**Fraud Detection for Cross River Bank**

**Performing tasks on Structured Data(Customer, Loans, Transactions) using MySQL. We have created a schema, imported the dataset as table and formatted the datatypes, assigned the Primary Key, Foreign Key for tables. Then based on the tasks we have done the query on the tables data.**

|  |  |
| --- | --- |
| **Task 1** | **Customer Risk Analysis: Identify customers with low credit scores and high-risk loans to predict potential defaults and prioritize risk mitigation strategies.** |
| **Solution** | SELECT C.CUSTOMER\_ID,C.NAME, C.CREDIT\_SCORE  FROM CUSTOMER\_TABLE C  JOIN LOAN\_TABLE L  ON C.CUSTOMER\_ID = L.CUSTOMER\_ID  WHERE C.CREDIT\_SCORE < 650  AND L.DEFAULT\_RISK = 'High'  ORDER BY 2 ASC; |
| **Findings** | We took an threshold credit score of 650 and there are 605 customers present whose credit score is lower than threshold. So, giving loan to them is Highly risky. |
| **Recommendation** | Prioritize risk mitigation strategies for these customers by offering tailored financial products or reviewing loan terms to reduce potential defaults. |

|  |  |
| --- | --- |
| **Task 2** | **Loan Purpose Insights: Determine the most popular loan purposes and their associated revenues to align financial products with customer demands** |
| **Solution** | SELECT L.LOAN\_PURPOSE,SUM(T.TRANSACTION\_AMOUNT) AS REVENUE  FROM LOAN\_TABLE L  JOIN TRANSACTION\_TABLE T  ON L.LOAN\_ID = T.LOAN\_ID  WHERE L.LOAN\_STATUS IN ('Approved' , 'Closed')  GROUP BY L.LOAN\_PURPOSE  ORDER BY REVENUE DESC;  A screenshot of a computer  Description automatically generated |
| **Findings** | The most popular loan purposes are identified, along with their associated revenues, providing insights into customer preferences and revenue generation. |
| **Recommendation** | Align financial product offerings with the most popular loan purposes to optimize revenue and meet customer demand effectively. |

|  |  |
| --- | --- |
| **Task 3** | **High-Value Transactions: Detect transactions that exceed 30% of their respective loan amounts to flag potential fraudulent activities.** |
| **Solution** | SELECT T.TRANSACTION\_ID,T.TRANSACTION\_DATE,T.TRANSACTION\_AMOUNT, T.TRANSACTION\_TYPE  FROM TRANSACTION\_TABLE T  JOIN LOAN\_TABLE L  ON T.LOAN\_ID = L.LOAN\_ID  WHERE T.TRANSACTION\_AMOUNT > (L.LOAN\_AMOUNT \* 0.30); |
| **Findings** | Transactions exceeding 30% of their respective loan amounts are flagged, indicating potential instances of unusual or fraudulent activity. |
| **Recommendation** | Investigate flagged transactions for potential fraud and implement stricter monitoring and controls for high-value transactions. |

|  |  |
| --- | --- |
| **Task 4** | **Missed EMI Count: Analyze the number of missed EMIs per loan to identify loans at risk of default and suggest intervention strategies.** |
| **Solution** | SELECT C.CUSTOMER\_ID,L.LOAN\_ID,L.LOAN\_PURPOSE,L.LOAN\_AMOUNT,L.LOAN\_STATUS, C.NAME,COUNT(T.TRANSACTION\_ID) AS TOTAL\_MISSED\_EMI  FROM TRANSACTION\_TABLE T  JOIN LOAN\_TABLE L  ON T.LOAN\_ID = L.LOAN\_ID  JOIN CUSTOMER\_TABLE C  ON C.CUSTOMER\_ID = L.CUSTOMER\_ID  WHERE ((T.TRANSACTION\_TYPE= 'Missed EMI') OR (T.TRANSACTION\_TYPE = 'EMI Payment' AND T.STATUS = 'Failed'))  GROUP BY C.CUSTOMER\_ID,L.LOAN\_ID,L.LOAN\_PURPOSE,L.LOAN\_AMOUNT,L.LOAN\_STATUS,C.NAME  HAVING TOTAL\_MISSED\_EMI > 0  ORDER BY TOTAL\_MISSED\_EMI DESC;  A screenshot of a computer  Description automatically generated |
| **Findings** | Loans with missed or failed EMI payments are identified, highlighting customers who are at risk of default due to multiple missed payments. |
| **Recommendation** | Proactively engage with customers having missed EMIs to offer repayment restructuring or financial counseling to mitigate default risks. |

