**Chinook Data Analysis & Predictive Modeling Report**

**1. Introduction**

The Chinook dataset is a sample database representing a digital media store. It contains information about customers, invoices, invoice lines, music tracks, employees, and other entities. This project aimed to explore, analyze, and derive business insights using data science techniques, progressing through three levels of depth:

* Level 1: Descriptive and Explanatory
* Level 2: Analytical and Pattern-Based Insights
* Level 3: Predictive Modeling

This report summarizes findings from each level, covering the process, code, visualizations, and interpretations.

**2. Data Overview**

Data was extracted from the SQLite Chinook database. Key tables used include:

* **Customer**
* **Invoice**
* **InvoiceLine**
* **Track**
* **Genre**
* **Employee**
* **Album**

Data cleaning included:

* Converting date columns to datetime format
* Merging tables using appropriate keys
* Removing duplicates and unnecessary columns
* Handling missing values

**Code Snippet:**

invoice\_df['InvoiceDate'] = pd.to\_datetime(invoice\_df['InvoiceDate'])

data = invoice\_df.merge(customer\_df, on='CustomerId', how='left')

data.drop\_duplicates(inplace=True)

**3. Level 1: Descriptive and Explanatory Analysis**

**Objectives:**

* Understand customer and sales distribution
* Summarize key metrics

**Key Metrics:**

* Total Revenue
* Number of Invoices
* Average Invoice Value
* Country-wise Revenue Distribution

**Visualizations:**

* [Insert Bar Chart: Number of Customers per Country]
* [Insert Pie Chart: Top Genres by Track Count]

**Insights:**

* The USA contributes the highest revenue.
* Rock is the most common genre.
* Most customers are concentrated in North America and Western Europe.

**4. Level 2: Analytical and Pattern-Based Insights**

**A. Customer Segmentation with RFM Analysis**

RFM (Recency, Frequency, Monetary) analysis was used to cluster customers into behavioral groups.

**Code Snippet:**

rfm['Recency'] = (reference\_date - rfm['LastPurchaseDate']).dt.days

rfm['Frequency'] = rfm['TotalPurchases']

rfm['Monetary'] = rfm['TotalSpent']

rfm['Cluster'] = KMeans(n\_clusters=4).fit\_predict(rfm[['Recency', 'Frequency', 'Monetary']])

**Visualizations:**

* [Insert RFM Cluster Plot]

**Insights:**

* High-value customers purchase frequently and recently.
* Some clusters indicate churn risk based on old activity.

**B. Product Affinity Analysis (Market Basket)**

**Code Snippet:**

frequent\_itemsets = apriori(track\_customer\_matrix, min\_support=0.01, use\_colnames=True)

rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1)

**Insights:**

* Certain genres and artists tend to be purchased together, helpful for marketing bundles.

**C. Track Length vs Sales Correlation**

**Code Snippet:**

correlation = track\_df[['Milliseconds', 'Quantity']].corr()

**Findings:**

* Weak correlation; longer tracks do not significantly influence sales.

**D. Employee Sales Performance**

**Code Snippet:**

employee\_sales = invoice\_df.groupby('SupportRepId')['Total'].sum().sort\_values(ascending=False)

**Insights:**

* A few employees outperform others consistently.

**5. Level 3: Predictive Modeling**

**A. Predicting Invoice Total (Regression)**

**Objective:** Predict total invoice amount using features like track count, country, and invoice date attributes.

**Model Used:** Linear Regression

**Code Snippet:**

X = data.drop('Total', axis=1)

y = data['Total']

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

**Performance:**

* R² Score: 0.95
* MAE: 0.56
* RMSE: 0.93

**Feature Importance:**

* Country and number of tracks are strong predictors of invoice total.

**B. Track Recommendation System**

**Method:** Item-based collaborative filtering using cosine similarity.

**Code Snippet:**

similarity\_matrix = cosine\_similarity(track\_customer\_matrix.T)

def get\_recommendations(track\_id):

similar\_scores = list(enumerate(similarity\_matrix[track\_id]))

similar\_scores = sorted(similar\_scores, key=lambda x: x[1], reverse=True)

return [i[0] for i in similar\_scores[1:6]]

**Insights:**

* The system suggests tracks based on past purchases, enabling personalization.

**6. Conclusion**

This project demonstrated a complete pipeline from exploration to modeling:

* Descriptive analysis offered a clear overview of business operations.
* Analytical insights enabled segmentation and strategy formulation.
* Predictive models provided tools for sales forecasting and personalization.

**7. Future Work**

* Integrate additional data sources (e.g., streaming data)
* Expand recommendation system to hybrid models
* Deploy models into a web dashboard using Flask or Streamlit

**Appendix (Code & Visualizations)**

Visualizations and code snippets referenced in each section can be inserted inline or included here as high-resolution figures and full-length scripts.



















