MatchGPT:
Dual-Prediction Match Analysis System

Swipe smarter, not harder

linkedin.com/in/denis-beliaev
instagram.com/geekkon_x
Belyaevda.gk@gmail.com

Essence

MatchGPT is a **two-stage machine learning system** that **predicts both meeting probability and relationship success**. It analyzes the complete relationship cycle from online matching to real-world outcomes, focusing on connection quality rather than just match quantity.

'Met'

'Met' Prediction Function

This model calculates **meeting probability** using **demographic data and platform interaction metrics**. It processes age, physique, app source, and communication channels to identify which matches will likely lead to real meetings.

'Success'

Relationship 'Success'

Success means a **positive post-meeting outcome with relationship potential**. It's measured by combined metrics: emotional comfort, chemistry, physical satisfaction, and communication alignment.

linkedin.com/in/denis-beliaev

MatchGPT Relevance:
Efficiency in the Dating Era

Waste of Time

Users spend excessive time on virtual conversations that fail to lead to real-life meetings or meaningful connections.

Inefficient Matching

Platforms focus on maximizing "matches" or "swipes" but lack the tools to predict which connections have real-world potential.

User Frustration & Churn

The constant cycle of promising online interactions that lead to disappointing dates results in user fatigue and app abandonment.

Dual-Layer Prediction

Provides clear probabilities for both a meeting occurring and its potential success, setting realistic expectations and prioritizing effort.

User Experience Enhancement

Guides users toward more promising connections, reducing time wasted on dead-end conversations.

Actionable Analytics

Identifies and quantifies the key demographic, behavioral, and compatibility factors that lead to successful relationships.

GitHub Repository link: https://github.com/dbeliaev-tum/match-gpt.git

Algorithm

1 Data Preparation

We clean and structure raw user data for machine learning. This involves converting height ranges to numerical values, encoding categorical features like 'body type' and 'app source', and creating new combined features such as 'age-height interaction' to help the model detect complex patterns.

2 Meeting Model

This first predictive layer estimates the probability that two users will meet in person. The Random Forest algorithm analyzes demographic factors and platform interaction data to identify which online matches are most likely to lead to real-world dates, serving as the initial filter.

Class Balancing

We address data imbalance using SMOTE technique. Since successful meetings and relationships are naturally rare in datasets, this algorithm synthetically creates similar examples of the minority class to prevent the model from being biased toward predicting more common negative outcomes.

3

Success Model

This second predictive layer activates only for pairs who are likely to meet. It analyzes deeper compatibility metrics - emotional comfort, chemistry, communication style - to forecast the long-term potential of the relationship, providing a dual-layer assessment.

4

Analytics

The system generates actionable “business” intelligence beyond simple predictions. We provide ranked feature importance, showing exactly which factors most significantly influence both meeting probability and relationship success, enabling data-driven platform optimization.

5

Application

The trained models are deployed for real-time predictions. Users and platforms receive clear probability scores for both meeting and success, enabling better decision-making, personalized recommendations, and more efficient connection prioritization.

Data (meeting model)

Title	age	height	body_type	region_origin	sphere_of_employment	app_source	chat_channel	met
20936	48	160-170	overweight	WesternEurope	student	tinder	whatsapp	0
99122	43	170-180	slim	CentralEurope	unemployed	grindr	telegram	1
45112	36	180-190	mediocre	EasternEurope	self-employed	instagram	snapchat	
19952	52	190-200	average	SouthEurope	it / tech	Bumble	instagram dm	
48318	44	200-210	athletic	NorthAmerica	finance / banking / consulting	Hinge	grindr	
53012	30		muscular	SouthAmerica	healthcare / medicine	other	tinder	
87549	25			Asia	education / academia		other	
69978	33			Africa	arts / media / design			
3236	29			MiddleEast	sales / marketing			
45626	43				engineering / manufacturing			
28767	37				service / hospitality			
87312	36				public sector / government			
31661	35				other			

To get a prediction, input data based on your experience and observations to personalize the model. The data includes basic demographic features available on dating app profiles.

