News Recommender System

Artificial Intelligence & Computer Science – User Experience Design (AICS-UX-Design)

Edited by:
1. Apostolou Athanasios (mpsp 2203)
2. Birmpakos Georgios (mpsp 2220)
3. Evangelou Alexandros – Ioannis (mpsp 2210)

CONTENTS

1. Introduction – Scope Of The App

Our app is designed as a personalized news recommendation system that offers an extensive scope for enhancing the news consumption experience of users. Here, we delve deeper into the key aspects of its scope:

Personalized News Feed:

- The app's primary objective is to deliver a personalized news feed tailored to each user's unique preferences.
- User preferences encompass a range of factors, including chosen topics of interest, preferred news sources, and individual reading habits.
- By analyzing these preferences, the app ensures that users receive a curated selection of news articles, videos, and multimedia content that resonates with their interests.

Content Curation:

- Content curation stands at the heart of this app's functionality, offering users a well-rounded and relevant news experience.
- The app aggregates content from an array of news sources, ensuring a diverse yet coherent selection of articles and multimedia.
- Accurate recommendations are paramount for user engagement and satisfaction. The app continually refines its curation through user interactions, feedback, and machine learning algorithms.

Machine Learning and AI:

- Leveraging advanced machine learning and artificial intelligence techniques, the app constantly enhances its content recommendation system.
- User behavior and engagement patterns are meticulously tracked and analyzed to adapt recommendations over time.
- The app employs AI algorithms to detect emerging trends and evolving user interests.

User Profiles:

- Users can create personalized profiles, where they specify their preferences, favorite news sources, and tolerance for different biases in news content.
- These profiles serve as the foundation for delivering highly tailored news feeds, fostering a sense of ownership over one's news consumption.

Data Privacy and Security:

- Paramount to the app's success is its commitment to stringent data privacy and security measures.
- Users can trust that their personal preferences and data are handled responsibly and securely, fostering confidence in the platform.

2. Technologies, Libraries, Frameworks and APIs

2.1 Front-End Development - Anguar

Angular is a powerful and popular open-source framework for building dynamic web applications. Developed and maintained by Google, it has gained widespread adoption in the world of front-end development. In this brief presentation, we'll explore the key features and advantages of Angular.

Key Features:

1. **Declarative UI:** Angular uses declarative templates, making it easier to understand and manage the user interface of your web applications. HTML is extended with Angular directives to express the application's components.
2. **Component-Based Architecture:** Angular encourages a modular structure through components. Each component encapsulates a part of the user interface and logic, promoting code reusability and maintainability.
3. **Two-Way Data Binding:** Angular offers two-way data binding, allowing seamless synchronization between the model and the view. When the model changes, the view updates, and vice versa.
4. **Dependency Injection:** Agular’s built-in dependency injection system simplifies the management of component dependencies, making it easier to maintain and test your code.
5. **Routing and Navigation:** Agular’s powerful router enables the creation of single-page applications with deep linking. It offers robust support for navigation and lazy loading of modules.
6. **Reactive Programming:** Angular leverages RxJS, a powerful library for reactive programming. This facilitates the handling of asynchronous operations, such as HTTP requests, with ease.

Advantages:

1. **Productivity:** Angular's toolset, including Angular CLI, simplifies project setup, development, and testing. It streamlines common tasks, enabling developers to be more productive.
2. **Large Ecosystem:** Angular boasts a large and active ecosystem. It has a rich library of third-party modules and extensive community support.
3. **Maintainability:** With its strong emphasis on modular development, Angular applications are typically easier to maintain and scale as they grow.
4. **Performance:** Angular's ahead-of-time (AOT) compilation and tree-shaking techniques help deliver optimized and performant applications.
5. **Cross-Platform:** Angular can be used to build web applications as well as native mobile applications using technologies like NativeScript and Ionic.

Conclusion:

Angular is a comprehensive front-end framework that empowers developers to build robust, maintainable, and high-performing web applications. Its extensive feature set, strong community support, and Google's backing make it a compelling choice for modern web development projects.

Whether you are developing a small website or a complex enterprise application, Angular provides the tools and structure needed to create dynamic, responsive, and feature-rich user interfaces.

2.2 Back-End Development – Python (FastAPI, Poetry, SQL Alchemy)

- FastAPI

FastAPI is a cutting-edge, open-source web framework for building APIs with Python. It has gained rapid popularity among developers due to its high performance, ease of use, and modern features. In this brief presentation, we'll explore the key features and advantages of FastAPI.

Key Features:

1. **Pythonic:** FastAPI embraces Python's simplicity and expressiveness. Developers can use Python type hints for request and response data, making the code intuitive and self-documenting.
2. **Automatic Documentation:** FastAPI generates interactive, Swagger-compatible documentation automatically from the code. This feature reduces the need for separate documentation efforts.
3. **Async and Await:** FastAPI fully supports asynchronous programming with Python's async and await keywords. This enables high-concurrency and non-blocking I/O, ideal for handling heavy workloads.
4. **Data Validation:** FastAPI performs automatic data validation and serialization, reducing the chances of data-related errors. It raises clear and informative errors when input data doesn't match the expected format.
5. **Dependency Injection:** FastAPI includes a robust dependency injection system, simplifying the management of dependencies and promoting clean, testable code.
6. **Security:** The framework provides built-in security features such as input data validation, protection against common web vulnerabilities, and OAuth2 support for authentication and authorization.

