Write logic for this:

1. A retail chain wants to forecast product sales using factors like holidays, promotions, location, and inventory. The dataset has 80 features. How should they select the most impactful ones?

Answer:

1. Perform correlation analysis and remove features with very low correlation to the sales target.
2. Use Recursive Feature Elimination (RFE) with a regression model to rank features.
3. Apply regularization methods like Lasso Regression to penalize less important variables.
4. Validate results using cross-validation and retain only the features that consistently contribute to high R^2 scores.

2.A podcast platform wants to recommend new podcasts to users based on the categories and hosts they follow. How should they design this content-based system?

Answer:

1. Convert podcast metadata (categories, hosts, length) into a feature vector using TF-IDF or OneHotEncoding.
2. Represent each user by averaging the vectors of podcasts they've listened to.
3. Use cosine similarity between user vectors and available podcasts to rank and recommend.
4. Filter out already played episodes to avoid repetition.

3.An industrial firm wants to predict machine failure using sensor readings and system logs (200+ features). How do they choose which inputs to use?

Answer:

1. Remove features with constant values or high missing values.
2. Use mutual information to evaluate feature relevance with the failure label.

3. Apply tree-based models (e.g., Random Forest) to get feature importances.
4. Select top-ranked features that contribute most to reducing entropy in the target.

---

4. A fashion app wants to suggest outfits to users based on their previous likes, color preferences, and seasonal trends. How should they implement the recommendation?

Answer:

1. Encode outfit attributes (style, color, brand, season) using embedding or OneHotEncoding.
2. Track user interactions (likes, views) and assign weights.
3. Create a user profile vector and compute similarity with new outfit vectors.
4. 4.Rank and recommend top N items using cosine similarity or ANN search.

---

5. A research lab wants to predict heart disease using patient data. The dataset has lab results, habits, and vitals. How should they select only the most relevant features?

Answer:

1. Perform exploratory data analysis to remove irrelevant or redundant columns.
2. Use SelectKBest with f_classif to rank features based on ANOVA scores.
3. Cross-validate with Logistic Regression and Decision Tree to validate feature importance.
4. Keep the top 10-15 features with the highest predictive power.

---

6. A job portal wants to recommend job listings based on a user's resume and previous applications. How can they build a recommendation system?

Answer:

1. Convert resumes and job descriptions to vector form using TF-IDF or BERT embeddings.
2. Compare user profile vector with job vectors using cosine similarity.
3. Include application history and feedback to refine recommendations.
4. Regularly update the model as users apply to new roles.

---

7. A smart home system wants to predict energy usage using readings from multiple sensors. With 100+ features, how do they reduce complexity?

Answer:

1. Use PCA (Principal Component Analysis) to reduce dimensionality while retaining variance.
2. Analyze variance ratios to determine the number of components to keep.
3. Optionally, use tree-based feature importance scores for interpretation.
4. Train regression models with reduced features and compare RMSE.

---

8. An e-learning platform wants to suggest tutorials based on a learner's skill gaps identified from quizzes. How should they build this system?

Answer:

1. Map tutorials to skills using tags.
2. Identify skills the learner struggles with based on quiz results.
3. Recommend tutorials matching the weak areas using content filtering.
4. Use engagement data to personalize further.

9.A social media app wants to predict post engagement (likes/comments) using post features, timing, and user activity. How do they select the right features?

Answer:

1. Use feature importance from Gradient Boosting to find the top contributors.
2. Drop highly correlated or low-variance features.
3. Use backward feature elimination with cross-validation.
4. Keep features that generalize well and avoid overfitting.

---

10. Now you want to create an app named attendance in the project and register it.write logic for this Django project

Answer:

Step 1: python manage.py startapp attendance
Step 2: Add 'attendance' to INSTALLED_APPS in settings.py

✅ Step-by-Step Explanation:

1. Open the terminal and ensure you're inside the project folder.
2. Run: python manage.py startapp attendance
3. Open student_portal/settings.py
4. In INSTALLED_APPS, add this line:
   → 'attendance',