- *app_source* – The platform where you met the person of interest
- *chat_channel* – The platform you use to communicate with them

Data (success model)

days_to_meet	sex_quality	emotional_comfort	chemistry	communication_style_match	relationship_outcome	activity
1-7	0	1	1	1	0	active
7-15	1	2	2	2	1	not active
15-30	2	3	3	3	2	
30-90	3	4	4	4		
other	4	5	5	5		
NA	5					

If no meeting has occurred yet, these fields are automatically filled as “NA”.

- *relationship_outcome* – Indicates how your interaction developed after the first meeting
- *activity* – For relationship_outcome, activity is set to “not active”

Dual Architecture

Data

Meeting Prediction

Analyzes likelihood of real-world meeting

- Processes demographic & platform data
- Uses Random Forest classifier
- Serves as initial quality filter
- Key factors: Age, height, app source, chat channel

Success Prediction

Evaluates relationship potential for the met matches

- Requires meeting probability > threshold
- Incorporates compatibility metrics
- Uses separate Random Forest model
- Key factors: Emotional comfort, chemistry, communication style

Architecture Advantages

The dual architecture ensures that resource-intensive compatibility analysis is only performed on matches with highest meeting potential, creating an efficient and targeted prediction system

- Sequential Filtering
- Specialized Models
- Resource Efficiency
- Actionable Insights

Meeting model learning Outcome

MEETING FEATURE IMPORTANCE TABLE

Feature	Importance Success
age	0.124577
chat_channel_whatsapp	0.091874
age_height_interaction	0.088737
height_num	0.079194
body_type_athletic	0.055275

...

📊 **MEETING MODEL ACCURACY: 70.20%**

Meeting Model Classification Report:

	Precision	Recall	F1-Score
0	0.58	0.35	0.44
1	0.73	0.88	0.8
macro avg	0.66	0.61	0.62
weighted avg	0.68	0.7	0.68

📊 **Cross-Validation Results (Meeting Model):**

Accuracy:	0.696 (±0.008)		
Precision:	0.736 (±0.005)		
Recall:	0.851 (±0.010)		
F1:	0.790 (±0.006)		

*The test dataset is suitable for men 20-25 y.o., 170-180 cm, EasternEurope, Student

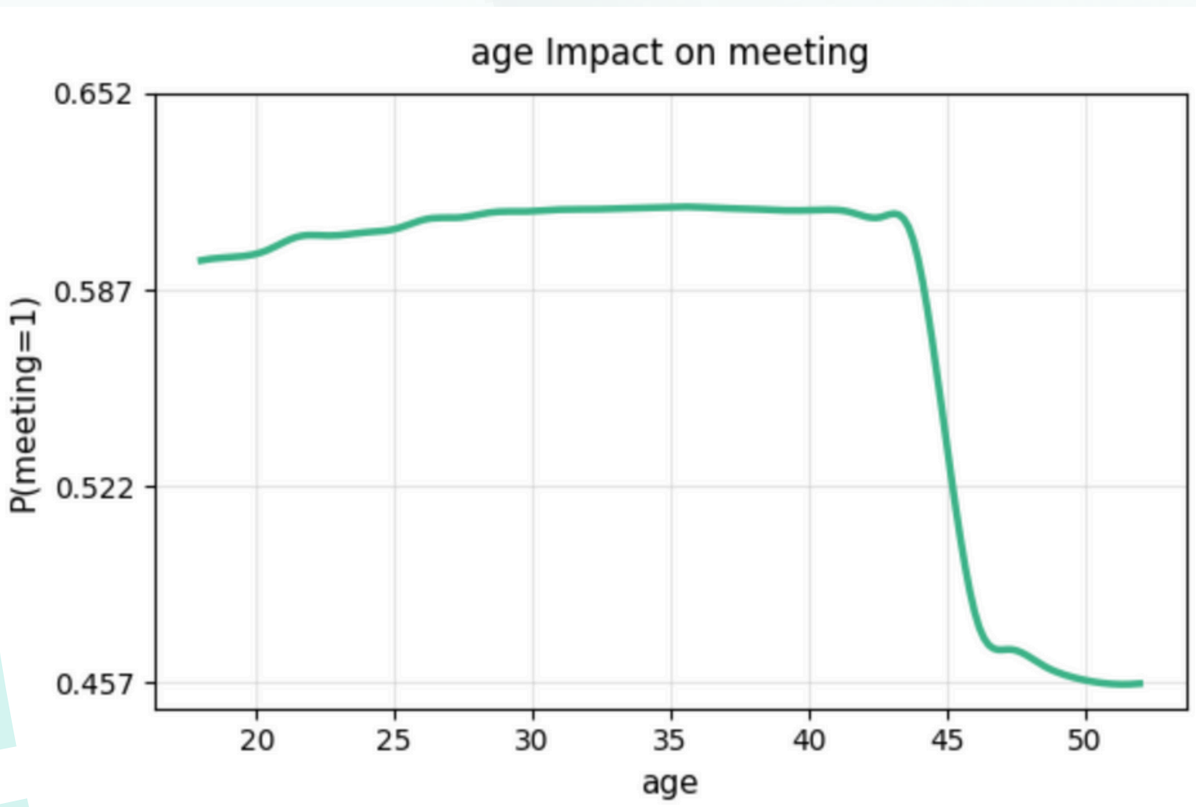
Categorical Factors (Meeting) - Sorted by Impact