Advantages:

1. **Performance:** FastAPI's asynchronous design and efficient request handling make it incredibly fast, making it a perfect choice for high-performance APIs.
2. **Productivity:** The framework's automatic documentation generation, data validation, and simplified syntax boost developer productivity.
3. **Easy Learning Curve:** Developers familiar with Python can quickly grasp FastAPI's concepts and start building APIs without a steep learning curve.
4. **Ecosystem:** FastAPI integrates seamlessly with other Python libraries and tools, allowing developers to leverage the broader Python ecosystem.
5. **Community:** The framework has a growing and active community that contributes to its development, documentation, and third-party extensions.

Conclusion:

FastAPI is a game-changer for API development in Python. Its combination of high performance, automatic documentation, data validation, and async capabilities makes it an ideal choice for building modern web APIs.

Whether you are creating RESTful APIs, microservices, or backend services, FastAPI empowers you to deliver robust, efficient, and well-documented APIs with ease. It's a framework that can boost your development productivity while ensuring your applications meet the demands of the modern web.

- Poetry

Poetry is a modern and user-friendly tool for managing Python dependencies and packaging Python projects. It offers a streamlined approach to building, publishing, and sharing Python packages. In this brief presentation, we'll explore the key features and advantages of Poetry.

Key Features:

1. **Dependency Management:** Poetry simplifies Python dependency management by using a single, user-friendly pyproject.toml file. It manages project dependencies, dev-dependencies, and their versions effortlessly.
2. **Dependency Resolution:** Poetry uses a robust dependency resolver to ensure consistent and reliable resolution of package dependencies, avoiding version conflicts.
3. **Virtual Environments:** Poetry automatically creates isolated virtual environments for your projects, ensuring clean and reproducible installations of dependencies.
4. **Package Building:** Poetry streamlines the process of building Python packages. It generates distributable packages, such as wheel and source distributions, with a single command.
5. **Publishing:** Poetry provides straightforward publishing to the Python Package Index (PyPI) and other repositories. It handles package versioning and distribution efficiently.
6. **Lock Files:** Poetry generates a poetry.lock file, guaranteeing deterministic installations by locking dependency versions.

Advantages:

1. **Simplicity:** Poetry's user-friendly configuration file and CLI commands make it easy for developers to manage dependencies and package their projects.
2. **Reproducibility:** With isolated virtual environments and lock files, Poetry ensures that project dependencies are consistent across different environments, enhancing project reproducibility.
3. **Modern Workflow:** Poetry promotes a modern development workflow, aligning with best practices for Python packaging and distribution.
4. **Community Support:** Poetry has gained traction within the Python community, with active development and an engaged user base.
5. **Extensibility:** Developers can extend Poetry's functionality through plugins, allowing customization to meet specific project needs.

Conclusion:

Poetry is a valuable addition to the Python ecosystem, simplifying dependency management and package distribution. Its focus on transparency, reproducibility, and ease of use makes it an ideal choice for Python developers and project maintainers.

Whether you are working on a small script or a complex Python package, Poetry empowers you to manage dependencies and package your projects efficiently. It's a tool that aligns with modern Python development practices and contributes to a more enjoyable and productive development experience.

- SQL Alchemy

SQLAlchemy is a robust and popular Python library that simplifies database interaction and management. It serves as both a SQL toolkit and an Object-Relational Mapping (ORM) framework. In this brief presentation, we'll explore the key features and advantages of SQLAlchemy.

Key Features:

1. **Database Abstraction:** SQLAlchemy abstracts the differences between various database systems, allowing developers to work with databases in a consistent and database-agnostic manner.
2. **SQL Expression Language:** SQLAlchemy provides a powerful SQL expression language that allows developers to construct complex SQL queries in Python code, enhancing code readability and maintainability.
3. **ORM Framework:** SQLAlchemy's ORM allows developers to map Python classes to database tables, enabling object-oriented database interactions. This simplifies database operations by using Python objects instead of writing raw SQL queries.
4. **Connection Pooling:** SQLAlchemy includes built-in connection pooling, optimizing database connections for improved performance and resource management.
5. **Flexibility:** Developers have the flexibility to choose between SQLAlchemy's core SQL toolkit for custom SQL queries and the ORM for high-level, Pythonic database interactions.
6. **Query Composition:** SQLAlchemy supports query composition, allowing developers to build complex queries incrementally and efficiently.

Advantages:

1. **Database Agnosticism:** SQLAlchemy abstracts the underlying database, making it easy to switch between database systems without rewriting code.
2. **ORM Benefits:** The ORM simplifies database operations, reducing the need for writing complex SQL queries. It also promotes code reusability and maintainability.
3. **Security:** SQLAlchemy includes features for protecting against SQL injection and other security vulnerabilities, enhancing application security.
4. **Community and Ecosystem:** SQLAlchemy has a large and active community, with extensive documentation and third-party extensions that enhance its functionality.

5. **Performance:** The combination of connection pooling and query optimization makes SQLAlchemy performant, even in high-demand applications.

Conclusion:

SQLAlchemy is an indispensable tool for Python developers working with databases. Its dual functionality as a SQL toolkit and ORM framework makes it versatile and adaptable to various database-related tasks.

Whether you are building a web application, a data-driven API, or a complex data processing pipeline, SQLAlchemy streamlines database interactions, promotes code quality, and ensures compatibility with multiple database systems. It's a must-have library for any Python project that involves database operations.

2.3 Database – MySQL

MySQL is an industry-leading open-source relational database management system (RDBMS) known for its speed, reliability, and scalability. It powers countless applications and websites worldwide. In this brief presentation, we'll explore the key features and advantages of MySQL.

