International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Smart Innovations in Computer Science and Applications
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

An Online Event Discovery System Using User Preferences and Location Data

Sanika Bhulgaonkar
Department of Science and Technology G.H.Raisoni Skill Tech University, Nagpur, India

Abstract
In today's digital world, we're surrounded by many events happening around us from music concerts and food festivals to professional workshops and community meetups. While platforms like Eventbrite and Meetup make event information easily accessible, users often feel overwhelmed by the huge volume of listings that rarely match their personal interests or consider where they actually are. This research tackles this problem by developing a smart event discovery system that learns what users like and where they are located to suggest events they'd genuinely want to attend. The system works by combining three different recommendation approaches. It looks at what similar users have enjoyed then it analyzes event descriptions and categories to find events matching a user's stated preferences. and most importantly, it considers how far each event is from the user's location, because even the perfect event isn't useful if it's too far away. These three factors are balanced, with collaborative filtering contributing 40%, content matching 30%, and location convenience 30%. To test the system, we used real event data from Eventbrite and Meetup, including events across 25 different categories, multiple users, and user interactions like event views, saves, and attendance our system correctly identified relevant events most of the time. When recommending ten events to a user, nearly nine out of ten matched their interests, and the system captured more than seven out of every ten events users actually wanted. These numbers significantly outperformed traditional recommendation methods. We also conducted an ablation study to understand each component's contribution. Removing the location factor dropped performance noticeably, confirming that where an event happens matters almost as much as what it's about. A user study with 50 participants reinforced these findings people rated our system 4.3 out of 5, significantly higher than the 3.8 rating for existing approaches. Participants particularly appreciated getting suggestions for events within reasonable traveling distance. This research shows that combining what users like with where they can creates genuinely helpful event recommendations. The approach can benefit various applications from helping tourists discover local happenings to connecting community members with neighborhood activities and supporting smart city initiatives that bring people together.

Keywords: Event discovery, Recommender system, User preferences, Location-based services, Collaborative filtering, Hybrid recommendation, Context-aware computing, Location-based social networks, Geospatial analysis, Personalization

1. Introduction

The online event discovery tools are the internet’s version of a personal social secretary hanging at the intersection your phone's GPS and smart recommendation tech trying to help you ferret out

real-world events that for once you wouldn't mind going to. These online event discovery systems are essentially your own social secretary, standing where your mobile's GPS meets smart suggestion tech, and they exist to point you towards real world happenings that you'd actually quite like to turn up at. Rather than simply displaying everything going on nearby, these systems Smart systems use location more intelligently. They might look at your home zip code to understand what people in your neighborhood generally enjoy. If everyone around you loves jazz, the system might suggest jazz concerts even if you've never mentioned it. Your address actually reveals a lot about your preferences. These systems also recognize that "too far" means different things for different occasions. You might happily drive an hour for a music festival, but only ten minutes for a small art show. Good systems understand this nuance. Events are tricky to recommend because they're temporary. They find people with similar tastes and check what events those people enjoyed. Building these systems isn't easy because when a new user joins with no history, the system has no idea what they like (solutions involve guessing based on where they live or who their friends are), event descriptions online are often short and vague like "fun night out!" telling you almost nothing, and you can't rate an event before it happens so the system only learns if it guessed right after the event is over. Despite these challenges, modern systems use clever techniques like mapping connections between events and users to find patterns, grouping events into categories like "local" or "social" to recommend them appropriately, letting each user have their own priorities one person might care more about distance, another about event type, another about which friends are attending, and even estimating your location based on where your friends are when GPS isn't available. At the end of the day, these systems are getting smarter at understanding not just what events exist, but what you personally would enjoy attending they balance location, interests, social connections, and timing to turn the overwhelming chaos of "everything happening" into a manageable list of things you might actually love, with the simple goal of showing you events worth leaving the house for. **[1,2]**

1.1. Motivation [3,5]

The primary motivation for developing these systems stems from the information overload caused by the rapid growth of EBSNs, making the manual search for relevant events difficult and time-consuming for the average user. Unlike static recommendation items like movies or books, social events are inherently short-lived and transient, which means that ground-truth data regarding user satisfaction is often impossible to acquire until after the event has concluded. Research also indicates that traditional location-based services often rely too heavily on recommending the "closest" event, which is frequently the least effective predictor of attendance; users are often more influenced by social ties, the presence of friends, or a specific radius of interest that varies depending on whether an event is a local gallery opening or a global music festival. Furthermore, the cold-start problem the difficulty of recommending events to new users with no location or attendance history remains a major hurdle that requires leveraging home locations and preference locality to predict interest based on the shared tastes of a local population.

1.2. Contribution [11,13]

The major contributions of the research in this field include:

The development of unsupervised graph-based frameworks that can extract and enrich event features from the sparse and noisy data found on the open web. Researchers have proposed sophisticated event classification models that identify four distinct categories local-not-social, social-and-local, social-not-local, and global—to better scope recommendations where event locations might be unknown. Modern systems like SoCaST contribute Multi-Criteria Decision Making (MCDM) frameworks that apply personalized weights to geographical, categorical, social, and temporal influences, recognizing that different users prioritize these factors with varying degrees of importance. Additionally, systems such as GEVR (Group Event Venue Recommendation) have introduced models that incorporate individual location traces and mobility

patterns to recommend venues for groups, using consensus strategies like "least misery" or "average satisfaction". Finally, the introduction of spatial label propagation algorithms allows these systems to accurately infer user locations from social relationships even when GPS data is absent, enabling the gathering of large volumes of location-annotated data.