Factor	Value	P(success=1)	Δ from average	Direction
chat_channel	whatsapp	0.644086	0.06564	↑ increases
region_origin	SouthEurope	0.609656	0.031211	↑ increases
body_type	athletic	0.606704	0.028259	↑ increases
region_origin	NorthAmerica	0.600362	0.021916	↑ increases
region_origin	MiddleEast	0.595534	0.017088	↑ increases
...				
chat_channel	telegram	0.561916	-0.01653	↓ decreases
sphere_of_employment	public sector / government	0.556761	-0.021684	↓ decreases
chat_channel	tinder	0.550022	-0.028423	↓ decreases
chat_channel	snapchat	0.547256	-0.031189	↓ decreases
region_origin	WesternEurope	0.542259	-0.036186	↓ decreases

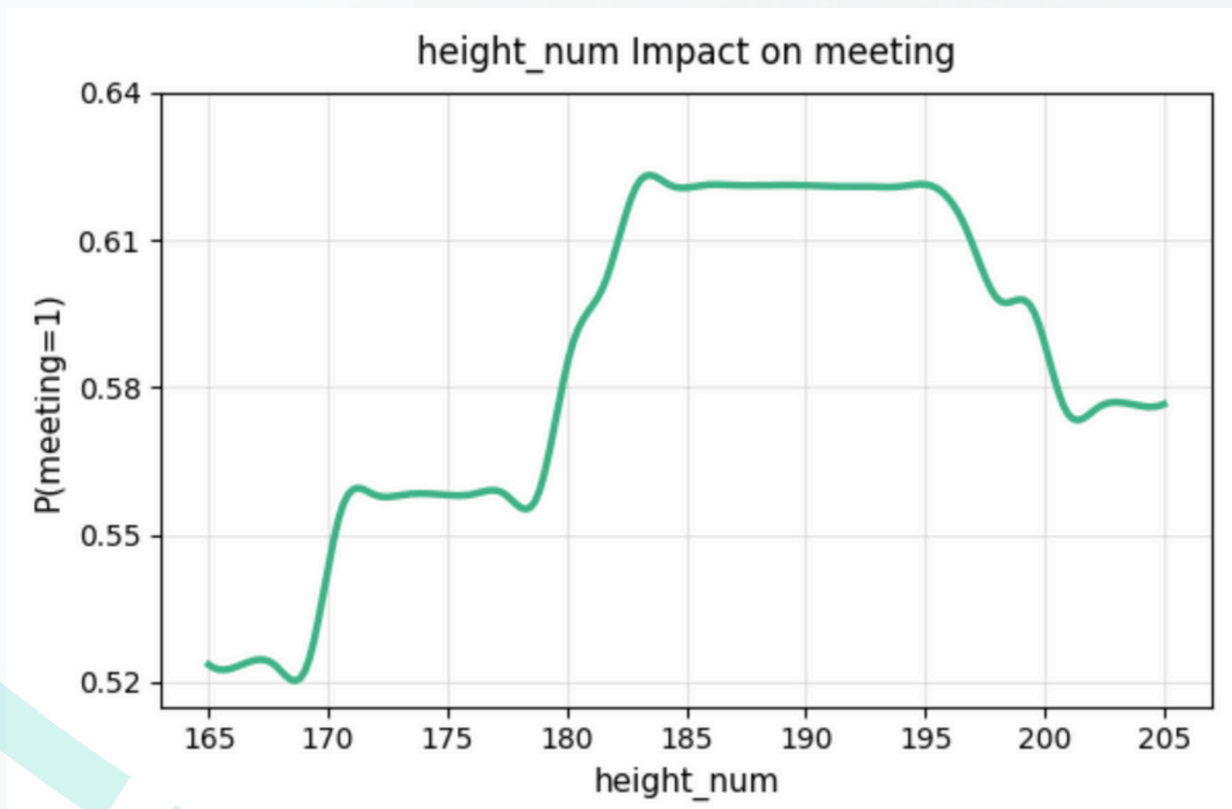
The analysis successfully distinguishes between factors that predict real-world meetings and those that indicate conversations will remain within the dating app.

linkedin.com/in/denis-beliaev

Numerical Factors (Meeting)



Based on the dataset, people over 43 years old are less inclined to meet.



The influence of height on meeting potential has a step-like pattern. Individuals with heights of 185-195 cm (higher than average) are most predisposed to meet. Meanwhile, people shorter than 170 cm (lower than average) are less inclined to meet.

Success model learning Outcome

SUCCESS FEATURE IMPORTANCE TABLE

Feature	Importance Success
communication_style_match	0.152463
emotional_comfort	0.149078
sex_quality	0.103857
chemistry	0.083407
app_source_tinder	0.063343

...

📊 SUCCESS MODEL ACCURACY: 87.81%

Success Model Classification Report:

	Precision	Recall	F1-Score
0	0.77	0.4	0.53
1	0.89	0.98	0.93
macro avg	0.83	0.69	0.73
weighted avg	0.87	0.88	0.86

📊 Cross-Validation Results (Success Model):