Key Features:

1. **Relational Database:** MySQL follows a relational model, allowing data to be stored in structured tables with defined relationships. It supports SQL (Structured Query Language) for data manipulation.
2. **Speed and Performance:** MySQL is optimized for speed, making it suitable for high-demand applications. It can handle complex queries and large datasets efficiently.
3. **Scalability:** MySQL offers various scaling options, including replication, clustering, and partitioning, to accommodate growing data needs and high traffic loads.
4. **Data Security:** MySQL provides robust security features, including user authentication, access control, encryption, and auditing, to protect sensitive data.
5. **ACID Compliance:** MySQL ensures data integrity and consistency through ACID (Atomicity, Consistency, Isolation, Durability) compliance, which is crucial for transactional applications.
6. **Community and Support:** MySQL has a vibrant community of developers, a wealth of online resources, and commercial support options, making it accessible and well-supported.

Advantages:

1. **Open Source:** MySQL is open source and free to use, making it cost-effective for startups and enterprises alike.
2. **Cross-Platform:** It is available for various operating systems, including Windows, Linux, macOS, and more, ensuring compatibility with different environments.
3. **High Availability:** MySQL supports replication and clustering for high availability and failover, reducing downtime and ensuring continuous operation.
4. **Ease of Use:** It has a user-friendly command-line interface and graphical tools like phpMyAdmin for easy database administration.

5. **Integration:** MySQL integrates seamlessly with popular programming languages and web frameworks, making it versatile for different development needs.

Conclusion:

MySQL is a trusted and widely adopted RDBMS that underpins a vast array of applications, from small-scale projects to large enterprises. Its combination of speed, scalability, security, and an active community makes it an excellent choice for managing relational data.

Whether you are developing a web application, an e-commerce platform, a content management system, or a data analytics solution, MySQL offers the reliability and performance needed to handle your data with confidence. It continues to be a cornerstone of modern database technology.

2.4 TensorFlow

TensorFlow is a powerful open-source machine learning framework developed by Google. It has become a leading choice for building, training, and deploying machine learning models. In this brief presentation, we'll explore the key features and advantages of TensorFlow.

Key Features:

1. **Deep Learning:** TensorFlow excels in deep learning tasks, allowing developers to create complex neural networks for tasks such as image recognition, natural language processing, and reinforcement learning.
2. **Flexibility:** TensorFlow provides a flexible framework for designing and implementing machine learning models, making it suitable for a wide range of applications and research.
3. **Keras Integration:** TensorFlow includes Keras, a high-level neural networks API, which simplifies model building and prototyping while maintaining TensorFlow's power and flexibility.
4. **TensorBoard:** TensorFlow comes with TensorBoard, a visualization tool that helps users understand and optimize their models through real-time visualization of training metrics.
5. **Deployment Options:** TensorFlow offers various deployment options, including TensorFlow Serving for serving models in production, TensorFlow Lite for mobile and embedded devices, and TensorFlow.js for web applications.
6. **Community and Ecosystem:** TensorFlow has a thriving community and ecosystem with extensive documentation, tutorials, and a wide range of pre-trained models.

Advantages:

1. **Scalability:** TensorFlow is designed to scale from a single device to distributed clusters, making it suitable for both small-scale experiments and large-scale production deployments.

2. **Open Source:** TensorFlow is open source, fostering collaboration and innovation in the machine learning community. Developers can contribute to its development and customize it to their needs.
3. **Broad Language Support:** TensorFlow supports multiple programming languages, including Python, C++, and more, enabling developers to choose the language that best suits their project.
4. **Transfer Learning:** TensorFlow facilitates transfer learning, allowing developers to leverage pre-trained models and fine-tune them for specific tasks, saving time and resources.
5. **Model Hub:** TensorFlow Hub provides a repository of pre-trained models and modules, simplifying model reuse and integration into new projects.

Conclusion:

TensorFlow has revolutionized machine learning by providing a versatile framework for developing cutting-edge models. Whether you are a researcher, a data scientist, or an engineer, TensorFlow empowers you to build, train, and deploy machine learning solutions efficiently.

From image classification to natural language understanding and reinforcement learning, TensorFlow offers the tools and resources to tackle a wide range of AI challenges. It continues to lead the way in shaping the future of artificial intelligence and machine learning.

2.5 Application Programming Interface (API) – https://newsapi.org

News API is a versatile and user-friendly platform that provides developers and businesses with easy access to a wealth of news data from around the world. In this brief presentation, we'll explore the key features and advantages of News API.

Key Features:

1. **Comprehensive News Coverage:** News API aggregates news articles, headlines, and sources from a wide range of reputable news outlets, ensuring access to diverse and up-to-date content.
2. **Developer-Friendly:** The API offers a straightforward and well-documented interface, making it easy for developers to integrate news data into their applications and services.
3. **Customizable Queries:** News API allows users to tailor their queries by specifying keywords, sources, languages, and more, ensuring highly targeted news retrieval.
4. **Real-Time Updates:** Users can access real-time news updates, enabling applications to provide the latest news as it happens.
5. **Categorized Content:** News articles are categorized by topics such as sports, technology, business, health, and more, facilitating content filtering and organization.
6. **Rich Metadata:** The API provides additional metadata such as publication dates, author names, and article summaries, enhancing the depth of information available.

Advantages:

1. **Versatility:** News API caters to a wide range of use cases, including news apps, research, content aggregation, and sentiment analysis.
2. **Global Reach:** It covers news from around the world, allowing users to access news from different regions and languages.
3. **Cost-Effective:** News API offers a range of pricing plans, including a free tier with limited requests, making it accessible to developers and startups.
4. **Integration:** It integrates seamlessly with various programming languages and frameworks, making it adaptable to different development environments.
5. **Community and Support:** News API has an active community and provides support to assist users in optimizing their news data integration.