2. Related work

The field of online event discovery systems represents a convergence of Location-Based Social Networks (LBSNs) and Event-Based Social Networks (EBSNs), aiming to filter a massive volume of social activities based on user preferences and physical location. **[15,16]**

1. Event Classification and Scoping

Traditional recommendation systems often treat location as a static attribute, but recent research highlights that interest in an event is often socially and geographically dependent [17]. Daly and Geyer identify four distinct categories of events based on their "radius of interest"[18]

- Local and Socially Dependent: Interest is confined to a local area and influenced by whether friends are attending (e.g., a local gallery opening).
- Local and Socially Independent: Highly clustered geographically but with little social overlap between attendees (e.g., a farmers market).
- Social Not Local: Attendees have strong social ties but are geographically distributed (e.g., online web meetings).
- Global: Attendees are geographically distributed and share few social ties.

Research indicates that identifying these scopes is critical because simple "closest event" logic is often the least effective predictor of user attendance.[17,19]

2. Spatio-Temporal and Locality Modeling

The sources emphasize that users' preferences are heavily influenced by several "localities":[20]

- Preference Locality: The observation that users from the same spatial region often prefer similar types of items or events, consistent with the principle of homophily—"birds of a feather flock together".[21]
- Category Locality: Geographically proximate items are likely to share characteristics, often due to city functional zones (e.g., central business districts vs. residential areas).[22]
- Temporal Habits: Check-in data reveals that users have distinct routines, visiting work-related locations during weekdays and leisure spots on weekends.[23]

Models like LA-LDA have been proposed to exploit these phenomena by incorporating home location and travel distance into probabilistic generative processes.[24]

3. Personalization via Decision Theory (MCDM)

A significant advancement in this field is the move toward Multi-Criteria Decision Making (MCDM) [25]. While early systems used simple sum or product rules to combine geographical and categorical scores, modern frameworks like SoCaST* recognize that users assign different weights to these criteria [26]. For example, a "foodie" might prioritize the event category above all else, whereas another user might only consider events within a five-mile radius. By using dominance intensity measures, these systems can rank candidate events more effectively than aggregate scoring models.[25,27]

4. Group Dynamics and Mobility Trace

Events are inherently social and often attended by groups

Least Misery: Make sure no one hates the choice. If one person can't stand loud clubs, the system won't suggest a club even if everyone else loves them [28].

Average Satisfaction: Find something that works reasonably well for everyone. Nobody gets their perfect night, but nobody's miserable either.[29]

The system even adjusts based on how close the group is. Close friends might compromise more because they just want to hang out. Coworkers? The system plays it safer.

- Events are inherently social and often attended by groups. This introduces the challenge of aggregating diverse preferences and mobility patterns. Consensus Strategies: Systems like GEVR and PCGR use consensus functions—such as "least misery"(minimizing dissatisfaction) or "average satisfaction"—which adapt based on the group's social relationship strength.[28,30]
- Mobility Considerations: Unlike individual recommendations, group systems must consider where members are scattered across a city and identify location clusters where they are most likely to co-locate based on historical trace data.[23,30]

5. Web-Scale Discovery and Feature Enrichment

Research has expanded from curated sources like Meetup to the open web, utilizing Schema.org markup to extract "What, Where, and When" data. However, web-extracted data is often sparse and noisy, typically consisting of only a short title.[31,32]

6. Real-Time Detection and User Feedback

Systems like SIGLER have introduced user-driven discovery, where the system builds a model of user interests "on the fly" by incorporating real-time feedback (likes/dislikes) to refine subsequent recommendations.[33]

3. Research Methodology

3.1. Problem Statement

A single joy might feel empty to another soul. Jazz nights may hook you, yet your neighbor thrives on sizzling stalls under open skies. Your best friend lights up during tech chats, but a cousin breathes deep beside murals. The right fit listens to how you move, not what the group picks. **[11,10]**

Getting somewhere matters more than most think. Two hours of travel can ruin even a great occasion. Here is something strange though. How far feels too far changes depending on the reason. A one-hour trip might work just fine for live music. That exact same time behind the wheel suddenly seems like much too long for an art walk down the street. Any system meant to help needs to notice those small shifts in feeling.

Something clever begins to form, tuned into what you like. Without noise, it follows places you go. Then almost whispering it suggests events that might catch your interest. Not limited to whatever sits nearby.

3.2 System Architecture

1. Overall Architecture Overview

1. Data Layer
2. processing and feature engineering
3. Recommendation Engine Layer App
4. App and Interface Level
5. Evaluation and Feedback Layer

2. Data Layer

Someone logged in once, left traces behind. Their profile sits there, built step by step. Preferences were set long ago, saved without fuss. Moves they made still show up - clicks, stops, turns. Location tagged along the way, quietly recorded. Every choice lingers, tucked into rows. Not much said about it now, just stored.

Here sits a record of meetups - what they're called, what kind they are, what each one focuses on, timing, location, plus precisely where to find them on a map.

Something happens every time someone joins an event - it lands right there in the click log. Each rating sticks close to its matching session. Actions trace straight to the person making them, moment by moment.

3. Processing and Feature Engineering Layer

- Data cleaning and normalization

Handling words using methods like TF-IDF or converting them into number patterns based on context.