|  |  |
| --- | --- |
| **Task 5** | **Regional Loan Distribution: Examine the geographical distribution of loan disbursements to assess regional trends and business opportunities.** |
| **Solution** | SELECT SUBSTR(TRIM(SUBSTRING\_INDEX(C.ADDRESS, ',', -1)),1,2) AS REGION,  COUNT(L.LOAN\_ID) AS TOTAL\_LOANS,  SUM(L.LOAN\_AMOUNT) AS TOTAL\_LOAN\_AMOUNT,  AVG(L.LOAN\_AMOUNT) AS AVERAGE\_LOAN\_AMOUNT,  MAX(L.LOAN\_AMOUNT) AS MAX\_LOAN\_AMOUNT,  MIN(L.LOAN\_AMOUNT) AS MIN\_LOAN\_AMOUNT  FROM LOAN\_TABLE L  JOIN CUSTOMER\_TABLE C  ON C.CUSTOMER\_ID = L.CUSTOMER\_ID  GROUP BY REGION  ORDER BY TOTAL\_LOAN\_AMOUNT DESC; |
| **Findings** | The geographical distribution of loan disbursements reveals regional trends in loan volume and value, highlighting areas with higher loan activity. |
| **Recommendation** | Focus marketing and product development efforts in high-loan regions while exploring opportunities to increase loan disbursements in underperforming areas. |

|  |  |
| --- | --- |
| **Task 6** | **Loyal Customers: List customers who have been associated with Cross River Bank for over five years and evaluate their loan activity to design loyalty programs.** |
| **Solution** | SELECT C.CUSTOMER\_ID,C.NAME,C.AGE,C.INCOME,C.CREDIT\_SCORE,C.ADDRESS,  C.CUSTOMER\_SINCE,  TIMESTAMPDIFF(YEAR, STR\_TO\_DATE(C.CUSTOMER\_SINCE, '%m/%d/%Y'), CURDATE()) AS ASSOCIATED\_YEARS,  COUNT(L.LOAN\_ID) AS TOTAL\_LOANS,  SUM(L.LOAN\_AMOUNT) AS TOTAL\_LOAN\_AMOUNT,  AVG(L.LOAN\_AMOUNT) AS AVERAGE\_LOAN\_AMOUNT,  MAX(L.LOAN\_AMOUNT) AS MAX\_LOAN\_AMOUNT,  MIN(L.LOAN\_AMOUNT) AS MIN\_LOAN\_AMOUNT  FROM CUSTOMER\_TABLE C  JOIN LOAN\_TABLE L  ON C.CUSTOMER\_ID = L.CUSTOMER\_ID  WHERE TIMESTAMPDIFF(YEAR, STR\_TO\_DATE(C.CUSTOMER\_SINCE, '%m/%d/%Y'), CURDATE()) > 5  GROUP BY C.CUSTOMER\_ID, C.NAME, C.AGE, C.INCOME, C.CREDIT\_SCORE, C.ADDRESS, C.CUSTOMER\_SINCE, ASSOCIATED\_YEARS  ORDER BY ASSOCIATED\_YEARS DESC,TOTAL\_LOAN\_AMOUNT DESC; |
| **Findings** | Customers who have been with Cross River Bank for over five years are identified, along with their loan activity, showcasing their long-term engagement and financial contributions. |
| **Recommendation** | Design tailored loyalty programs to reward these long-term customers and incentivize continued business by offering exclusive benefits based on their loan activity. |