Conclusion:

News API simplifies the process of accessing and integrating news data into applications and services. Whether you are building a news aggregator, enhancing content discovery, or conducting research, News API offers a convenient and cost-effective solution.

Its flexibility, extensive news coverage, and developer-friendly approach make it a valuable resource for businesses and developers seeking to leverage news data to enhance their applications and services.

2.6 Firebase Authentication

Firebase Authentication is a core component of the Firebase platform, providing developers with a powerful and user-friendly solution for authenticating users in mobile and web applications. In this brief presentation, we'll explore the key features and advantages of Firebase Authentication.

Key Features:

1. **Multiple Authentication Providers:** Firebase Authentication supports a wide range of authentication providers, including email/password, social identity providers (Google, Facebook, Twitter, etc.), phone number verification, and more.
2. **Security:** Firebase Authentication is designed with security in mind. It offers industry-standard practices for securely managing user identities and protecting sensitive data.
3. **Single Sign-On (SSO):** Users can sign in once and access multiple apps and services, enhancing the user experience and reducing friction.
4. **Custom Authentication System:** Developers have the flexibility to integrate their own authentication systems and user data while still benefiting from Firebase's infrastructure.
5. **Multi-Factor Authentication (MFA):** Firebase Authentication supports MFA to add an extra layer of security by requiring users to provide two or more verification factors before gaining access.
6. **User Management:** Developers can easily manage user accounts, reset passwords, and disable or block users when necessary, through the Firebase Console or Authentication API.

Advantages:

1. **Rapid Integration:** Firebase Authentication offers straightforward SDKs and libraries for various platforms, allowing developers to add authentication to their apps quickly.
2. **Scalability:** Firebase scales seamlessly to handle authentication demands, whether you're serving thousands or millions of users.
3. **Security Best Practices:** Firebase Authentication follows best practices for security, including secure token generation and storage, protecting against brute force attacks, and encryption of data in transit and at rest.
4. **Cross-Platform Compatibility:** Firebase Authentication supports a wide range of platforms, including web, iOS, Android, and server environments.
5. **Analytics Integration:** Authentication events are integrated with Firebase Analytics, providing insights into user sign-up, sign-in patterns, and conversion rates.

Conclusion:

Firebase Authentication is a dependable and versatile authentication solution for mobile and web applications. It simplifies the implementation of user authentication while maintaining the highest security standards. Whether you're building a social app, e-commerce platform, or enterprise solution, Firebase Authentication offers the tools to secure user data and provide a seamless sign-in experience.

By leveraging Firebase Authentication, developers can focus on creating engaging user experiences while ensuring user data remains protected and accessible only to authorized individuals.

3. User Interface Flow

3.1 Authentication

In our application, ensuring a smooth and secure user authentication process is paramount. This process not only guarantees a personalized experience but also safeguards user data:

Step 1: Firebase Authentication

To get started, users are required to complete Firebase authentication. This initial step establishes a secure connection between the user and your application. Firebase, a robust platform developed by Google, is renowned for its reliability and security.

Step 2: Existing Users

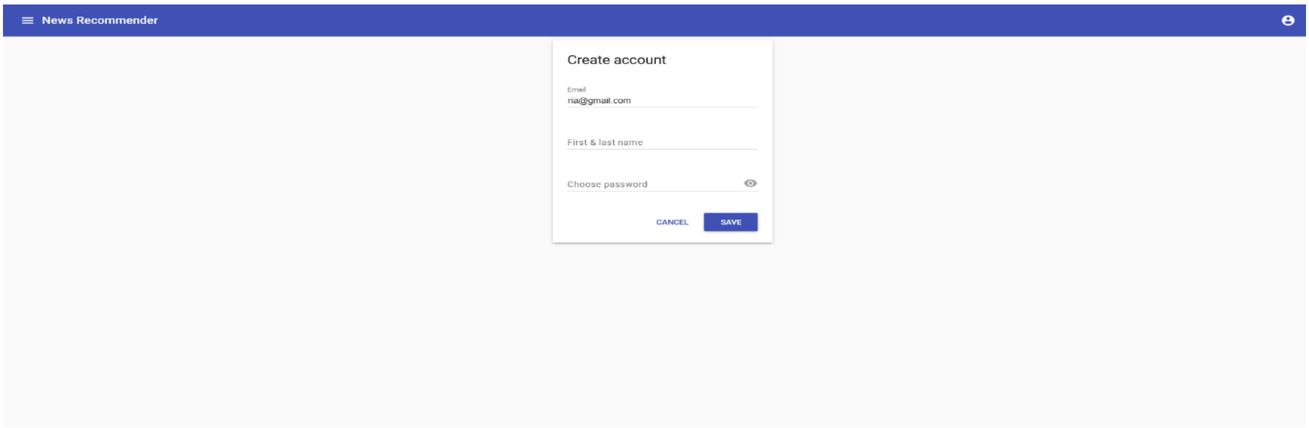
If a user already has an account, they can effortlessly access the app. There's no need to go through the sign-up process again, providing a hassle-free experience for returning users. This streamlined access encourages users to engage with your app regularly.

Step 3: New Users Sign Up

For new users, a straightforward sign-up process awaits. They need to provide their email and create a secure password. This information is essential for ensuring their identity and facilitating future access. User-friendly forms and clear instructions enhance the sign-up experience.

