AI Product Service Prototype Development and Business/Financial Modelling

Report By
Saniya Binth Zubair

ABSTRACT:
This report outlines the development of an AI-based fashion recommendation system prototype utilizing transfer learning with ResNet-50 and the Annoy library for KNN optimization. The system offers personalized fashion suggestions based on visual data analysis from over 30,000+ images. The purpose is to convert this prototype into a scalable business with a sustainable financial model. Key aspects such as feasibility, viability, and monetization are assessed to ensure success.

INTRODUCTION:
Artificial Intelligence (AI) is transforming industries with advanced tools like machine learning, deep learning, and transfer learning. In the fashion industry, the integration of AI-powered recommendation systems has the potential to revolutionize user experiences by delivering personalized suggestions. This report focuses on converting an AI-based fashion recommendation system prototype into a commercially viable product. The system leverages ResNet-50 for image feature extraction and KNN through Annoy for similarity searches.

PROBLEM STATEMENT:
The fashion industry faces the challenge of catering to individual style preferences while maintaining scalability. Many online platforms lack sophisticated recommendation systems that adapt to evolving customer needs, resulting in lower customer satisfaction and engagement. The proposed AI-driven fashion recommendation system addresses this gap by providing tailored recommendations, improving customer retention, and driving sales through a better user experience

AI Product Service Prototype Development
Step 1: Prototype Selection

Prototype Idea: **AI Fashion Recommendation System**

Feasibility: The system is based on proven AI technologies like ResNet-50 and KNN, which are mature and have existing implementations. It can be developed within 2-3 years, with improvements in accuracy and speed through hardware optimization and software refinement.

Viability: Fashion recommendation systems are highly relevant and can survive in the long term (20-30 years) due to the continuous growth of online shopping, e-commerce platforms, and the increasing demand for personalized user experiences.

Monetization: The system can be directly monetized through partnerships with e-commerce platforms, affiliate marketing, subscription models for premium features, and data-driven advertisements. Indirect monetization includes leveraging user data insights for trend prediction and inventory management.

AI Product/Service Description

Image Feature Extraction:
Using ResNet-50, the system extracts meaningful features from fashion images.
Similarity Search: Annoy is used for K-Nearest Neighbor (KNN) searches to find visually similar items.
User Interface:
A front-end application or API for e-commerce platforms to integrate into their product display.
Technology Stack:

- **AI Model:** ResNet-50 for transfer learning.
- **Database:** Scalable storage for the image dataset.
- **Backend:** Fast API or Flask for recommendation system requests.
- **Front-end:** React.js or Vue.js for user-facing platforms.

Financial Model

Cost Assumptions:
Development Costs:
AI Model Development (e.g., data preprocessing, model training): $50,000 - $100,000.
API/Interface Development: $30,000 - $60,000.
Cloud Infrastructure (GPU, data storage, bandwidth): $5,000 - $15,000 monthly.
Operational Costs:
Server Maintenance: $2,000/month.
Marketing/Advertising: $20,000 for the first launch phase.
Revenue Streams:
Subscription Model: Offer premium subscription for enhanced features or more frequent recommendations at $10/month per user.
Affiliate Marketing: Earn commissions (5-10%) through partnerships with fashion retailers.
Advertisements: Implement targeted ads in the recommendation feed, generating an estimated $0.50 per 1000 impressions.

Financial Projections

Year 1:

Expected Users: 100,000 users.

Revenue:

Subscription Model: $120,000 annually.
Affiliate Marketing: $50,000 annually.
Ads: $30,000 annually.

Total Revenue: $200,000.

Costs: $250,000 (development + marketing + infrastructure).

Net Profit: -$50,000 (loss due to initial setup costs).

Year 2:

Expected Users: 500,000 users.

Revenue:

Subscription Model: $600,000 annually.
Affiliate Marketing: $300,000 annually.
Ads: $150,000 annually.

Total Revenue: $1,050,000.

Costs: $300,000 (maintenance + marketing).

Net Profit: $750,000.

Long-Term Strategy

- **Expansion:** Incorporate multi-modal recommendation systems, including text and audio, expanding the product beyond fashion.
- **Data Monetization:** Offer anonymized user data insights to brands for trend forecasting and inventory management.
- **International Markets:** Launch the product in multiple regions, adapting to local fashion trends.