|  |  |
| --- | --- |
| **Task 7** | **High-Performing Loans: Identify loans with excellent repayment histories to refine lending policies and highlight successful products.** |
| **Solution** | SELECT L.LOAN\_ID,L.LOAN\_PURPOSE,  SUM(L.LOAN\_AMOUNT) AS TOTAL\_LOAN\_AMOUNT,  L.REPAYMENT\_HISTORY,L.LOAN\_STATUS  FROM LOAN\_TABLE L  WHERE L.REPAYMENT\_HISTORY >= 8  AND L.LOAN\_STATUS = 'Closed'  GROUP BY L.LOAN\_ID,L.LOAN\_PURPOSE,L.REPAYMENT\_HISTORY,L.LOAN\_STATUS  ORDER BY L.REPAYMENT\_HISTORY DESC; |
| **Findings** | Loans with excellent repayment histories (score ≥ 8) and a 'Closed' status indicate strong repayment patterns and successful product performance. |
| **Recommendation** | Focus on promoting loans with high repayment scores and 'Closed' status as best-performing products to refine future lending policies. |

|  |  |
| --- | --- |
| **Task 8** | **Age-Based Loan Analysis: Analyze loan amounts disbursed to customers of different age groups to design targeted financial products.** |
| **Solution** | SELECT  CASE  WHEN C.AGE BETWEEN 18 AND 25 THEN '18-25'  WHEN C.AGE BETWEEN 26 AND 35 THEN '26-35'  WHEN C.AGE BETWEEN 36 AND 45 THEN '36-45'  WHEN C.AGE BETWEEN 46 AND 55 THEN '46-55'  WHEN C.AGE BETWEEN 56 AND 65 THEN '56-65'  WHEN C.AGE > 65 THEN '65+'  END AS AGE\_GROUP,  COUNT(\*) AS LOAN\_COUNT,  SUM(LOAN\_AMOUNT) AS TOTAL\_LOAN\_AMOUNT,  AVG(LOAN\_AMOUNT) AS AVG\_LOAN\_AMOUNT  FROM CUSTOMER\_TABLE C  JOIN LOAN\_TABLE L  ON C.CUSTOMER\_ID = L.CUSTOMER\_ID  GROUP BY AGE\_GROUP  ORDER BY AGE\_GROUP; |
| **Findings** | The 46-65 age group has the highest total loan amounts, while the 18-25 group has the lowest average loan amounts. |
| **Recommendation** | Focus on offering higher-value loans and financial products to the 46-65 age group, while providing accessible loan options with lower amounts for the 18-25 group. |

|  |  |
| --- | --- |
| **Task 9** | **Seasonal Transaction Trends: Examine transaction patterns over years and months to identify seasonal trends in loan repayments.** |
| **Solution** | SELECT YEAR(STR\_TO\_DATE(TRANSACTION\_DATE,'%m/%d/%Y')) AS YEAR,  MONTH(STR\_TO\_DATE(TRANSACTION\_DATE,'%m/%d/%Y')) AS MONTH,  COUNT(\*) AS NUMBER\_OF\_TXN,  SUM(TRANSACTION\_AMOUNT) AS TOTAL\_TXN\_AMT,  AVG(TRANSACTION\_AMOUNT) AS AVERAGE\_TXN\_AMT  FROM TRANSACTION\_TABLE  GROUP BY YEAR,MONTH  ORDER BY YEAR ASC, MONTH ASC; |
| **Findings** | Transaction data reveals consistent seasonal fluctuations, with certain months showing higher loan repayment volumes and larger transaction amounts. |
| **Recommendation** | Leverage seasonal trends to optimize loan repayment reminders and offer tailored financial products during peak repayment months to maximize customer engagement. |