Step 4: Secure Credential Storage

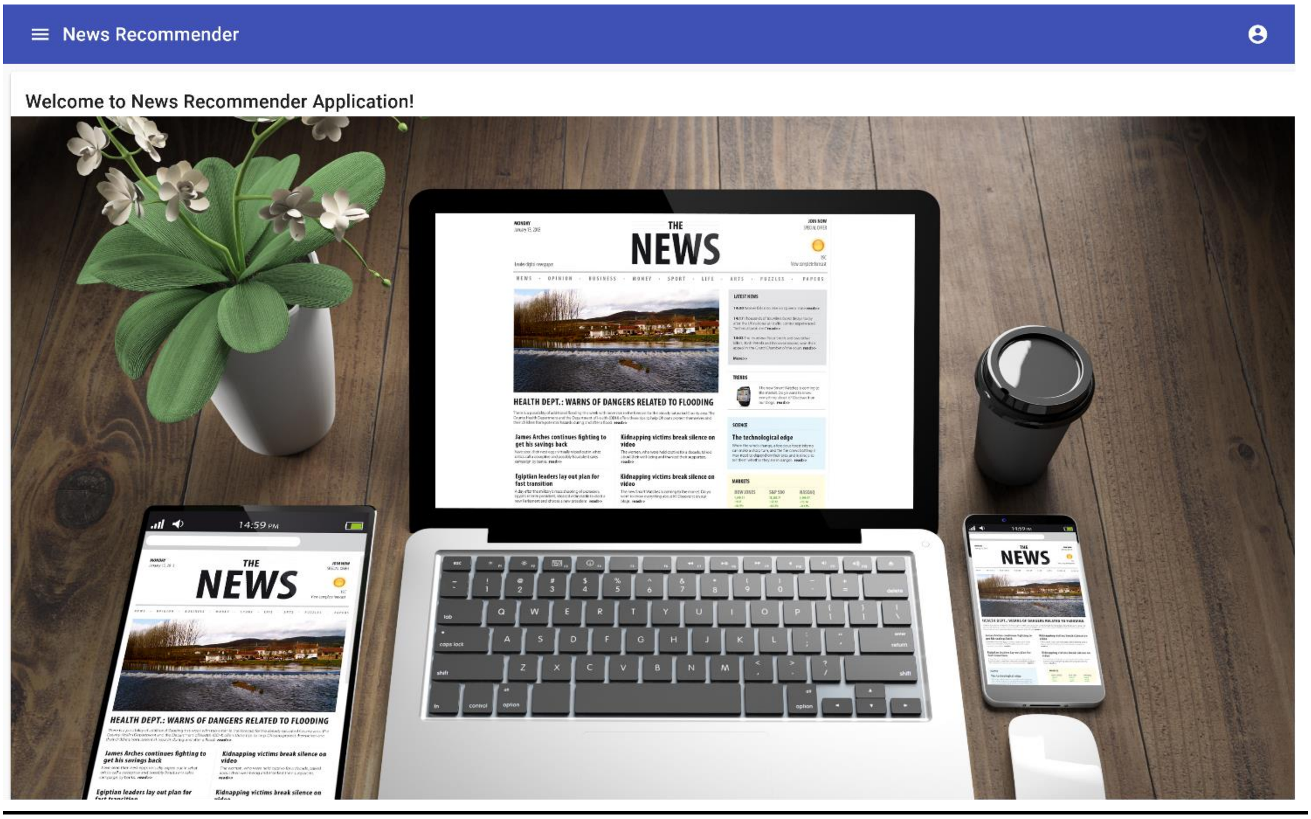
Once new users sign up, their credentials are securely stored in the MySQL database. MySQL, a trusted and scalable relational database management system, ensures the safekeeping of user data. Security measures such as encryption and hashing are applied to protect sensitive information.



Email
ria@gmail.com

News Recommender Firebase Authentication

3.2 Completing a Questionnaire



Welcome to News Recommender Application!

News Recommender Homepage

Understanding user preferences and interests enables the curation of a newsfeed that resonates with everyone.

Step 1: User Preferences

Users are given the opportunity to share their preferences through a set of thoughtfully crafted questions. These questions cover a range of categories, including:

1. **Favorite Category:** Users can specify their preferred news categories, ensuring they receive content aligned with their interests.
2. **Extrovert or Introvert:** Understanding users' personality traits can help in tailoring content that suits their preferences for social or solitary activities.
3. **Age:** Age can be a significant factor in content relevance. Users are encouraged to share their age for a more customized experience.
4. **Educational Level:** Educational background can influence the depth and complexity of news articles that users prefer.
5. **Sports Interest:** For sports enthusiasts, specifying their sports interests ensures a steady stream of sports-related news.