Step 2: Prototype Development

- To validate the feasibility of the AI-powered fashion recommendation system, a small-scale code implementation and model building exercise is conducted. This prototype aims to test the core functionality, specifically the ability to extract visual features from fashion images and provide accurate and relevant recommendations.

Feature Extraction:
ResNet-50 is used to extract deep visual features from a subset of 5,000 images taken from a public fashion dataset
Preprocessing includes resizing images and normalizing pixel values to ensure uniform input to the ResNet-50 model.

```
In [ ]: #Importing Libraries

In [1]: import numpy as np
        import pickle as pkl
        import tensorflow as tf
        from tensorflow.keras.applications.resnet50 import ResNet50,preprocess_input
        from tensorflow.keras.preprocessing import image
        from tensorflow.keras.layers import GlobalMaxPool2D
        from tqdm import tqdm
        from sklearn.neighbors import NearestNeighbors
        import os
        from numpy.linalg import norm

In [ ]: #Extract Filenames from Folder

In [2]: filenames = []
        for file in os.listdir('images'):
            filenames.append(os.path.join('images',file))
```


```
In [ ]: #Extract Filenames from Folder

In [2]: filenames = []
        for file in os.listdir('images'):
            filenames.append(os.path.join('images',file))
```


```
In [ ]: #Importing ResNet50 Model and Cofiguration

In [5]: base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224,224,3))
        base_model.trainable = False

        inputs = tf.keras.Input(shape=(224, 224, 3))
        x = base_model(inputs, training=False)
        x = GlobalMaxPool2D()(x)
        model = tf.keras.Model(inputs, x)


        model.summary()
```

Similarity Search:

The Annoy library is utilized to index the extracted image features. The K-Nearest Neighbor (KNN) algorithm is applied to find the top 5 visually similar items to a user's selected input image.

```
In [79]: from IPython.display import Image

In [80]: Image('1571.jpg')

Out[80]:
```

```
In [81]: Image(filenames[indices[0][1]])

Out[81]:
```

```
In [82]: Image(filenames[indices[0][2]])

Out[82]:
```

Validation:

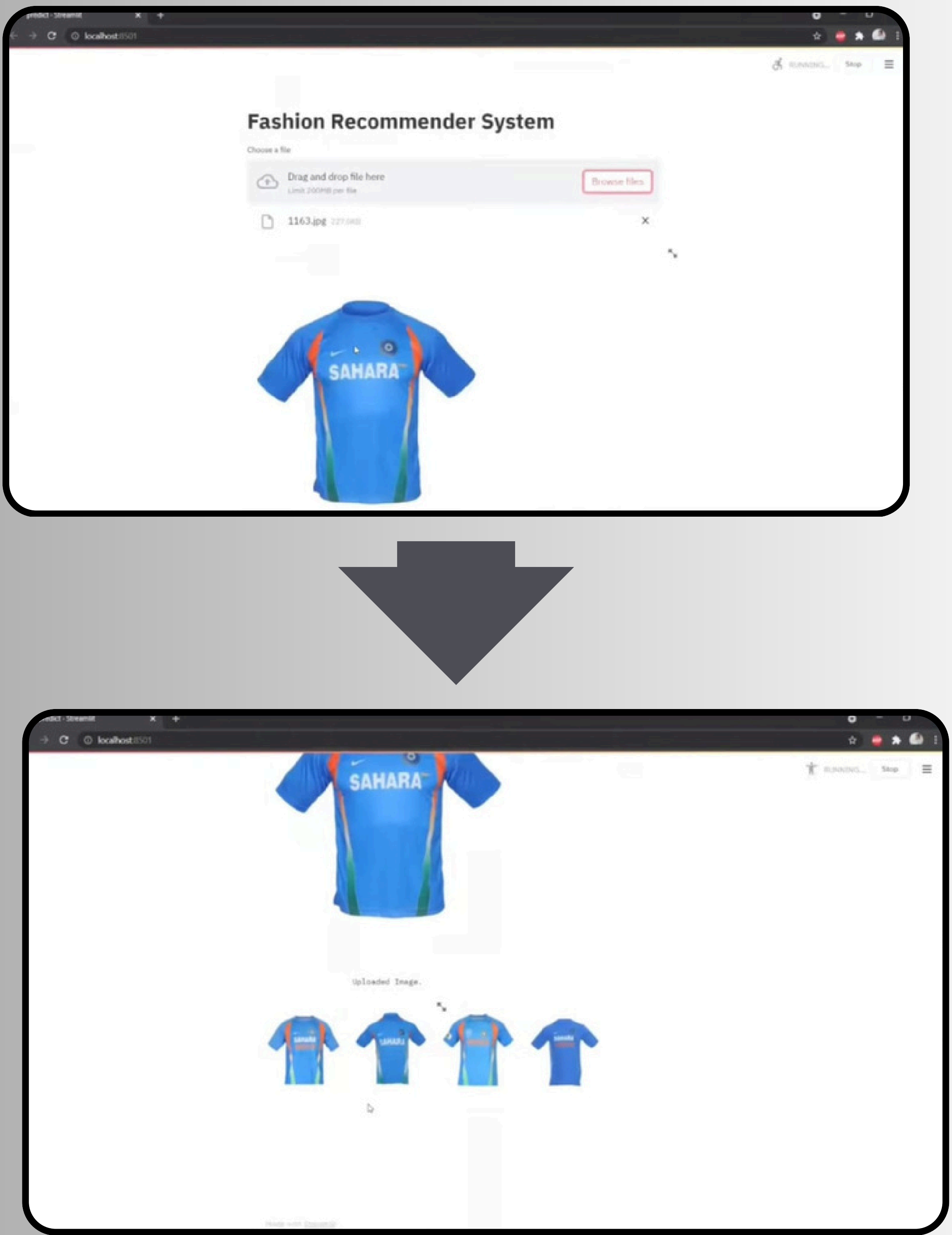
For validation, a small-scale test is performed with a dataset of 5,000 images. The system successfully identifies similar items within milliseconds, validating the functionality and efficiency of the combination of ResNet-50 and Annoy for image-based recommendation systems.

```
In [8]: def extract_features_from_images(image_path, model):
            img = image.load_img(image_path, target_size=(224,224))
            img_array = image.img_to_array(img)
            img_expand_dim = np.expand_dims(img_array, axis=0)
            img_preprocess = preprocess_input(img_expand_dim)
            result = model.predict(img_preprocess).flatten()
            norm_result = result/norm(result)
            return norm_result

In [9]: extract_features_from_images(filenames[0], model)

        1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 310ms/step

Out[9]: array([0.        , 0.01761617, 0.00171608, ..., 0.01247239, 0.02726404,
               0.06899218], dtype=float32)

In [10]: image_features = []
         for file in tqdm(filenames):
             image_features.append(extract_features_from_images(file, model))
         image_features

        1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 311ms/step
```

App/Website Prototype:

A simple Streamlit web based interface is developed (using Flask), where users can upload an image and receive top 5 fashion recommendations.

UI/UX: A minimal interface allowing users to interact with the recommendation system through an upload feature and a results page displaying the most similar items.



Step 3 : Business Modelling

Developing a robust business model for the AI-powered fashion recommendation system is essential for ensuring both profitability and scalability. The model needs to capture the operational structure, revenue generation strategies, and sustainability for long-term success.

Business Model Components

Target Market:

The system is designed for fashion e-commerce platforms, online retailers, and users looking for personalized fashion recommendations. Primary users include individuals who prefer customized shopping experiences and retailers aiming to enhance customer engagement.

Revenue Model: The following revenue streams will be integrated:

- **Subscription Model**: Offer premium services such as enhanced personalization, faster recommendations, or exclusive fashion insights through a subscription tier.
  Pricing: $10/month per user.
- **Affiliate Marketing:** Partnerships with fashion retailers, earning a commission on products sold through the recommendation system.
  Estimated Commission: 5-10% on sales.
- **Advertising:** Leverage targeted advertisements in the recommendation interface. Fashion brands can promote their products based on user preferences.
  Pricing: Estimated $0.50 per 1,000 impressions.

Value Proposition:

- For Consumers: Deliver tailored fashion recommendations to match individual styles, improving user satisfaction and engagement.
- For Retailers: Increased conversion rates through more relevant product displays, driving higher sales and better inventory management.

Cost Structure:

- **Development Costs:** Initial costs for AI model development, API integration, and web/app development. Estimated at $80,000 - $160,000.
- **Operational Costs:** Ongoing server and maintenance costs, customer support, and scaling infrastructure. Estimated at $10,000 - $20,000 annually.
- **Marketing Costs:** Initial marketing and advertising to promote the product. Estimated at $20,000 for the first year.
- **Personnel Costs:** Engineers, data scientists, and marketing teams.