|  |  |
| --- | --- |
| **Task 10** | **Fraud Detection: Highlight potential fraud by identifying mismatches between customer address locations and transaction IP locations.** |
| **Solution** | **Note::** **No column related to IP Location is present in Transaction table, so from Behavior\_Logs we are taking “IP Address” and based on that providing the solution.**  SELECT C.CUSTOMER\_ID,T.TRANSACTION\_ID,T.TRANSACTION\_AMOUNT,T.TRANSACTION\_DATE,C.ADDRESS,B.LOCATION,B.IP\_ADDRESS  FROM CUSTOMER\_TABLE C  JOIN BEHAVIOR\_LOGS B  ON C.CUSTOMER\_ID = B.CUSTOMER\_ID  JOIN TRANSACTION\_TABLE T  ON T.CUSTOMER\_ID = C.CUSTOMER\_ID  WHERE C.ADDRESS NOT LIKE CONCAT('%',B.LOCATION,'%')  ORDER BY T.TRANSACTION\_DATE DESC; |
| **Findings** | A significant number of transactions show discrepancies between the customer's address and the IP location, indicating potential fraudulent activities. |
| **Recommendation** | Implement stricter fraud detection measures, including real-time alerts and verification processes, when transaction IP locations do not match customer address details. |

|  |  |
| --- | --- |
| **Task 11** | **Repayment History Analysis: Rank loans by repayment performance using window functions.** |
| **Solution** | SELECT L.LOAN\_ID,  L.LOAN\_PURPOSE,  L.LOAN\_AMOUNT,  L.LOAN\_STATUS,  L.REPAYMENT\_HISTORY,  RANK() OVER(ORDER BY L.REPAYMENT\_HISTORY DESC, L.LOAN\_AMOUNT DESC) AS REPAYMENT\_RANK  FROM LOAN\_TABLE L  WHERE L.LOAN\_STATUS IN ('Approved','Closed')  ORDER BY REPAYMENT\_RANK ASC; |
| **Findings** | Loans with higher repayment histories are ranked at the top, while larger loan amounts further influence their ranking, showing strong repayment performance. |
| **Recommendation** | Prioritize promoting loans with high repayment ranks and large amounts as successful products while refining lending policies for higher repayment success. |

|  |  |
| --- | --- |
| **Task 12** | **Credit Score vs. Loan Amount: Compare average loan amounts for different credit score ranges.** |
| **Solution** | SELECT  CASE  WHEN C.CREDIT\_SCORE BETWEEN 300 AND 550 THEN '300-550'  WHEN C.CREDIT\_SCORE BETWEEN 551 AND 599 THEN '551-599'  WHEN C.CREDIT\_SCORE BETWEEN 600 AND 750 THEN '600-750'  WHEN C.CREDIT\_SCORE > 750 THEN '750+'  END AS CREDIT\_SCORE\_RANGE,  CASE  WHEN C.CREDIT\_SCORE BETWEEN 300 AND 550 THEN 'Very Poor'  WHEN C.CREDIT\_SCORE BETWEEN 551 AND 599 THEN 'Poor'  WHEN C.CREDIT\_SCORE BETWEEN 600 AND 750 THEN 'Good'  WHEN C.CREDIT\_SCORE > 750 THEN 'Excellent'  END AS CREDIT\_SCORE\_TYPE,  AVG(L.LOAN\_AMOUNT) AS AVERAGE\_LOAN  FROM CUSTOMER\_TABLE C  JOIN LOAN\_TABLE L  ON C.CUSTOMER\_ID = L.CUSTOMER\_ID  GROUP BY CREDIT\_SCORE\_RANGE,CREDIT\_SCORE\_TYPE  ORDER BY CREDIT\_SCORE\_RANGE DESC,CREDIT\_SCORE\_TYPE DESC; |
| **Findings** | The average loan amounts are relatively similar across credit score ranges, with slight variations, indicating that factors other than credit scores might influence loan amounts. |
| **Recommendation** | Consider incorporating additional factors like income or loan purpose alongside credit scores to refine loan amount determinations and mitigate risks. |

|  |  |
| --- | --- |
| **Task 13** | **Top Borrowing Regions: Identify regions with the highest total loan disbursements.** |
| **Solution** | SELECT SUBSTR(TRIM(SUBSTRING\_INDEX(C.ADDRESS, ',', -1)),1,2) AS REGION,  C.NAME, L.LOAN\_PURPOSE,  COUNT(L.LOAN\_ID) AS TOTAL\_LOANS,  SUM(L.LOAN\_AMOUNT) AS TOTAL\_LOAN\_AMT  FROM LOAN\_TABLE L  JOIN CUSTOMER\_TABLE C  ON L.CUSTOMER\_ID = C.CUSTOMER\_ID  GROUP BY REGION,C.NAME,L.LOAN\_PURPOSE  ORDER BY TOTAL\_LOAN\_AMT DESC; |
| **Findings** | Certain regions have significantly higher total loan disbursements, indicating a concentration of borrowing activity in specific areas. |
| **Recommendation** | Focus on targeting high-borrowing regions with tailored financial products and marketing strategies to maximize loan disbursement and customer engagement. |