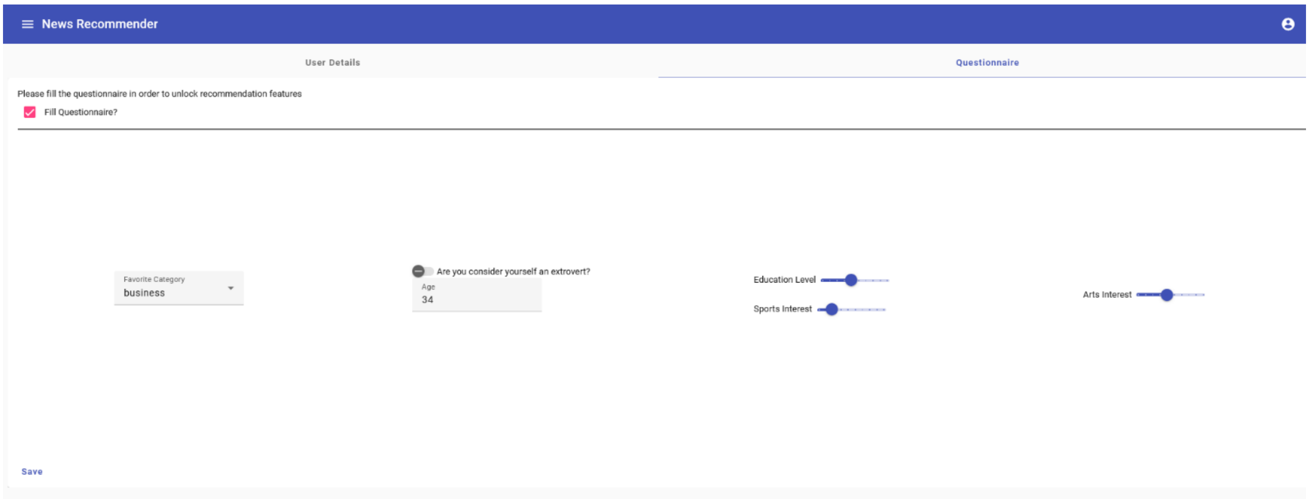
6. **Arts Interest:** Art aficionados can indicate their arts interests, leading to a more arts-centric news selection.

Step 2: Neural Network Adaptation

Once users provide their responses, a sophisticated neural network comes into play. This neural network, powered by state-of-the-art machine learning algorithms, considers the user's answers to the questions.

Step 3: Content Customization

Based on the neural network's analysis, news content is dynamically adjusted to match the user's preferences. This personalization ensures that each user's newsfeed reflects their unique tastes and interests.



Please fill the questionnaire in order to unlock recommendation features

☑ Fill Questionnaire?

Favorite Category: business

Are you consider yourself an extrovert?

Age: 34

Education Level

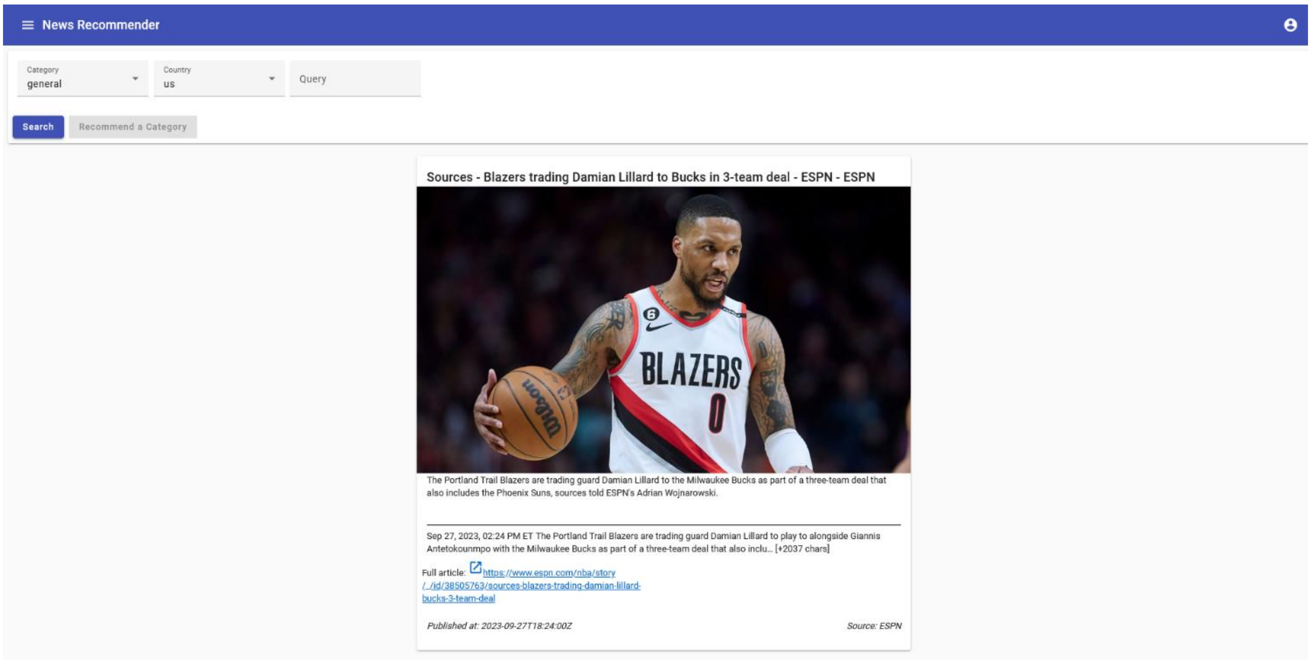
Sports Interest

Arts Interest

News Recommender Questionnaire

<u>3.3 Main News Page (No Questionnaire Completed)</u>

However, we understand that not all users may complete the questionnaire. In such cases, the news stream will continue but in a randomized fashion. This ensures that even users who haven't provided preferences still receive a diverse range of news articles, making the experience enjoyable and informative.



Sources - Blazers trading Damian Lillard to Bucks in 3-team deal - ESPN - ESPN

The Portland Trail Blazers are trading guard Damian Lillard to the Milwaukee Bucks as part of a three-team deal that also includes the Phoenix Suns, sources told ESPN's Adrian Wojnarowski.