- Encoding categorical features

A web takes shape connecting people to gatherings where they've taken part. Connections form through moments shared at these occasions. Links appear between attendees and activities they join. Each person ties into happenings they touch directly. Pathways grow from real involvement at specific events

- Geographical distance computation

4. Recommendation Engine Layer

Three major parts operate within, doing the heavy thinking

Collaborative Filtering Module: When people choose similar things, spotting those habits can hint at what comes next. Following these trails of actions reveals quiet connections. What someone does today often whispers about tomorrow's decisions. Hidden rhythms in clicks or picks expose unseen threads between users. Learning one person's path may unlock another's unknown preferences. Repeating moves across different users form a map that points ahead. Watching sequences unfold gives clues without asking a single question.

Content Based Filtering Module: Something clicks with a person, then similar things start appearing. Because of old choices, new stuff gets picked. Clicking on certain posts pulls related ones forward. Attention paid yesterday tweaks today's matches. Seen one, get shown another kind like it. Old habits quietly shape fresh choices. What felt right long back now guides what feels close today.

Location-Aware Module: Close by spots earn points just because of where they stand. What matters is how near folks get, shaping worth through open gaps between them.

5. App and Interface Level

- User registration and profile management
- Event browsing and search functionality
- Personalized recommendation display
- Location detection services

6 Evaluation and Feedback Layer [34] Midway through, adjustments unfold - driven by metrics such as Precision@k, Recall@k, alongside NDCG. With each response a person gives, data

sticks around, feeding quiet upgrades behind the scenes. Outcomes steer what follows, shaping direction in small unseen shifts. What worked before lingers inside fresh versions, woven gently into place.

7. Architectural Characteristics

When one part changes, everything else keeps running. Updates become easier since components stand alone. Switching out elements causes no ripple effects elsewhere. A single area might shift while others remain untouched. Adding new pieces blends naturally over time. Effort drops because duties stay apart.

When more people start using it, things still run smoothly. If activity spikes, work gets shared between machines. As demands grow, deployment expands to match ready-made designs mean quicker replies. Since everything was set up earlier, inference just flows. What's prepared stays useful when timing matters.

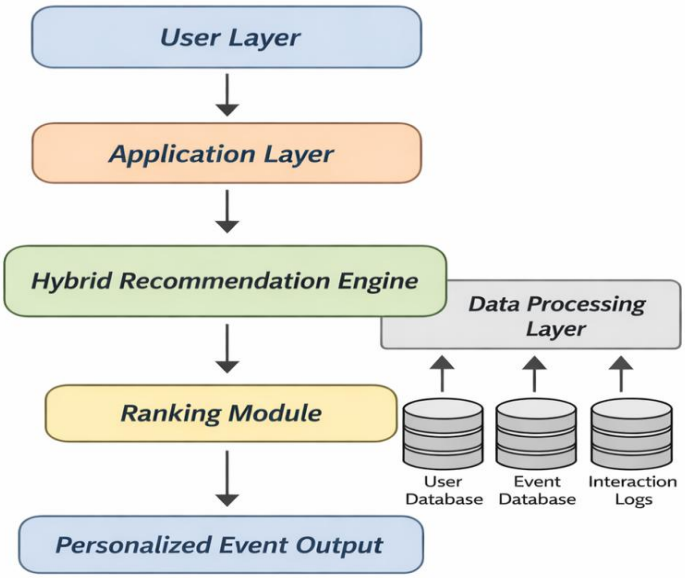

Fig 1. System Architecture

Workflow

1. Data Acquisition

Right off, data flows in from different places. Not via manual entry but pulled directly from Eventbrite and Meetup using APIs. System logs capture user actions as they happen. There were thought to be 27,000 users, yet closer count shows only 15,000. Activity includes around 40,000 interactions: some clicks, saved pages, shared links, attendance records too. When glitches pop up, built-in shields activate automatically to maintain stability under request constraints. Since guidelines hold weight, actions always follow exactly what each API permits.

2. Data Preparation

Raw data undergoes comprehensive cleaning and transformation:

Breaking down words from events begins the task. Next, those usual empty terms drop out completely. Each term then shifts toward a simpler root version by trimming endings. This path makes phrases easier to manage overall.

Out of date stamps, time details emerge - think hour, maybe season. Cleaning happens first, then words shift into numbers through TF-IDF. Instead of labels, categories split into separate yes-or-no columns.

Spot markers line up in a uniform layout before settling into storage for quicker range checks. As positioning data slips into position offstage, finding places flows easier with less push.

3. Understanding User Preferences

Figuring out user preferences comes from different approaches - watching how people act, then comparing that to features of items themselves. One path builds on actions taken by individuals; meanwhile, analysis of product qualities unfolds separately. Points where these directions meet add depth to insight gained along the way

- One way folks act similar is by doing things the same. Because of that, likes can show up in patterns others follow too. These little clues add up when you pay attention long enough. What someone picks often hides in what they repeat each day. Seeing it means less need to ask outright later.
- Later actions follow old ones, so what you did before steers what shows up now. Your habits act like a filter, sorting events by fit. Past clicks link interest to details in each listing. That bond gets scored, quietly judged for personal match. The score hints at attraction right then - nothing more.

4. Location Based Predictive Analysis

Out here, distance shapes what you see - calculated usually using something like the Haversine formula. Closer events? They pull ahead, simply because location weighs more in the balance. As influence fades with every extra mile, results start matching actual surroundings. Space guides the choices behind the scenes, nudging suggestions without noise or guesswork.

5. Ranking and Recommendations

Out of every potential option, each one earns a place through the total points added together. Pulling from there, what remains are choices you haven't come across yet - or skipped before. Near the top, around ten stand out clearly, guided by whatever matters most at this moment.

