import UIKit

struct Movie {

let id: Int

let title: String

let genres: [String]

let embedding: [Double] // Вектор эмбеддинга сюжета+жанра

}

class MovieRecommender {

let movies: [Movie]

init(movies: [Movie]) {

self.movies = movies

}

// Косинусное сходство между двумя векторами

func cosineSimilarity(\_ v1: [Double], \_ v2: [Double]) -> Double {

guard v1.count == v2.count else { return 0 }

var dot = 0.0

var mag1 = 0.0

var mag2 = 0.0

for i in 0..<v1.count {

dot += v1[i] \* v2[i]

mag1 += v1[i] \* v1[i]

mag2 += v2[i] \* v2[i]

}

let denominator = sqrt(mag1) \* sqrt(mag2)

return denominator == 0 ? 0 : dot / denominator

}

// Рекомендуем N похожих фильмов

func recommend(for movie: Movie, topN: Int = 5) -> [Movie] {

let filtered = movies.filter { $0.id != movie.id }

let scored = filtered.map { candidate -> (Movie, Double) in

(candidate, cosineSimilarity(movie.embedding, candidate.embedding))

}

.sorted { $0.1 > $1.1 }

return Array(scored.prefix(topN).map { $0.0 })

}

}

class MoviesListViewController: UITableViewController {

let recommender: MovieRecommender

let movies: [Movie]

init(movies: [Movie]) {

self.movies = movies

self.recommender = MovieRecommender(movies: movies)

super.init(style: .plain)

self.title = "Фильмы"

}

required init?(coder: NSCoder) { fatalError("init(coder:) не используется") }

override func viewDidLoad() {

super.viewDidLoad()

tableView.register(UITableViewCell.self, forCellReuseIdentifier: "Cell")

}

// MARK: - TableView DataSource

override func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

movies.count

}

override func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)

let movie = movies[indexPath.row]

cell.textLabel?.text = movie.title

return cell

}

// MARK: - TableView Delegate

override func tableView(\_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {

let selectedMovie = movies[indexPath.row]

let recommendations = recommender.recommend(for: selectedMovie)

let detailVC = MovieDetailViewController(movie: selectedMovie, recommendations: recommendations)

navigationController?.pushViewController(detailVC, animated: true)

}

}

class MovieDetailViewController: UIViewController, UITableViewDataSource {

let movie: Movie

let recommendations: [Movie]

private let titleLabel = UILabel()

private let genresLabel = UILabel()

private let tableView = UITableView()

init(movie: Movie, recommendations: [Movie]) {

self.movie = movie

self.recommendations = recommendations

super.init(nibName: nil, bundle: nil)

self.title = movie.title

}

required init?(coder: NSCoder) { fatalError("init(coder:) не используется") }

override func viewDidLoad() {

super.viewDidLoad()

view.backgroundColor = .white

titleLabel.text = movie.title

titleLabel.font = UIFont.systemFont(ofSize: 24, weight: .bold)

titleLabel.translatesAutoresizingMaskIntoConstraints = false

genresLabel.text = "Жанры: " + movie.genres.joined(separator: ", ")

genresLabel.font = UIFont.systemFont(ofSize: 16)

genresLabel.translatesAutoresizingMaskIntoConstraints = false

tableView.translatesAutoresizingMaskIntoConstraints = false

tableView.dataSource = self

tableView.register(UITableViewCell.self, forCellReuseIdentifier: "RecCell")

view.addSubview(titleLabel)

view.addSubview(genresLabel)

view.addSubview(tableView)

NSLayoutConstraint.activate([

titleLabel.topAnchor.constraint(equalTo: view.safeAreaLayoutGuide.topAnchor, constant: 12),

titleLabel.leadingAnchor.constraint(equalTo: view.leadingAnchor, constant: 16),

titleLabel.trailingAnchor.constraint(equalTo: view.trailingAnchor, constant: -16),

genresLabel.topAnchor.constraint(equalTo: titleLabel.bottomAnchor, constant: 8),

genresLabel.leadingAnchor.constraint(equalTo: titleLabel.leadingAnchor),

genresLabel.trailingAnchor.constraint(equalTo: titleLabel.trailingAnchor),

tableView.topAnchor.constraint(equalTo: genresLabel.bottomAnchor, constant: 16),

tableView.leadingAnchor.constraint(equalTo: view.leadingAnchor),

tableView.trailingAnchor.constraint(equalTo: view.trailingAnchor),

tableView.bottomAnchor.constraint(equalTo: view.bottomAnchor)

])

}

func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

recommendations.count

}

func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

let cell = tableView.dequeueReusableCell(withIdentifier: "RecCell", for: indexPath)

let movie = recommendations[indexPath.row]

cell.textLabel?.text = movie.title

return cell

}

}

// MARK: - AppDelegate и SceneDelegate для запуска без Storyboard

import SwiftUI

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?

let sampleMovies: [Movie] = [

Movie(id: 1, title: "Фильм А", genres: ["Драма", "Приключения"], embedding: [0.8, 0.1, 0.0, 0.9]),

Movie(id: 2, title: "Фильм Б", genres: ["Драма", "Романтика"], embedding: [0.7, 0.2, 0.1, 0.6]),

Movie(id: 3, title: "Фильм В", genres: ["Комедия"], embedding: [0.1, 0.9, 0.7, 0.0]),

Movie(id: 4, title: "Фильм Г", genres: ["Драма", "Приключения"], embedding: [0.85, 0.05, 0.0, 0.88]),

Movie(id: 5, title: "Фильм Д", genres: ["Приключения", "Фэнтези"], embedding: [0.9, 0.0, 0.3, 0.85])

]

func application(\_ application: UIApplication,

didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

window = UIWindow(frame: UIScreen.main.bounds)

let navController = UINavigationController(rootViewController: MoviesListViewController(movies: sampleMovies))

window?.rootViewController = navController

window?.makeKeyAndVisible()

return true

}

}