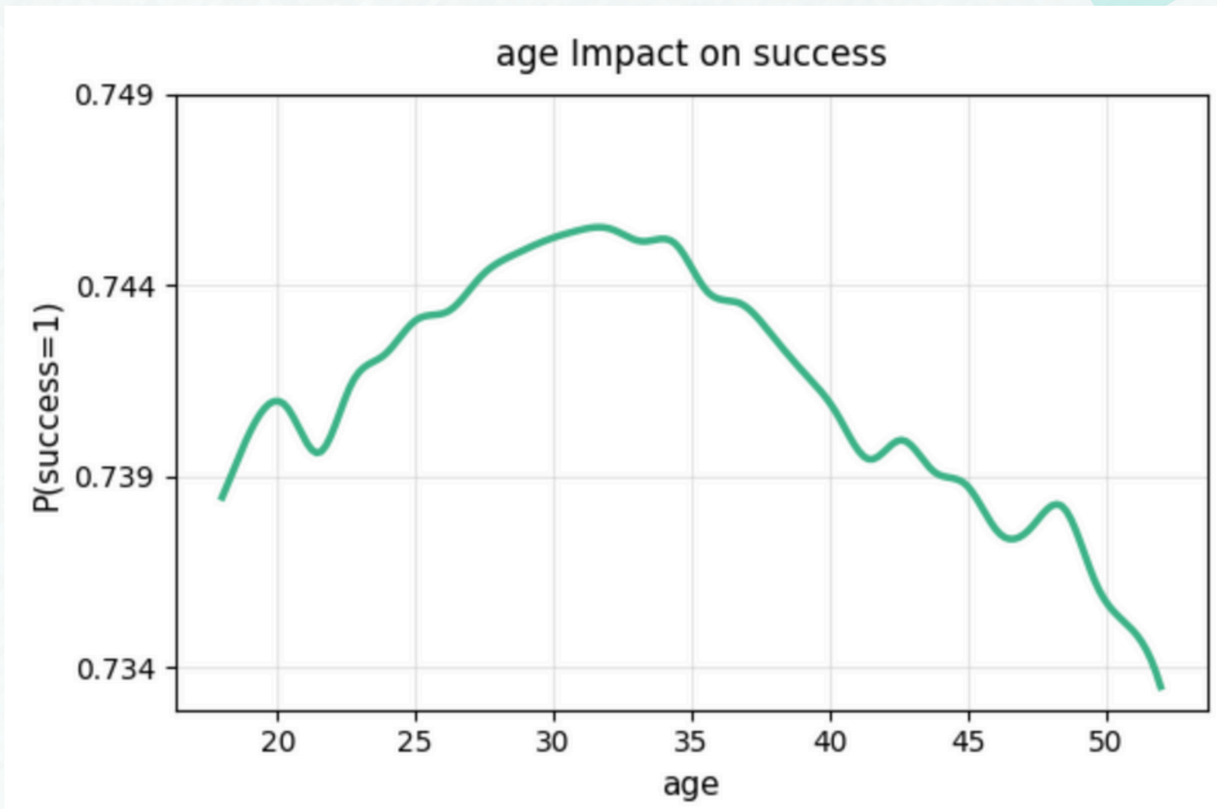
Accuracy:	0.876 (±0.006)		
Precision:	0.890 (±0.006)		
Recall:	0.971 (±0.006)		
F1:	0.929 (±0.004)		