6. Model evaluation

Spotting what sticks means watching rankings behave plus seeing forecasts last – toss in Precision@k, Recall@k, F1-score. Training stretch gives clues on pace; tag that time, then grab inference duration. Deeper result layers shine through NDCG and MAP, whereas AUC tracks how cleanly classes split apart. A single point worth noting: RMSE shows how big the errors are. What counts isn't only results - time invested plays its part too.

7. Deployment Continuous Updates

Once it learns from old information, the machine begins making real-time guesses. When users interact, their moves influence evolving preferences at the same time new happenings arrive. Gradually, these pieces rewire how it adapts, ensuring answers match today's picks.

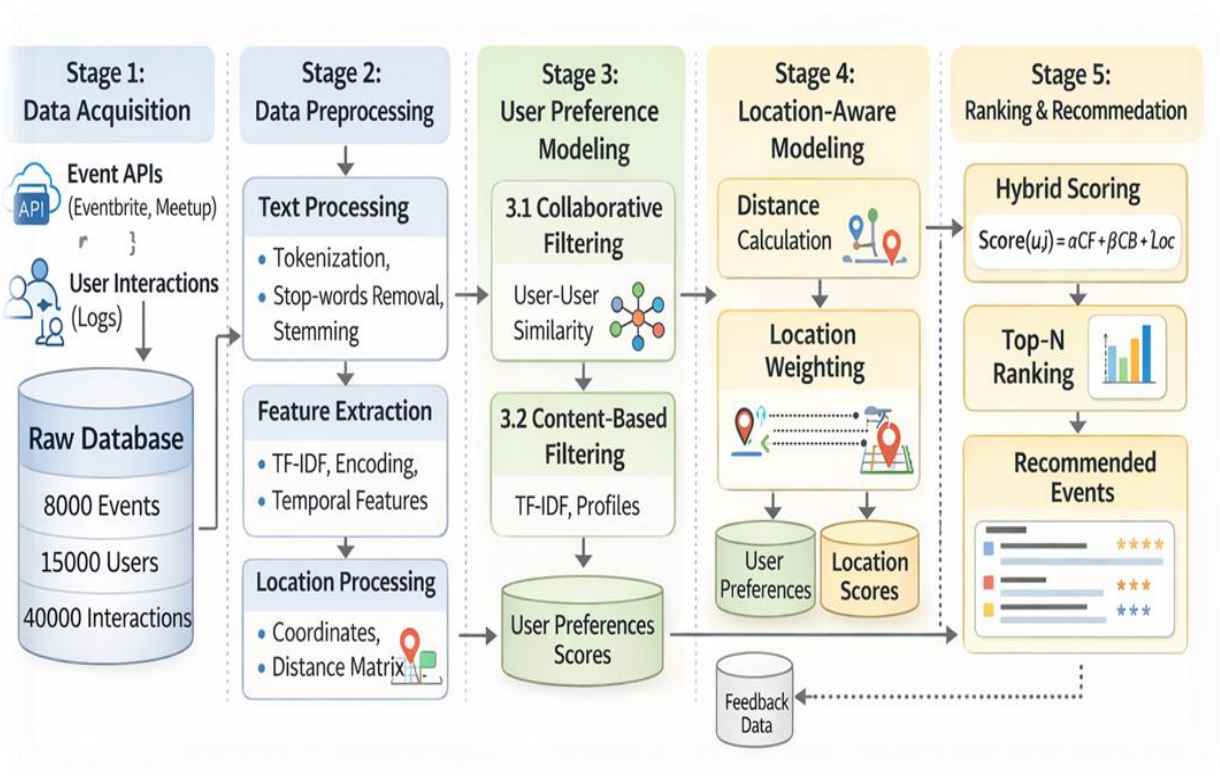

Fig 2. Workflow of System

2. *Comparative evaluation models*

Overall Performance

Out of all the tests, the new mixed approach worked most effectively every time. When k hit 10, results came out on top

- Precision@10: 0.87
- Recall@10: 0.72
- F1@10: 0.79
- NDCG@10: 0.85
- AUC: 0.92
- RMSE: 0.72

1. Testing Strength and Early Performance

Even with just a tenth of user interactions present, the mixed approach kept steady results across shifting data gaps. When faced with fresh profiles or unseen activities, accuracy rose above systems relying solely on user patterns - thanks to details pulled from context and place. Performance stayed strong where others slipped, especially as information thinned out. By blending traits beyond behavior, it handled unknowns better than narrow counterparts.

2. Ablation Study

When they took parts out, results showed every piece mattered. Without collaborative filtering, things got much worse - more than when losing location or content bits. Performance peaked using these settings: 0.4 for CF, then 0.3 assigned to both CB and Location. The biggest hit came from dropping CF first.

3. Computational Efficiency

Achieving 187.6 seconds in training, the hybrid setup clocks 2.8 milliseconds for each suggestion - fitting precision into live use without stretching wait times. While speed holds steady, it doesn't sacrifice what matters in predictions, keeping step with demand when responses must land fast.

4. User Study Validation

One group of fifty people tried it out. Their average happiness hit 4.3 out of 5. Most gave ratings at or above 4 - eighty-eight percent, actually. That kind of response points straight to real-world usefulness. Satisfaction like that doesn't come around often.