|  |  |
| --- | --- |
| **Task 14** | **Early Repayment Patterns: Detect loans with frequent early repayments and their impact on revenue.** |
| **Solution** | SELECT C.CUSTOMER\_ID,C.NAME,L.LOAN\_ID,L.LOAN\_AMOUNT,L.LOAN\_DATE,  L.LOAN\_STATUS,COUNT(T.TRANSACTION\_ID) AS EARLY\_REPAY\_CNT,  SUM(T.TRANSACTION\_AMOUNT) AS TOTAL\_EARLY\_REPAY\_AMT  FROM LOAN\_TABLE L  JOIN CUSTOMER\_TABLE C  ON C.CUSTOMER\_ID = L.CUSTOMER\_ID  JOIN TRANSACTION\_TABLE T  ON L.LOAN\_ID = T.LOAN\_ID  WHERE L.LOAN\_STATUS IN ('Approved','Closed')  AND T.TRANSACTION\_TYPE = 'Prepayment'  AND T.STATUS = 'Successful'  GROUP BY C.CUSTOMER\_ID,C.NAME,L.LOAN\_ID,L.LOAN\_AMOUNT,L.LOAN\_DATE,L.LOAN\_STATUS  ORDER BY EARLY\_REPAY\_CNT DESC,TOTAL\_EARLY\_REPAY\_AMT DESC; |
| **Findings** | Customers with frequent early repayments contribute to higher total early repayment amounts, potentially reducing overall interest revenue. |
| **Recommendation** | Review loan terms and conditions to balance early repayment flexibility with profitability, possibly introducing early repayment fees to maintain revenue while accommodating customer preferences. |

|  |  |
| --- | --- |
| **Task 15** | **Feedback Correlation: Correlate customer feedback sentiment scores with loan statuses.** |
| **Solution** | **Note::** **No column related to Customer feedback sentiment is present in Customer table, So from Customer\_Feedback table we are taking “SENTIMENT\_SCORE” and based on that providing the solution.**  SELECT  C.NAME,  L.LOAN\_STATUS,  AVG(IFNULL(CF.SENTIMENT\_SCORE,0)) AS SENTIMENT\_SCORE,  COUNT(\*) AS TOTAL\_FEEDBACKS  FROM LOAN\_TABLE L  JOIN CUSTOMER\_FEEDBACK CF  ON L.LOAN\_ID = CF.LOAN\_ID  JOIN CUSTOMER\_TABLE C  ON C.CUSTOMER\_ID = CF.CUSTOMER\_ID  GROUP BY C.NAME,L.LOAN\_STATUS  ORDER BY NAME DESC; |
| **Findings** | Loan statuses with higher average sentiment scores indicate more positive customer feedback, suggesting a link between loan experience and satisfaction. |
| **Recommendation** | Focus on enhancing the loan experiences for customers with less favorable loan statuses to improve sentiment and overall satisfaction. |

**Performing tasks on Un-Structured Data(Behavior Logs, Customer Feedback) using MongoDB. We have created a database, imported the dataset as collection. Then based on the tasks we have done the query on the collection data.**

|  |  |
| --- | --- |
| **Task 1** | **Insert New Feedback** |
| **Solution** | { "loan\_id": 2001,  "customer\_id": 101,  "feedback\_text": "The loan approval process was seamless, but the interest rate was slightly higher than expected.",  "sentiment\_score": 0.7,  "feedback\_category": "Approval Process",  "escalation\_flag": false,  "escalation\_reason": null,  "timestamp": "2024-11-15T10:30:00Z"  } |
| **Result** |  |

