ericrono /
Product-Cross-selling-Recommender-System

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights

0 stars · 2 forks · 1 watching · Branches · Activity
Tags

Public repository

2 Branches · 0 Tags · Go to file t · Go to file · + · Add file · Code

Mosota-Kemunto-Gladys-2020 Update 'tfidf.pkl' · 2fdaf81 · 6 minutes ago

DATA	Repo folders	3 days ago
IMAGES	Repo folders	3 days ago
WORK	Move 'Investment Product Cross Selli...	21 minutes ago
.gitignore	updated ipynb document	last week
Investment Product Cross Selli...	final.ipynb file'	13 minutes ago
Investment_Product_Cross_Sel...	Update notebook.pkl'	6 minutes ago
Presentation.pdf	Add new files and update existing on...	14 hours ago
Presentation.pptx	Add new files and update existing on...	14 hours ago
README.md	Added visualizations on to README	last week
main_interface.py	Update main_interface.py	2 days ago
model.pkl	Update 'model.pkl'	1 hour ago
notebook.pdf	Add notebook.pdf	19 minutes ago
tfidf.pkl	Update 'tfidf.pkl'	6 minutes ago

README

Financial Investment Product Recommender-System

Overview

This project aims to develop a recommendation system to enhance product offerings for both new and existing customers. For existing customers, the system focuses on cross-selling and upselling by utilizing data on their geographic location, demographics, and investment behaviors to recommend products that match their profile. For new customers, recommendations are tailored to their specific needs, including factors such as risk appetite, investment duration, and investment amount. The recommendation engine is designed to be user-friendly and highly effective in identifying investment products that best align with each customer's unique characteristics.

Table of Contents

1. Getting Started

This section sets up the environment, loads necessary libraries, and prepares the workspace for analysis. Libraries used here are for data manipulation,numerical operations, data visualization and Modelling.

2. Data Loading and Exploration

Here, the notebook loads the customer profile and investment product datasets, performs an initial examination of their structure, and displays key attributes. The focus is on understanding the available fields, which include:

- **Investor Age**: Age of the investor.
- **Town**: Investor's town or location.
- **Beneficiary Age**: Age of any associated beneficiaries.
- **Product Portfolio**: Information about available investment products.

Key steps:

- Load the data using `pd.read_csv` or similar methods.
- Display the first few rows using `head()` to get an overview.
- Check for missing values and datatype consistency.

3. Data Preprocessing

This section cleans and prepares the data for further analysis. Typical preprocessing steps include:

- **Handling Missing Values**: Filling or dropping missing values to ensure data consistency.
- **Dropping Duplicate Rows**: Dropped replicated rows within our dataset which reduced the number of rows from about 7.5 million to 120,000.
- **Mapped Columns**: Mapped Relationship,Gender and Portfolio Columns.

- **Calculated Ages**: Using Members and Beneficiary Date of Birth we calculated their current age.

4. Exploratory Data Analysis (EDA)

In the EDA section, the notebook examines relationships within the data, especially focusing on:

- **Age Distributions**: Age distribution of investors and beneficiaries. alt text
- **Town Distribution**: Count of investors per town. alt text
- **Investment Product Preferences**: Analysis of the most popular products by age group. alt text

The notebook uses bar plots, histograms, and scatter plots to visualize these relationships, providing insights into potential cross-selling opportunities based on demographic patterns.

5. Feature Engineering

The notebook creates new features that may improve the recommendation model, such as:

- **Age Grouping**: Categorizing investors into age groups (e.g., young, middle-aged, senior).
- **Beneficiary Status**: Creating a binary variable indicating whether an investor has a beneficiary.
- **Regional Grouping**: If there are multiple towns, grouping them by similar regions might improve recommendation specificity.

6. Recommendation System

This is the core section where the recommendation system is designed. The system is composed of:

- **Rule-Based Recommendations**: Based on predefined rules, specific products are suggested to certain age groups.
- **Pattern-Based Recommendations**: Analyzing past investment choices to make data-driven recommendations.

The notebook also implements:

- **K-Nearest Neighbors (KNN)** to suggest products similar to the investor's previous choices.
- **Rule-based Logic**: Mapping specific age ranges to recommended product types.

Each recommendation function is thoroughly documented, explaining input parameters and the logic behind suggested products.

7. Evaluation

This section evaluates the recommendation system's performance, using Accuracy Metric.

Evaluation also included testing our model using existing customers data and sample customers.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 6

Languages

- HTML 59.4%
- Jupyter Notebook 39.8%
- Python 0.8%

Suggested workflows

Based on your tech stack

Jekyll using Docker image

Configure

Package a Jekyll site using the jekyll/builder Docker image.

SLSA Generic generator

Configure

Generate SLSA3 provenance for your existing release workflows

Python Package using Anaconda

Configure

Create and test a Python package on multiple Python versions using Anaconda for package management.

More workflows

Dismiss suggestions