Customer Churn Prediction and Revenue Forecasting

This code covers multiple steps in analyzing customer and business data. Below is a breakdown of its components, which can help you create a PowerPoint presentation to explain each step:

Slide 1: Introduction

Title: *Customer and Business Data Analysis*

Key Points:

- Merging datasets for comprehensive analysis
- Cleaning and preprocessing data
- Performing machine learning tasks like classification and clustering
- Forecasting future trends with Prophet

Slide 2: Data Loading and Merging

Code Section:

```
Customers = pd.read_csv("/content/Customers.csv")

Subscriptions = pd.read_csv("/content/Subscription.csv")

Transactions = pd.read_csv("/content/Transcation.csv")

Usage = pd.read_csv("/content/Usage.csv")

merged_df = pd.merge(Customers, Subscriptions, on="CustomerID", how="left")

merged_df = merged_df.merge(Transactions, on="CustomerID", how="left")

merged_df = merged_df.merge(Usage, on="CustomerID", how="left")
```

Explanation:

- Imported necessary datasets: Customers, Subscriptions, Transactions, and Usage.
- Merged them on the CustomerID field using a left join to retain customer information even if other data is missing.

Slide 3: Handling Date Columns

Code Section:

```
merged_df["StartDate"] = pd.to_datetime(merged_df["StartDate"], errors="coerce")

merged_df["EndDate"] = pd.to_datetime(merged_df["EndDate"], errors="coerce")

merged_df["tenure"] = (merged_df["EndDate"] - merged_df["StartDate"]).dt.days
```

Explanation:

- Converted StartDate and EndDate to datetime objects.
- Calculated tenure as the difference in days between start and end dates.
- Ensured invalid dates were handled gracefully with errors="coerce".

Slide 4: Feature Engineering

Code Section:

```
merged_df["average_monthly_spend"] = merged_df["amount"] / (merged_df["tenure"] / 30.0)

merged_df["average_monthly_spend"].fillna(0, inplace=True)
```

Explanation:

- Created a new feature, average_monthly_spend, dividing total amount by tenure in months.
- Replaced NaN values with 0 to avoid issues in downstream tasks.

Slide 5: Data Visualization

Code Section:

```
plt.figure(figsize=(8, 5))

sns.histplot(merged_df["Age"], bins=10, kde=True)

plt.title("Age Distribution")

plt.show()
```

Visualization: A histogram showing the distribution of customers' ages.

Explanation:

- Used Seaborn's histplot to visualize the age distribution.
- Helped understand the demographic makeup of the customer base.

Slide 6: Encoding Categorical Data

Code Section:

```
from sklearn.preprocessing import LabelEncoder

merged_df["Gender"] = LabelEncoder().fit_transform(merged_df["Gender"])

merged_df["Location"] = LabelEncoder().fit_transform(merged_df["Location"])
```

Explanation:

- Used LabelEncoder to convert categorical fields like Gender and Location into numerical values.
- Prepared the data for machine learning tasks.

Slide 7: Churn Prediction with Random Forest

Code Section:

```
features = ["Age", "Gender", "Income", "tenure", "average_monthly_spend"]

X = merged_df[features]

y = (merged_df["Status"] == "Churned").astype(int)

model = RandomForestClassifier(random_state=42)

model.fit(X_train, y_train)

print(classification_report(y_test, predictions))
```

Explanation:

- Selected features and target variable (churned).
- Trained a Random Forest model to predict whether a customer would churn.
- Evaluated performance using a classification report.

Slide 8: Customer Segmentation with K-Means

Code Section:

```
kmeans = KMeans(n_clusters=3, random_state=42)

merged_df["segment"] = kmeans.fit_predict(X_imputed)
```

Explanation:

- Applied K-Means clustering to segment customers into three groups.
- Analyzed these segments to understand customer behavior patterns.

Slide 9: Revenue Forecasting with Prophet

Code Section:

```
revenue_data =
Transactions.groupby(Transactions["transaction_date"].dt.to_period("M"))["amount"].sum().r
eset_index()

model = Prophet()

model.fit(revenue_data)

future = model.make_future_dataframe(periods=12, freq="M")

forecast = model.predict(future)

model.plot(forecast)

plt.show()
```

Explanation:

- Used Facebook Prophet to forecast monthly revenue trends.
- Visualized the forecasted revenue for the next 12 months.

Slide 10: Saving Results

Code Section:

```
merged_df.to_csv("customer_analysis_results.csv", index=False)

forecast.to_csv("revenue_forecast.csv", index=False)
```

Explanation:

- Exported the results for further use or presentation.

Slide 11: Conclusion

Key Points:

- Analyzed customer data to understand churn, segmentation, and revenue trends.
- Combined machine learning models with visualization for actionable insights.
- Forecasted future revenue to support business planning.

Thank You