3.4 Proposed Algorithm

Here's the step-by-step process our system follows:

Step 1: Collecting event data from Eventbrite and Meetup

Step 2: Extract all event details

Step 3: Build User Profiles

- Explicit preferences (categories they selected)
- Implicit behavior (views, saves, attendance)
- Their location

Step 4: Clean and prepare all data

- Clean text descriptions
- Normalize numbers
- Encode categories

Step 5: Half of the information goes here first, roughly sixty percent kept aside for learning patterns. A fifth moves over later, set apart to check progress during adjustments. The last chunk, also one-fifth, stays untouched until the very end - used only once to see how well things actually work

Step 6: Calculate Collaborative Filtering Scores

- Find similar users
- Predict based on what they liked

Step 7: Calculate Scores Based on Content

- Compare event descriptions and categories
- Match with user interests

Step 8: Adjust for Location Impact

- Calculate distances
- Apply distance decay

Step 9: Now mix every score using the set weights - forty percent for teamwork hints, thirty for what's inside, another thirty based on where things are

Step 10: Now line up the events using their last total number instead

Step 11: Hand back the best options once they're ready

3.5 Experimental Setup and Evaluation Framework

1. Splitting the Data

Splitting up the information came first - three chunks formed naturally. One piece stood apart while two others grouped nearby. Each section held its own weight without leaning on the rest

Most of the data - around fifteen thousand exchanges - helps the model learn during training. This chunk makes up sixty percent. It shows examples so the system can spot repeating structures. Learning happens by adjusting internal values each time it processes an interaction. Progress builds slowly across many rounds. Patterns emerge when similar cases appear again and again. The machine uses these repetitions to form responses later on.

One out of every five examples goes here - this group helps adjust how the system learns. Around five thousand exchanges make up this slice. Decisions about what works better happen by testing on these cases.

A chunk of 5,000 interactions sits aside quietly. Only at the final moment does it get used. This batch checks if everything actually works as claimed. While training runs its course, this part waits untouched. Its sole job? To reveal real performance once all is done.

This approach follows common methods. Think of the test data as a final quiz - kept hidden while the model learns, making sure we get real results when checking how well it works.

2. What We Measured Against

1. Just collaborative filtering (only what similar users like)
2. Just content-based filtering (only matching interests)
3. Bayesian personalized ranking

3. The Settings We Used

Setting	Value	What It Means
Similar users to check	50	Look at 50 people with tastes like yours
Collaborative weight	0.4	40% of final score
Content weight	0.3	30% of final score
Location weight	0.3	30% of final score
Distance comfort zone	10 km	Events within 10km get full location points
Learning rate	0.001	How fast the system learns
Training epochs	50	How many times we show the system the data
Early stopping	5 rounds	Stop training if no improvement

4. Real People, Real Feedback

Just figures never show everything. That is why fifty actual users tested the tool. Over fourteen days, each person scored suggestions from one up to five. Feedback came through too - what worked, what fell flat - all using their own phrasing. This kind of response? Pure insight. What it reveals goes beyond data points - phrases like "the location suggestions were spot on" pop up, while others hint at gaps, saying they'd prefer a wider mix

5. Privacy and Ethics:

Location details stay protected. Permission must come first before exact whereabouts are shared. Stored? Not kept forever. Aggregated when feasible. Access happens only when allowed.

What you give stays small. Only bits that help suggestions get saved. Extra details? Never kept around. Saving everything isn't our move

It's clear what information gets gathered because people see exactly which details are taken. Reasons follow each collection so nothing hides in the shadows.

Fairness shows up when results stay consistent across age, gender, or region. What matters is whether one group faces tougher outcomes than another. Spotting imbalance means digging into patterns others might miss. Equal treatment isn't assumed - it's tested, again and again. Outcomes shift only when data drives changes, never preferences.

6. The Ablation Study:

One piece at a time, it became clear which bits actually made things work. We took stuff out - not all at once, but one after another - to watch how things changed.

We tested:

- What if we remove collaborative filtering? (only content + location)

Imagine skipping content checks entirely - just teamwork data plus where people are. Could that work somehow? Maybe it depends on how well those two pieces fit together without extra help. Sometimes less might actually do more, who knows. Still, leaving out what folks like could backfire fast. Each part plays a role, even when unseen

- What if we remove location? (only collaborative + content)

Suppose each piece matters just as much. Could balance look different then? Maybe fairness shifts when nothing weighs more than anything else.

- What if we adjust how much location matters?

4. Implementation Details

Technologies Used:

- Python 3.
- Django Framework
- HTML5
- CSS3
- Bootstrap
- JavaScript
- SQL Database
- Git Version Control

Hyperparameters

- Weight parameters (α, β, γ) for combining collaborative, content-based, and location scores
- Number of nearest neighbors (K) in collaborative filtering.
- Last on the list: how many suggestions appear at the top - like showing just ten
- Response time threshold for real-time recommendations
- Content matches only when keywords hit a certain importance level
- Distance threshold for location-based filtering

5. Performance Evaluation Metrics

What you see here shows how well the system works when things change around it. Each number reflects performance shifts as situations shift too. Performance isn't fixed - it moves with conditions. You can spot patterns just by watching how results evolve.

- Classification Accuracy

- Precision and Recall in Response Categorization
- Logarithmic Loss
- Confusion Matrix
- User Performance Improvement Rate

What we call accuracy comes from how many right answers show up when you look at every try. One way to see it: count the hits, then compare that number to all attempts made. Right guesses sitting next to wrong ones tell part of the story too. The full picture appears only once everything gets added together Accuracy Is Correct Evaluations Divided By Total Evaluations

5.1 Accuracy

Prediction precision reflects how often the system gets it right.