Sep 27, 2023, 02:24 PM ET The Portland Trail Blazers are trading guard Damian Lillard to play to alongside Giannis Antetokounmpo with the Milwaukee Bucks as part of a three-team deal that also inclu… [+2037 chars]

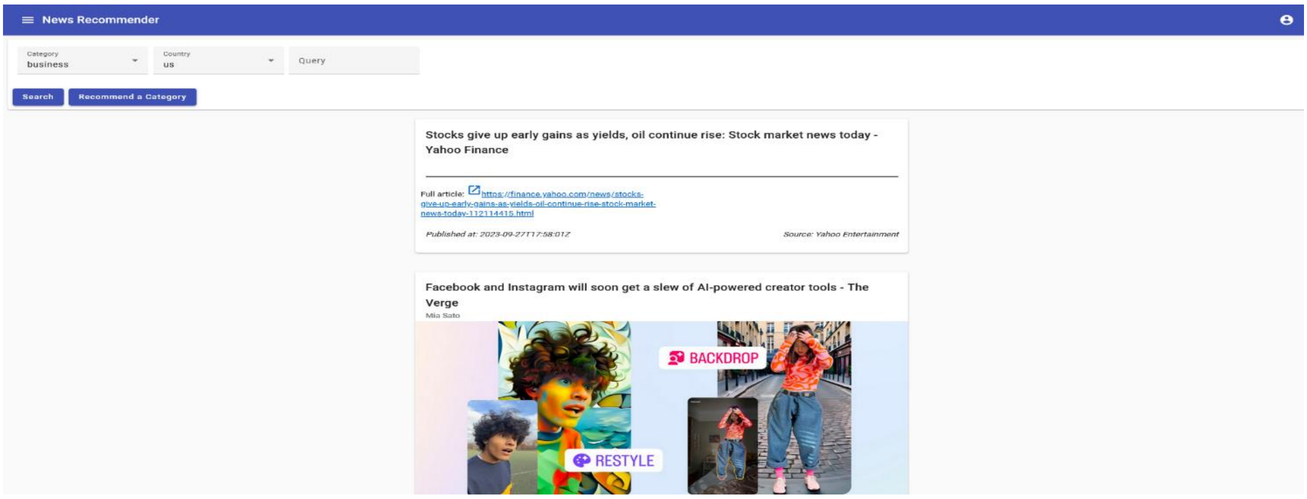
Full article: https://www.espn.com/nba/story/_/id/38505763/sources-blazers-trading-damian-lillard-bucks-3-team-deal

Published at: 2023-09-27T18:24:00Z *Source: ESPN*

News Recommender Questionnaire News (Questionnaire not completed)

3.4 Main News Page (Questionnaire Completed)

For users who have completed the questionnaire, their news stream is meticulously customized based on their preferences. This personalization ensures that each user's newsfeed is a tailored selection of articles that align with their unique tastes and interests.



Stocks give up early gains as yields, oil continue rise: Stock market news today - Yahoo Finance

Full article: https://finance.yahoo.com/news/stocks-give-up-early-gains-as-yields-oil-continue-rise-stock-market-news-today-112114415.html

Published at: 2023-09-27T17:58:01Z *Source: Yahoo Entertainment*

Facebook and Instagram will soon get a slew of AI-powered creator tools - The Verge

Mia Sato

News Recommender Questionnaire News (Questionnaire completed)

4. Neural Netwrok

4.1 Neural Network Description

For our recommendation system, we need to create a Neural Network. This neural network model should be able to analyze the data from the users' Questionnaires and recommend a News Category to a user.

We solve this as a Classification problem. The **FEATURES** will be the following 5 values from the questionnaire:

1. **Extrovert or Introvert**
2. **Age**
3. **Educational Level**
4. **Sports Interest**
5. **Arts Interest**

The **LABEL** will be the user **Favorite Category**.

Our model will be trained with data from existing users with filled questionnaires and recommend a category to a user who should have also filled his own questionnaire.

4.2 Neural Network Creation

The neural network is created by the system's administrator using our own command line utility. The administrator can create multiple neural network models, but only 1 can be active at each time.

The administration runs the command below in order to create an active neural network model, with name "default-nn".

```
poetry run python news_recommender_backend/main_cli.py nn create --name default-nn --active
```

We implement our model as a Tensorflow Sequential Model as shown in the code below:

```
def create_model() -> tf.keras.Sequential:
    metrics, _ = create_metrics()
    learning_rate: float = 0.01
    model = tf.keras.Sequential([
        tf.keras.layers.Flatten(input_shape=(5,)),
        tf.keras.layers.Dense(64, activation='relu'),
        tf.keras.layers.Dense(128, activation='relu'),
        tf.keras.layers.Dense(len(NewsCategoriesEnum), activation='softmax'),
    ])
```

