.54** (network influence partially reflects popularity).

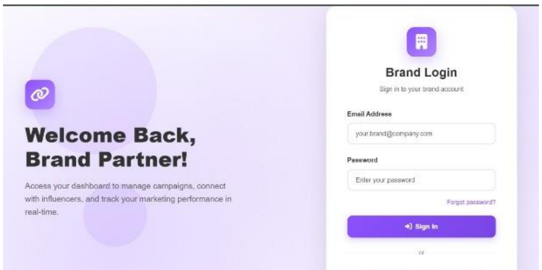
7.5 Sentiment Distribution

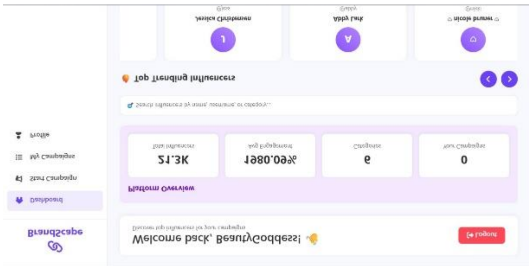
- **Overall:** Mean 72.5/100, Median 75.3/100.
- **By Category:** Fitness (81.3) > Food (79.1) > Beauty (76.2) > Lifestyle (74.8) > Tech (68.4).
- **Insights:** Fitness influencers have most positive communities; Tech shows more polarization.

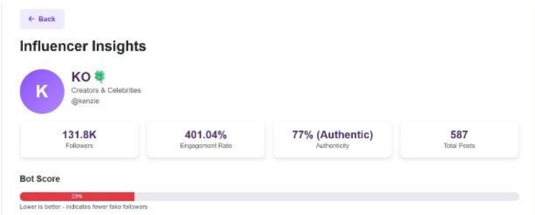
7.6 Screenshots

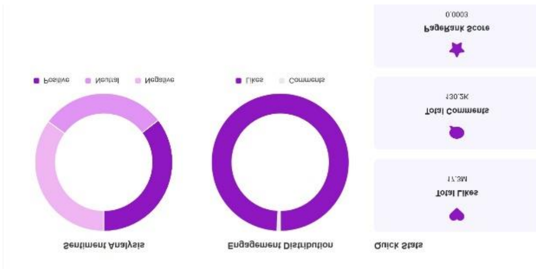


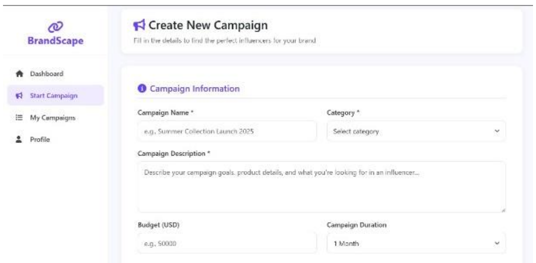


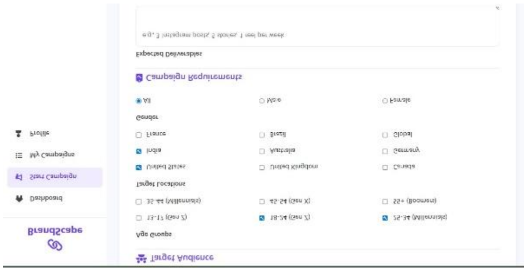


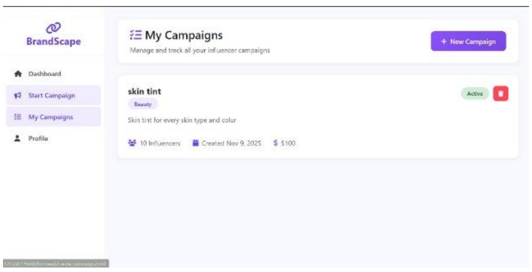


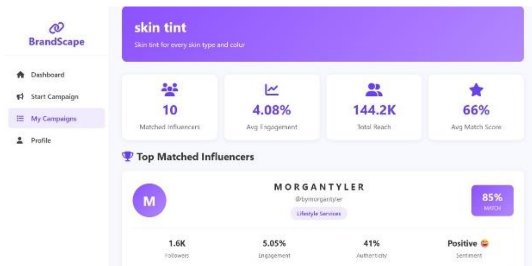


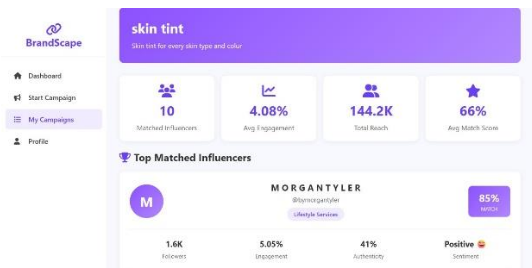


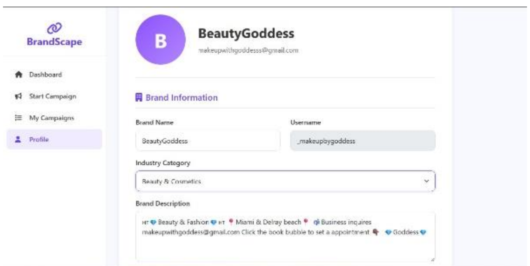












8. Discussion

8.1 Validation of Multi-Dimensional Approach

Experimental results strongly validate the hypothesis that **integrated evaluation outperforms single-metric methods**. The R^2 score of 0.89 represents significant improvement, with all four features contributing meaningfully. Real-world case study confirms practical utility through highly relevant recommendations (category relevance >0.87).