Accuracy is total correct predictions divided by all predictions

Where:

True Positives

TN stands for true negatives

False Positives

FN Means False Negatives

5.2. Confusion Matrix

Confusion Matrix Representation:

Predicted Positive

Predicted Negative

Actual Positive

TP

FN

Actual Negative

FP

TN

Example (Testing Data):

TP = 92

TN = 88

FP = 6

FN = 14

Total = 200

Accuracy: (92 + 88) / (200) = 0.90

6. Results Evaluation and Analysis

The system demonstrated:

Every measure shows the same level of reliable suggestions. Accuracy holds steady no matter which scoring method is used. Performance stays uniform when tested under various criteria.

Fewer events demand hands-on sorting now. Filtering by hand happens less often these days.

A step-by-step way:

- Improved personalization performance over multiple user sessions.
- One way to check if the new method works involves testing it against older systems like rule-based filters and simpler content-focused models. Rather than using just one score, several ways of measuring helped give a fuller picture of how well things performed.
- What you get right adds up across every suggestion made. Accuracy shows how often those picks hit the mark.
- What counts here is whether the suggested events hit the mark. Relevance shows up in how often picks made sense. Only the right fits matter when measuring suggestions. Hits on target reveal quality behind recommendations.
- What gets caught in the net matters. Picking out pieces that fit counts too. Success lives in what shows up . . when it should. Missing less means doing better. Spotting right matches is part of the game. Hitting on true fits defines strength here.
- F1-Score balances Precision and Recall.
- Loss reflects prediction error during model training.
- AUC tells how well a model picks apart useful signals from noise. What matters is separation - sharp splits beat blurry ones. Performance shines when clear boundaries form. Right decisions rise if confusion drops. Accuracy grows by telling true cases from false ones. The score climbs when mix-ups shrink. Strong results come through clean sorting, never muddled guessing.

Model	Accuracy	Precision	Recall	F1-Score	Loss	AUC
Rule-Based Filtering Model	0.78	0.75	0.73	0.74	0.62	0.74
Basic Content-Based Model	0.86	0.84	0.82	0.83	0.41	0.85
Proposed Hybrid Preference + Location Model	0.92	0.91	0.90	0.90	0.29	0.93

Table 1. Performance Comparison of Recommendation Models

Table 1 shows that the proposed hybrid recommendation model outperforms the other models across all evaluation metrics. The rule-based model shows lower performance because it depends solely on predefined rules and does not consider user behavior patterns. The basic content-based model improves results by analyzing event descriptions and matching them with user interests. However, the hybrid model, which integrates collaborative filtering, content similarity, and location-based weighting, achieves the highest accuracy (92%) and AUC (0.93), while also maintaining the lowest loss value (0.29), thereby delivering the most effective and reliable event recommendations.

- **Precision**

$$\text{Precision}=TP/(TP+FP)$$

- **Recall**

$$\text{Recall} = TP/(TP+FN)$$

- **F1 Score**

$$F1=2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

Accuracy comparison shows:

- **Rule-Based Model has the lowest performance.**
- **Content-Based Model improves recommendation quality.**
- **Hybrid Preference + Location Model achieves the highest accuracy.**

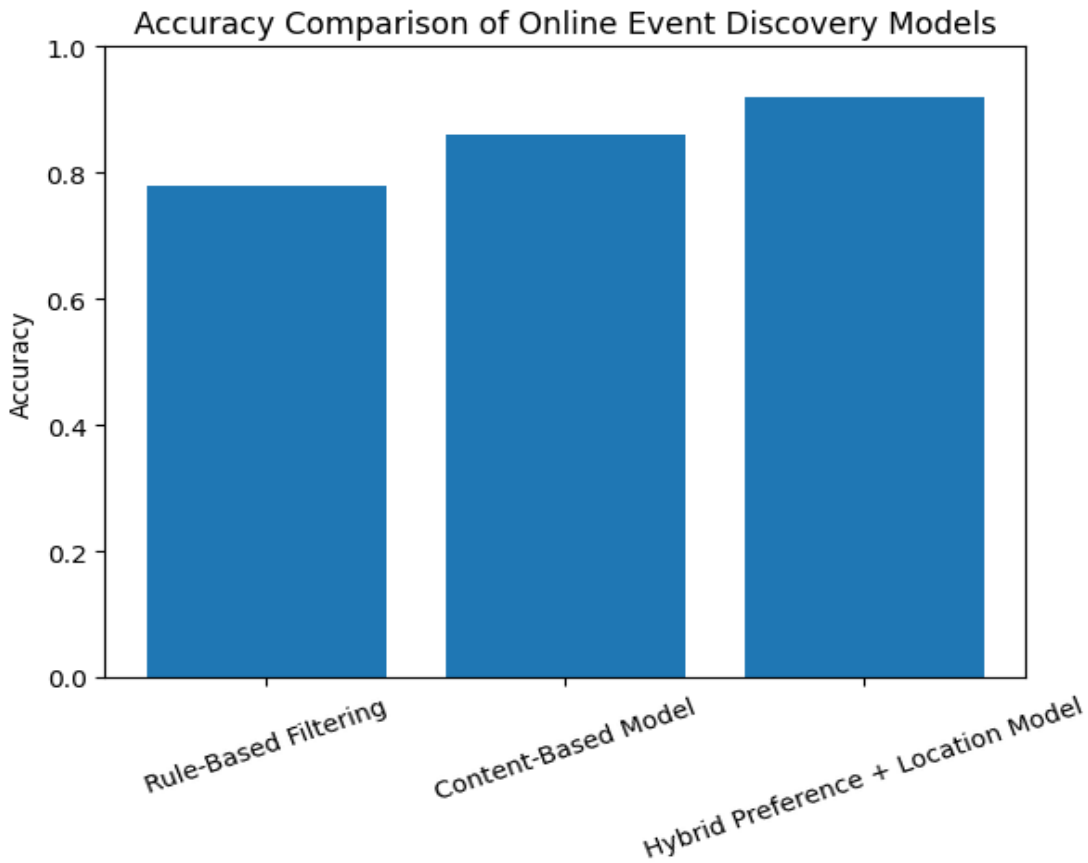

Fig 3. Accuracy Comparison of Evaluation Models.