Categorical Factors (Success) - Sorted by Impact

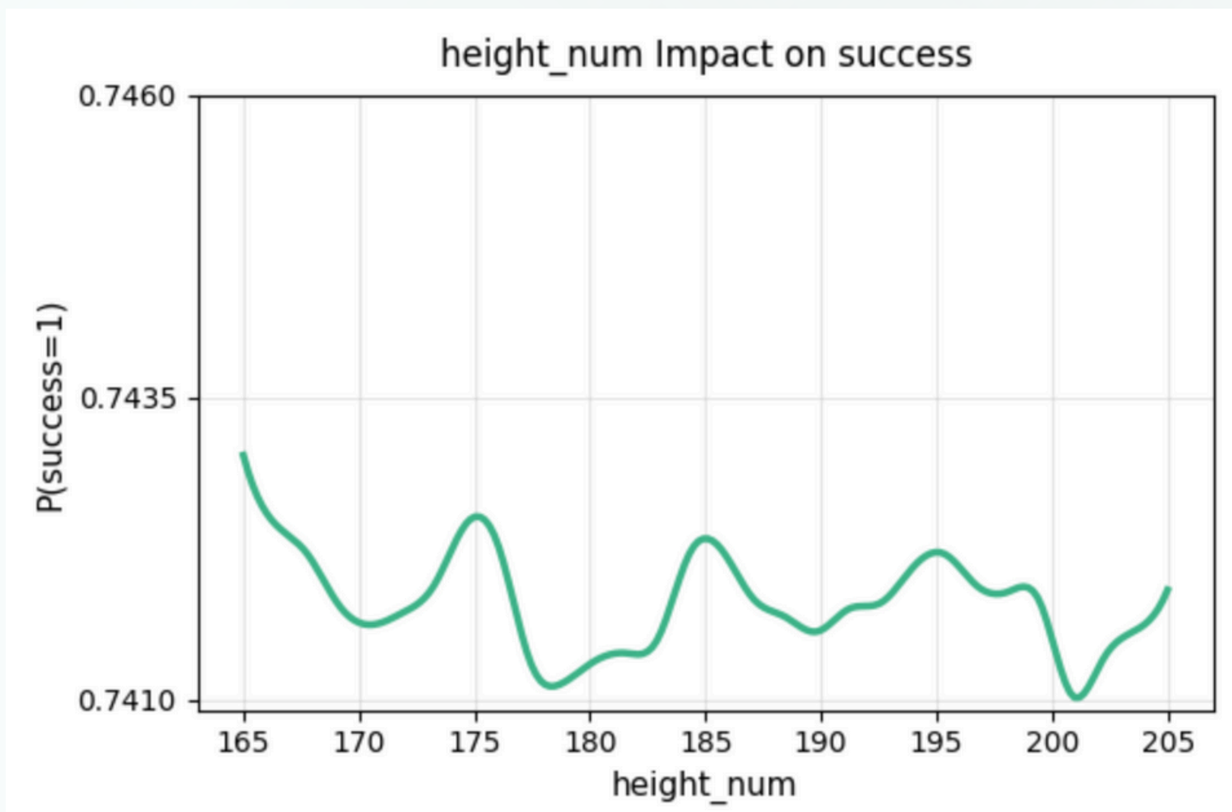
Factor	Value	P(success=1)	Δ from average	Direction
app_source	Bumble	0.767826	0.025009	↑ increases
sphere_of_employment	it / tech	0.76452	0.021703	↑ increases
app_source	Hinge	0.761179	0.018362	↑ increases
chat_channel	instagram dm	0.753122	0.010305	↑ increases
sphere_of_employment	healthcare / medicine	0.752993	0.010176	↑ increases
...				
body_type	overweight	0.721843	-0.020974	↓ decreases
app_source	other	0.720973	-0.021844	↓ decreases
sphere_of_employment	engineering / manufacturing	0.720666	-0.022151	↓ decreases
chat_channel	telegram	0.720466	-0.022351	↓ decreases
app_source	tinder	0.715764	-0.027053	↓ decreases

GitHub Repository link: https://github.com/dbeliaev-tum/match-gpt.git

Numerical Factors (Success)



Given that the model was primarily trained on data from the 20-30 age range, it is natural that the highest potential for successful relationships is with partners aged 25-35 years.
The age group over 40 is the least likely to build successful relationships when the age difference is approximately two-fold.



No clear trend was identified regarding the influence of height on relationship success, which was unsurprising. Successful relationships can be built with a partner of any height.

Application

```
# USER PROFILE INPUT SECTION
# Fill in the characteristics of your potential match below
# Replace the example values with your actual data for accurate predictions

# Basic Demographics
age = 27                                # Integer: 18-99 (e.g., 25, 30)
height = '180-190'                      # String: Height range in cm