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=learning_rate),

loss=tf.keras.losses.CategoricalCrossentropy(from_logits=False),
                  metrics=metrics)
    return model
```

Our model will have 2 middle layers with 64 and 128 neurons each. The final layer will have neurons equal to the length of our news categories. We use the **softmax** activation function so that the results are probabilities for which is the best suited News Category. The available News Categories in our used api are 7 and will be encoded as following:

1. **business**→ [1 0 0 0 0 0 0]
2. **entertainment**→ [0 1 0 0 0 0 0]
3. **general**→ [0 0 1 0 0 0 0]
4. **health**→ [0 0 0 1 0 0 0]
5. **science**→ [0 0 0 0 1 0 0]
6. **sports**→ [0 0 0 0 0 1 0]
7. **technology**→ [0 0 0 0 0 0 1]

Our model will be trained using the CategoricalCrossentropy loss function.

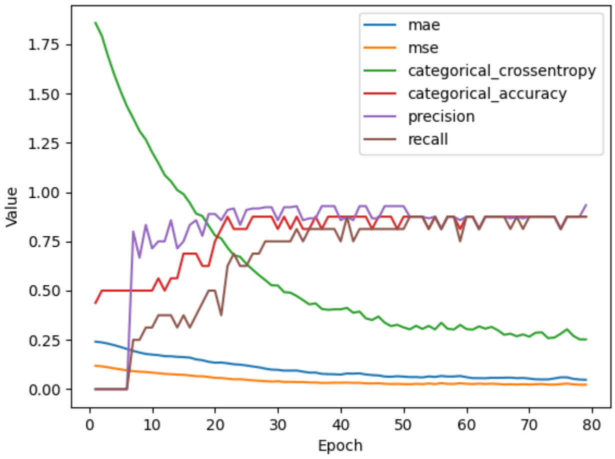
4.3 Neural Network Training

We fetch all the users with filled Questionnaires from our db. We then make 2 sets a TRAIN one (90% of data) and a TEST one (10%) of data. We then create the sets X_train containing the FEATURES data as described above and Y_train containing the LABLES data of the TRAIN set. Similarly we create the X_test and Y_test data from our TEST set.

We show below the basic code that performs the training:

```
def user_to_feature_list(user: User) -> list[float]:
    assert user.questionnaire is not None
    return [1 if user.questionnaire.is_extrovert else 0, user.questionnaire.age,
user.questionnaire.educational_level, user.questionnaire.sports_interest,
user.questionnaire.arts_interest]

def user_to_label_one_hot_vector(user: User) -> NDArray[float64]:
    assert user.questionnaire is not None
    one_hot_vector = np.zeros(len(NewsCategoriesEnum))
    for index, category in enumerate(NewsCategoriesEnum):
        logging.info(f"index,category={index},{category}")
        if category == user.questionnaire.favorite_category:
            one_hot_vector[index] = 1
    return one_hot_vector

def create_train_test_sets(users: list[User]) -> tuple[NDArray, NDArray, NDArray,
NDArray]:
    X_list = list(map(user_to_feature_list, users))
    X = np.array(X_list, dtype=float64)
    X = preprocessing.normalize(X, axis=0)
    Y_list = list(map(user_to_label_one_hot_vector, users))
    Y = np.array(Y_list, dtype=int64)
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.10,
random_state=42)
    return X_train, X_test, Y_train, Y_test


def train_model(mymodel: tf.keras.Sequential, metrics_names: list[str], epochs:
int, batch_size: int, train_features: NDArray, train_labels: NDArray, plot_file:
Path):
    history = mymodel.fit(train_features, train_labels,
                          epochs=epochs, batch_size=batch_size)
    # The list of epochs is stored separately from the rest of history.
    epochs = history.epoch
    # To track the progression of training, gather a snapshot
    # of the model's mean squared error at each epoch.
    hist = pd.DataFrame(history.history)
    plot_curves(epochs, hist, metrics_names, plot_file)

def evaluate_model(mymodel: tf.keras.Sequential, batch_size: int, test_features:
NDArray, test_labels: NDArray):
    evaluation = mymodel.evaluate(test_features, test_labels,
batch_size=batch_size)
    logging.info('evaluation')
    return evaluation

def train_evaluate_model(mymodel: tf.keras.Sequential, users: list[User],
plot_file: Path) -> tf.keras.Sequential:
    epochs = 80
    batch_size = 10
    _, metrics_names = create_metrics()
    X_train, X_test, Y_train, Y_test = create_train_test_sets(users)
    train_model(mymodel, metrics_names, epochs, batch_size, X_train, Y_train,
plot_file)
    evaluation = evaluate_model(mymodel, batch_size, X_test, Y_test)
    return mymodel
```

We show below the metrics values for each epoch during our training:



In order to avoid overfitting, we use only 80 epochs when the CategoricalCrossentropy graph line begins to flatten.

We evaluate our model against our TEST sets:

mae	mse	categorical_crossentropy	categorical_accuracy	precision	recall
3.7786965370178223	0.2776724696159363	0.2316981852054596	3.7786965370178223	0.0	0.0

4.4 Neural Network Predictions

When the user clicks the “Recommend a Category” button, our backend first selects the active Neural Network from our db. It then uses the user’s questionnaire as input to the predictions of the model. Because, as we have explained earlier, we use the **softmax** actiovation function, the result will be an array of 7 probabilities (each for every category). The one with the highest probability, is the recommended Category we show to the user!

We show the basic code of the prediction functionality below:

```
def predict_model(mymodel: tf.keras.Sequential, batch_size: int, features:
list[list[float]]):
    logging.debug("start predict_model")
    predictions = mymodel.predict(features, batch_size=batch_size)
    return predictions

def predict_model_for_user(mymodel: tf.keras.Sequential, user: User) ->
NewsCategoriesEnum:
    logging.debug("start predict_model")
    batch_size: int = 10
    user_features = user_to_feature_list(user)
    predictions = predict_model(mymodel, batch_size, [user_features])
    predicted_classes = np.argmax(predictions, axis = 1)
    predicted_class = predicted_classes[0]
    predicted_category: NewsCategoriesEnum | None = None
    for index, category in enumerate(NewsCategoriesEnum):
        if index == predicted_class:
            predicted_category = category

    assert predicted_category is not None
    return predicted_category
```

Our development system has limited number of users (about 20 to 30 users) so the results of the Neural Network are not very accurate. In a production system with thousands of users, the Neural Network will be trained much better and will be much more accurate.