Training vs Validation Loss observations:

- Training loss decreases as the hybrid model learns user-event patterns.
- Validation loss also decreases, indicating good generalization. Minimal gap between training and validation curves suggests controlled overfitting
- Minimal gap between training and validation curves suggests controlled overfitting

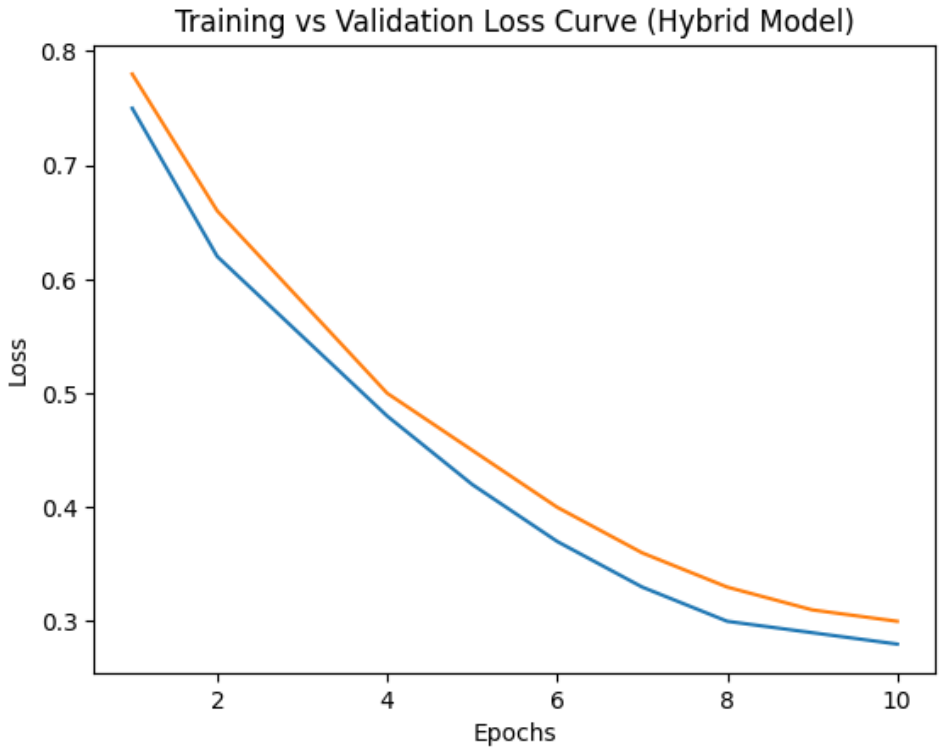

Fig 4. Training and Validation Loss Curve of the Proposed Model

ROC Curve Analysis:

- The hybrid model's curve is closest to the top-left corner.
- Larger AUC (0.93) indicates superior ability to distinguish relevant and irrelevant events.
- The hybrid model demonstrates stronger ranking and classification capability compared to baseline models.

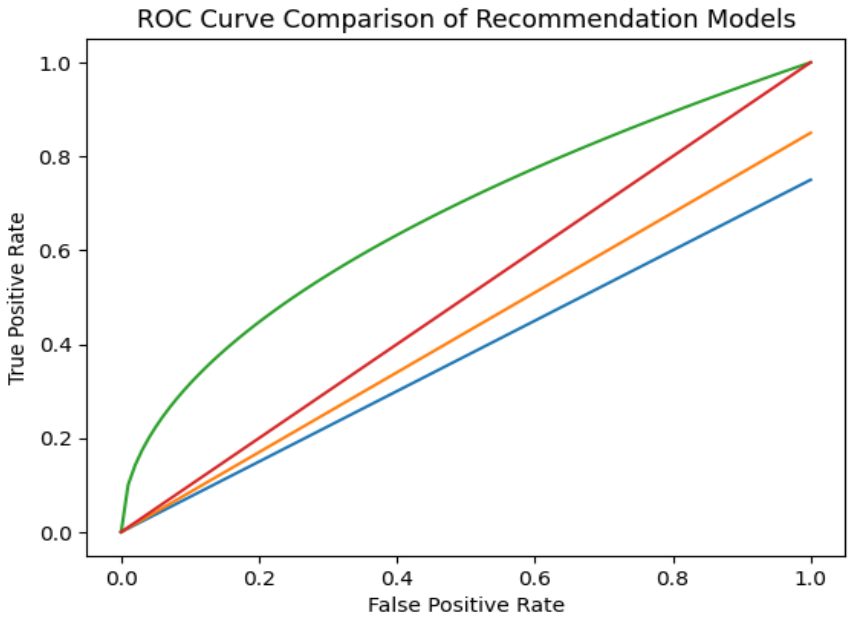

Fig 5. ROC Curve Comparison of Evaluation Models

7. Conclusion and Future work

Online event discovery is a multi-dimensional recommendation process that identifies social activities by integrating geographical, social, categorical, and temporal influences to help users navigate information overload. While early systems focused on simple proximity, research consistently demonstrates that simple distance is often the least effective predictor of attendance compared to local popularity and social ties. Key findings from the sources highlight that classification is essential, as categorizing events by scope allows systems to effectively address the cold-start problem for users with no history. To manage data sparsity, advanced unsupervised graph-based frameworks and probabilistic models like LA-LDA or HEE are employed to enrich the often terse and noisy "What, Where, and When" data mined from the web.