Key Activities:
Building partnerships with online fashion platforms.
Scaling the system to handle millions of users and high-traffic volumes.
Regularly updating the AI models with new fashion trends and user data.
Key Partners:
E-commerce platforms like Shopify, WooCommerce, and large online fashion retailers.
Affiliate networks for partnerships and collaboration.
Customer Channels:
Integration with existing e-commerce platforms as a plugin or API.
Direct-to-consumer web applications or mobile apps.
Key Metrics:
User Acquisition Cost (UAC): How much it costs to acquire a paying user.
Customer Lifetime Value (CLV): Revenue a customer generates over their entire relationship with the system.
Conversion Rate: How effectively the recommendations lead to purchases.

Business Model Reference Links:

Business Plan Templates: https://create.microsoft.com/en-us/search
General Business Model Explanation:
https://www.investopedia.com/terms/b/businessmodel.asp#:~:text=For%20instance%2C%20direct%20sales%2C%20franchising,sporting%20organizations%20like%20the%20NBA.
Examples of Different Business Models: https://alcorfund.com/insight/18-business-model-example-explained/

Step 4: Financial Modelling (Equation) with Machine Learning & Data Analysis

a. Identifying the Market

The AI-powered fashion recommendation system will be launched in the online fashion retail market. This market is growing rapidly due to the increasing trend of online shopping and personalization in user experiences. Key markets include North America, Europe, and Asia-Pacific.

b. Collecting Market Data

Data can be collected from online sources such as:

- **Global Market Reports**: For growth rates, market size, and trends in the online fashion retail market.
- **E-commerce Statistics:** Statistics on consumer behavior, average order value, and growth in fashion e-commerce.

For example, the global fashion e-commerce market was valued at around $533 billion in 2023, with an expected compound annual growth rate (CAGR) of 9.4% from 2024 to 2030.

c. Market Forecasting with Machine Learning

To perform market forecasts and predictions, machine learning models such as regression or time series forecasting can be used. This step provides insights into future market trends and sales projections.

Steps for Forecasting:

- **Data Collection:** Gather past sales data, market growth rates, and trends.
- **Time Series Forecasting:**

Use methods like ARIMA (Auto-Regressive Integrated Moving Average) or Prophet to predict future sales or market size.
Time series analysis helps model seasonal variations in sales, providing more accurate projections.

Example:

Using historical sales data, you can apply time series models to predict how the online fashion retail market will grow. Let’s assume the sales data for the past 5 years indicates a steady annual growth of 12%. This trend can be projected forward to predict future sales.

Machine Learning Resources:

https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-to-time-series-analysis/

https://www.analyticsvidhya.com/blog/2021/10/machine-learning-for-stock-market-prediction-with-step-by-step-implementation/

d. Financial Equation Design

Once the market trends and sales predictions are modeled, you can design a simple financial equation for revenue and profit. This equation will consider the cost of running the business and the sales volume.

Example: Financial Equation
Let's assume the following:

- The product (AI fashion recommendation system) is priced at $50 per unit.
- The monthly operating cost (infrastructure, personnel, marketing, etc.) is $2,000.
- In the month of June, the company forecasts selling 500 units.
-

The total revenue for June can be calculated as:
Revenue for June = 50×500−2000=25,000−2,000=23,000

Now, assuming the sales number is x, the financial equation becomes:
y=50x−2000y = 50x - 2000y=50x−2000

Where:

- y is the total revenue
- x is the total number of sales in a given month

Financial Projection Using Market Forecasting:
If the regression or time series model predicts a 10% increase in sales every month, you can adjust the equation accordingly.

For example, starting with 500 units sold in June, and assuming a 10% increase per month, the equation for the next month (July) becomes:

xjuly = 500 x (1+0.10) = 550
Revenue for July = 50 x 550 - 2000 = 27,500 - 2,000 = 25,500

Thus, you can forecast revenue for the coming months based on the predicted sales growth.

Conclusion

By applying machine learning models like regression and time series forecasting, you can predict future sales and market trends. This data is then incorporated into a simple financial equation that links revenue to the sales number, helping you model the business's profitability over time.

GIthub link:

https://github.com/Saniya-BZ/Feynn_labs-final_project