                                        (e.g., '170-180')
body_type = 'athletic'                  # String: Physical build
                                        (slim, athletic, muscular, etc.)
region_origin = 'MiddleEast'            # String: Geographic origin
                                        (MiddleEast, Asia, etc.)

# Professional & Platform Information
sphere_employment = 'it / tech'         # String: Employment sector
                                        (healthcare / medicine, etc.)
app_source = 'tinder'                   # String: Dating app used (tinder, grindr, etc.)
chat_channel = 'instagram dm'           # String: Communication platform
                                        (whatsapp, snapchat, etc.)
activity = 'active'                     # String: Activity level (active, not active)

# RELATIONSHIP COMPATIBILITY METRICS (For success prediction)
# These metrics are typically rated after initial interaction on a 0-5 scale:
# 0 = No sex (applicable for sex_quality only),
# 1 = Poor, 2 = Below average, 3 = Average, 4 = Good, 5 = Excellent

sex_quality = 3                         # Float: Sexual compatibility rating (0-5)
emotional_comfort = 3                   # Float: Emotional connection comfort (1-5)
chemistry = 3                           # Float: Overall chemistry and spark (1-5)
communication_style = 3                 # Float: Communication style match (1-5)
```

To receive a prediction, simply provide the data of your potential match. The system will analyze the information through its dual-stage pipeline and return clear probability scores for both meeting likelihood and relationship success.