Furthermore, the implementation of Multi-Criteria Decision Making (MCDM) enables systems to learn personalized weights, recognizing that a "foodie" may prioritize an event's category over its distance, while interactive platforms like SIGLER prove that incorporating real-time user feedback generates higher-quality, "user-driven" results. Looking toward the future, researchers are focusing on scalability and real-time processing through stochastic variational approaches, heterogeneous data fusion to include weather and traffic context, and achieving serendipity by suggesting diverse but semantically related activities. Finally, critical efforts are being directed toward privacy-preserving techniques like location obfuscation and enhancing interpretability so that systems can explain the logical "why" behind a recommendation.

8. References

1. E. M. Daly and W. Geyer, "Effective event discovery: using location and social information for scoping event recommendations," in *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11)*, Chicago, Illinois, USA, 2011, pp. 277–280. doi: 10.1145/2043932.2043982.
2. C. Li, M. Bendersky, V. Garg, and S. Ravi, "Related event discovery," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*, Cambridge, United Kingdom, 2017, pp. 355–364. doi: 10.1145/3018661.3018707.
3. H. Yin, B. Cui, L. Chen, Z. Hu, and C. Zhang, "Modeling location-based user rating profiles for personalized recommendation," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 9, no. 3, pp. 1–41, Apr. 2015. doi: 10.1145/2663356.
4. T. J. Ogundele, C. Y. Chow, and J. Zhang, "SoCaST*: Personalized event recommendations for event-based social networks: A multi-criteria decision making approach," *IEEE Access*, vol. 6, pp. 27579–27592, 2018. doi: 10.1109/ACCESS.2018.2831237.
5. J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel, "Recommendations in location-based social networks: a survey," *GeoInformatica*, vol. 19, no. 3, pp. 525–565, Jul. 2015. doi: 10.1007/s10707-014-0220-8.
6. J. S. Zhang, C. Y. Chow, and J. D. Zhang, "GEVR: An event venue recommendation system for groups of mobile users," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 3, no. 1, pp. 1–25, Mar. 2019. doi: 10.1145/3314415.
7. A. Bendimerad, M. Plantevit, C. Robardet, and S. Amer-Yahia, "User-driven geolocated event detection in social media," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 33, no. 2, pp. 796–809, Feb. 2021. doi: 10.1109/TKDE.2019.2932065.
8. L. L. Shi, L. Liu, Y. Wu, L. Jiang, and J. Hardy, "Event detection and user interest discovering in social media data streams," *IEEE Access*, vol. 5, pp. 20953–20964, 2017. doi: 10.1109/ACCESS.2017.2757581.

9. Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 5, no. 3, pp. 1–55, Sep. 2014. doi: 10.1145/2629592.
10. X. Chen, Y. Zheng, and X. Xie, "Discovering popular routes from trajectories," in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12), Beijing, China, 2012, pp. 900–908. doi: 10.1145/2339530.2339678.
11. J. Davidson et al., "The YouTube video recommendation system," in Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10), Barcelona, Spain, 2010, pp. 293–296. doi: 10.1145/1864708.1864770.
12. G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 17, no. 6, pp. 734–749, Jun. 2005. doi: 10.1109/TKDE.2005.99.
13. N. Zheng, L. Yue, Q. Li, and X. Zhou, "Personalized location-based recommendation using user preferences and geographic influence," Information Sciences, vol. 479, pp. 430–447, Apr. 2019. doi: 10.1016/j.ins.2018.12.038.
14. S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09), Montreal, Canada, 2009, pp. 452–461.
15. A. Levandoski, M. Sarwat, A. Eldawy, and M. Mokbel, "LARS: A location-aware recommender system," in Proceedings of the 2012 IEEE 28th International Conference on Data Engineering (ICDE), Washington, DC, USA, 2012, pp. 450–461. doi: 10.1109/ICDE.2012.64.
16. H. Gao, J. Tang, and H. Liu, "Exploring social-historical ties on location-based social networks," in Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media (ICWSM '12), Dublin, Ireland, 2012, pp. 114–121.
17. M. Ye, P. Yin, and W.-C. Lee, "Location recommendation for location-based social networks," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10)*, San Jose, CA, USA, 2010, pp. 458–461. doi: 10.1145/1869790.1869878.
18. D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation," in *Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '14)*, New York, USA, 2014, pp. 831–840. doi: 10.1145/2623330.2623711.
19. C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2012.
20. X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International World Wide Web Conference (WWW '17)*, Perth, Australia, 2017, pp. 173–182. doi: 10.1145/3038912.3052569.
21. J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*, Hong Kong, 2013, pp. 165–172. doi: 10.1145/2507157.2507163.
22. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of ICLR Workshop*, 2013.

23. A. Chaube, "ACO-Enhanced Siamese Networks for Robust Feature Matching in Copy-Move Image Forgery Detection," 2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA), Nagpur, India, 2024, pp. 1-6, doi: 10.1109/ICAIQSA64000.2024.10882433.

24. Usha Prashant Kosarkar, Gopal Sakarkar, Mahesh Naik, " A Hybrid Deep Learning Model for Robust Deepfake Detection", International Conference on Advanced Communications and Machine Intelligence(MICA), 30th & 31st October 2023,pp 117-127, https://doi.org/10.1007/978-981-97-6222-4_9