|  |  |
| --- | --- |
| **Task 2** | **Update Escalation Flags** |
| **Solution** | **Filter query:**  {  $expr: {  $and: [  { $eq: ["$escalation\_flag", "True"] },  {  $or: [  { $eq: ["$escalation\_reason", null] },  { $eq: ["$escalation\_reason", "None"] },  { $eq: ["$escalation\_reason", ""] }  ]  }  ]  }  }  **Update query:**  {  $set: {  escalation\_reason: “Delayed response from customer service”  }  } |
| **Result** |  |

|  |  |
| --- | --- |
| **Task 3** | **Remove Duplicate Behavior Logs Input: Remove duplicate entries with the same customer\_id and timestamp.** |
| **Solution** | Checking duplicates using Aggregation:  **Group data:**  {  \_id: {  customer\_id: "$customer\_id",  timestamp: "$timestamp"  },  count: {  $sum: 1  }  }  **Match data where duplicates are found**:  {  count: {  $gt: 1  }  }  As we don’t have any duplicate data in original dataset, so we have inserted one record as duplicate and try to remove. |
| **Result** | **Before Deletion:**    **After Deletion:** |

|  |  |
| --- | --- |
| **Task 4** | **Retrieve Positive Feedback  Input: Retrieve feedback entries with sentiment scores greater than 0.5.** |
| **Solution** | **Filter data:**  { sentiment\_score: { $gt: 0.5 } }  **Columns to fetch(Project data):**  {  loan\_id: 1,  customer\_id: 1,  feedback\_text: 1,  sentiment\_score: 1  } |
| **Result** |  |
| **Finding** | Feedback entries with sentiment scores greater than 0.5 indicate positive user sentiment. |
| **Recommendation** | Focus on enhancing features or services highlighted in this positive feedback to further improve user satisfaction. |

|  |  |
| --- | --- |
| **Task 5** | **Fetch Logs for 'Missed Payment' Actions**  **Input: Retrieve behavior logs where the action is "Missed Payment."** |
| **Solution** | **Filter data:**  { action: "Missed Payment" }  **Columns to fetch(Project data):**  {  customer\_id: 1,  timestamp: 1,  device: 1,  ip\_address: 1  } |
| **Result** |  |
| **Findings** | Logs indicate specific instances of "Missed Payment" actions with details like customer ID, timestamp, device, and IP address. |
| **Recommendation** | Analyze these logs to identify patterns or causes for missed payments and implement targeted reminders or support measures. |