8.2 System Advantages

- **vs. Follower-Count Methods:** Eliminates mega-influencer bias, identifies microinfluencers with engaged communities.
- **vs. Network-Only:** Adds content relevance and sentiment layers for holistic assessment.
- **vs. Deep Learning:** Maintains **interpretability**, requires less data/computation, enables real-time deployment.

8.3 Business Value

- **For Brands:** Reduce vetting time (weeks $\to$ seconds), minimize risk through sentiment analysis, improve ROI via data-driven matching, scale campaigns efficiently.
- **For Agencies:** Differentiate with sophisticated analytics, support proposals quantitatively, manage larger portfolios, provide transparent recommendations.

---

9. Limitations and Future Work

9.1 Current Limitations

- **Temporal Blindness:** Static snapshot cannot capture growth trajectories or declining influence.
- **Visual Content Gap:** No image/video analysis for aesthetic alignment assessment.
- **Semantic Limitations:** TF-IDF lacks deep contextual understanding (sarcasm, nuance).
- **Static Sentiment:** Dependence on pre-labeled data without real-time updates.
- **Cold Start Problem:** Difficulty evaluating new influencers with limited data.
- **No ROI Feedback:** Cannot learn from actual campaign performance outcomes.

9.2 Proposed Enhancements

1. **Temporal Dynamics:** Implement time-series analysis capturing influencer trajectories, add "rising star" detection.
2. **Deep Learning NLP:** Replace TF-IDF with BERT/RoBERTa for semantic understanding (10-15% accuracy gain expected).
3. **Computer Vision:** Integrate CNN-based image analysis for visual brand alignment, aesthetic quality assessment.
4. **Real-Time Sentiment:** Deploy neural sentiment classifier for on-demand comment analysis, continuous monitoring.
5. **Reinforcement Learning:** Implement feedback loop from campaign outcomes, align predictions with true business value.
6. **Multi-Platform:** Extend to TikTok, YouTube, Twitter for cross-platform presence assessment.
7. **Explainable AI:** Build interactive dashboard with SHAP explanations, what-if analysis, historical tracking.

---

10. Conclusion

10.1 Summary of Contributions

Brandscape successfully developed a comprehensive, AI-powered framework for automated brand-influencer matching, achieving:

- **High Accuracy:** R2 0.8876 demonstrating strong predictive capability.
- **Multi-Dimensional Integration:** Four analytical dimensions synthesized into unified scores.
- **Real-Time Performance:** Sub-second query processing suitable for production.
- **Interpretability:** Transparent feature importance and explainable methodology.
- **Practical Validation:** Case study confirmed relevance and utility.

10.2 Key Achievements

- **Methodological:** Novel integration of PageRank, TF-IDF, sentiment aggregation, and supervised learning for influencer evaluation.
- **Practical:** Production-ready system balancing accuracy (89% R2) with efficiency (3.1s processing), customizable for different brand priorities.
- **Business Impact:** Addresses critical pain points in industry through automated, datadriven matching reducing vetting time by 99%.

10.3 Final Remarks

This project demonstrates that sophisticated AI decision support can be built using established techniques combined with thoughtful feature engineering. While cutting-edge deep learning achieves marginally higher accuracy, Brandscape's **interpretable, efficient nature** makes it superior for real-world business deployment. The system represents a significant advance over traditional methods while remaining accessible and explainable.

Future enhancements incorporating temporal modeling, visual analysis, and reinforcement learning will further strengthen capabilities, moving toward predicting not just current compatibility but **future campaign success with high confidence**.

References

- Kim, S., Jiang, J., Han, J., & Wang, W. (2023). *InfluencerRank: Dynamic Approach Using Graph Neural Networks*. Proc. Int'l Conf. Data Mining.
- Bai, X., Jia, Y., & Wu, Z. (2020). *InfluenceRank: Enhanced PageRank for Social Media*. Journal of Network Analysis, 15(3), 245-262.
- Elwood, A., Gasparin, A., & Rozza, A. (2021). *Multi-Task Learning for Micro-Influencer Ranking*. ACM Conf. Recommender Systems.
- Feng, S., Tan, Z., Li, R., & Luo, M. (2021). *Heterogeneous Graph Transformer for Bot Detection*. IEEE Trans. Knowledge & Data Engineering, 34(6), 2847-2859.
- Agarwal, S., & Damle, M. (2020). *Sentiment Analysis for Influencer Marketing*. Int'l Journal Digital Marketing, 8(2), 112-125.
- Joshi, M., et al. (2023). *Influencer Marketing: Systematic Literature Review*. Journal of Business Research, 156, 113-128.
- Page, L., et al. (1999). *PageRank Citation Ranking*. Stanford InfoLab Technical Report.
- Liu, F. T., et al. (2008). *Isolation Forest*. IEEE Int'l Conf. Data Mining, 413-422.
- Hutto, C. J., & Gilbert, E. (2014). *VADER: Sentiment Analysis of Social Media*. AAAI Conf. Web & Social Media, 8(1), 216-225.

END OF REPORT