Outcome

📊 PREDICTION RESULTS:
Meeting probability: 73.5%
Relationship success probability: 81.5%

Code Principles

Clean & Modular Architecture

- Separation of data processing, model training, and prediction logic
- Reusable functions for preprocessing and feature engineering
- Configurable parameters for easy experimentation and tuning

Production-Ready Implementation

- Comprehensive error handling and data validation
- Model serialization using joblib for deployment
- Automated pipeline from raw data to predictions
- Compatibility with real-time and batch processing

Performance Optimization

- Efficient memory management with pandas operations
- Optimized hyperparameters for Random Forest classifiers
- Stratified sampling for representative data splits
- SMOTE integration for handling class imbalance

Reproducible Results

- Fixed random seeds for consistent model training
- Version control for all data processing steps
- Complete logging of preprocessing transformations
- Cross-validation with consistent fold stratification

Maintainable & Scalable

- Clear documentation and inline comments
- Modular design allowing easy model upgrades
- Standardized input/output interfaces
- Support for incremental model retraining

These principles ensure robust performance, easy maintenance, and seamless integration into existing dating platform infrastructures while maintaining high prediction accuracy.

Tech Stack & Validation

Core Technologies

- Python 3.x - Base programming language
- Scikit-learn - Machine learning models and preprocessing
- Imbalanced-learn - SMOTE for class imbalance handling
- Joblib - Model serialization and deployment

ML Framework

- Algorithm
  - Random Forest Classifier
- Preprocessing
  - One-Hot Encoding, feature engineering
- Imbalance Handling
  - SMOTE oversampling technique
- Model Persistence
  - Pickle format for production deployment

Validation Methodology

- Stratified Train-Test Split
- 5-Fold Cross-Validation with stratification
- Comprehensive Metrics: Accuracy, Precision, Recall, F1-Score
- Feature Importance Analysis for model interpretability

Performance Assurance

- Cross-validation consistency across all folds
- Balanced metrics for all classes
- Feature importance alignment with domain knowledge
- Robustness testing with edge cases and missing data

linkedin.com/in/denis-beliaev

Prospects

Implementation Simplicity

- Minimal Setup
  - Pure Python environment with standard libraries
- Clear Documentation
- Modular Design
- Standardized Inputs
  - Simple CSV format for data integration

Future Development Roadmap

- API development for seamless platform integration
- Mobile application with personalized user insights
- Advanced compatibility metrics
- A/B testing framework for continuous model improvement
- Real-time adaptive learning from user outcomes

Scalability Path

Containerized deployment options

Cloud service readiness

Horizontal scaling support for large user bases

Batch processing capabilities for mass predictions

Conclusion

Transformative Approach to Matchmaking
MatchGPT introduces a paradigm shift from simple profile matching to comprehensive relationship outcome prediction. It reduces wasting of time on unprospective communication.

Proven Business Value
The system delivers immediate practical benefits through reduced user churn, enhanced platform engagement, and data-driven matching optimization. Feature importance analytics enable targeted improvements in user experience and algorithm performance.

Ready for Implementation
With production-ready architecture and comprehensive validation, MatchGPT offers seamless integration into existing dating platforms. The modular design ensures easy maintenance and straightforward future enhancements.

Thanks for your attention

linkedin.com/in/denis-beliaev
instagram.com/geekkon_x
Belyaevda.gk@gmail.com

GitHub Repository link: https://github.com/dbeliaev-tum/match-gpt.git

linkedin.com/in/denis-beliaev