|  |  |
| --- | --- |
| **Task 6** | **Analyze customer behavior and feedback to identify high-risk customers and prioritize operational improvements for fraud detection and customer satisfaction Using an aggregate pipeline.** |
| **Solution** | **Step 1 -- Join Feedback and Behavior Logs:**  Use $lookup to join the Customer Feedback along with Behavior Logs to get the merged data.  {  from: "Behavior\_Logs",  localField: "customer\_id",  foreignField: "customer\_id",  as: "behavior\_data"  } |
| **Result** |  |
| **Solution** | **Step 2 -- Flatten Joined Data:**  Use $unwind to flatten the nested array created by $lookup. This ensures a clean structure for subsequent aggregation stages.  {  path: "$behavior\_data",  preserveNullAndEmptyArrays: true  } |
| **Result** |  |
| **Solution** | **Step 3 -- Aggregate Feedback and Behavioral Metrics:**  Use $group to group the combined data by customer\_id to compute:   * Average sentiment scores. * Total missed payments. * Total session durations.   {  \_id: "$customer\_id",  avg\_sentiment\_score: {  $avg: {  $toDouble: "$sentiment\_score"  }  },  total\_missed\_pymt: {  $sum: {  $cond: [  {  $eq: ["$action", "Missed Payment"]  },  1,  0  ]  }  },  total\_sess\_dur: {  $sum: {  $toInt: "$behavior\_data.session\_duration"  }  },  UnresolvedEscalations: {  $sum: {  $cond: [  {  $and: [  {  $eq: ["$escalation\_flag", "True"]  }  ]  },  1,  0  ]  }  }  } |
| **Result** |  |
| **Solution** | **Step 4 -- Calculate Risk Scores:**  Use $addFields to add a composite risk score, incorporating:   * Missed payments (weighted heavily as an indicator of risk). * Negative sentiment (calculated as 1 - avgSentiment). * Total session durations (scaled if necessary). * Escalation counts.   {  Risk\_Scores: {  $add: [  {  $cond: {  if: {  $gt: ["$total\_missed\_pymt", 0]  },  then: 0.4,  else: 0  }  },  {  $multiply: [  {  $subtract: [1, "$avg\_sentiment\_score"]  },  0.3  ]  },  {  $multiply: [  {  $divide: ["$total\_sess\_dur", 100]  },  0.2  ]  },  {  $multiply: ["$UnresolvedEscalations", 0.1]  }  ]  }  } |
| **Result** |  |
| **New Collection build** | As device details is not present in the last aggregated result, so again using $lookup to join the last aggregated data along with Behavior Logs.   * **Joining Data:**   {  from: "Behavior\_Logs",  localField: "\_id",  foreignField: "customer\_id",  as: "device\_usage"  }   * **Flatten Data:**   {  path: "$device\_usage",  preserveNullAndEmptyArrays: true  }   * **Project Data(Take only required columns):**   {  \_id: 1,  avg\_sentiment\_score: 1,  total\_missed\_pymt: 1,  total\_sess\_dur: 1,  UnresolvedEscalations: 1,  Risk\_Scores: 1,  device: "$device\_usage.device",  action: "$device\_usage.action"  }  As there are different columns required for different pipeline , so taking this data after projection and merge it into a new collection.   * **Merge data:**   {  into: "Joined\_Aggregated\_Collection",  whenMatched: "merge",  whenNotMatched: "insert"  } |
|  | * **Joining and Flatten data:**      * **Project data:**      * **Merge data:** |
| **Solution** | **Step 5 -- Identify Device Usage Trends:**  Group behavior logs by device to detect preferences and anomalies in usage patterns, especially for high-risk actions like "Missed Payment."  Based on the new collection creating a separate pipeline again.   * **Group data(Group the data based on device and customer id and take some aggregated values):**   {  \_id: {  customer\_id: "$\_id",  devices: "$device"  },  count\_of\_device: {  $sum: 1  },  total\_actions: {  $sum: 1  },  miss\_pmt\_count: {  $sum: {  $cond: [  {  $eq: ["$action", "Missed Payment"]  },  1,  0  ]  }  }  }   * **Project data(Take only required aggregated values):**   {  \_id: 1,  count\_of\_device: 1,  total\_actions: 1,  miss\_pmt\_count: 1,  miss\_pmt\_pct: {  $multiply: [  {  $divide: [  "$miss\_pmt\_count",  "$total\_actions"  ]  },  100  ]  }  }   * **Sort data(Sort the data based on wherever the Missed Payments are high means risk is high):**   {  miss\_pmt\_pct: -1  } |
| **Result** | * **Group data:**      * **Project aggregated data:**      * **Sort data based on risk:** |
| **Solution** | **Step 6 -- Detect Session Outliers:**  Use the $bucket to identify customers with session durations that fall into extreme ranges (e.g., above the 90th percentile or below the 10th percentile).  Based on the new collection creating a separate pipeline again.  Using the $bucket to check how many customers fall under which bucket. Here we are taking 5 buckets.  **Creating buckets:**  {  groupBy: "$total\_sess\_dur",  buckets: 5,  output: {  customer\_id: {  $push: "$\_id"  }  }  } |
| **Result** |  |
| **Solution** | **Step 7 -- Escalation Analysis:**   * Filter for customers with: * Non-zero escalation counts. * High composite risk scores. * Highlight these customers for immediate follow-up and prioritization.   Use $match to filter out the data.  **Filter data:**  {  UnresolvedEscalations: {  $gt: 0  },  Risk\_Scores: {  $gte: 10  }  } |
| **Result** |  |
| **Findings** | 1. **High-Risk Customer Profiles**:    * Customers with high missed payments and negative sentiment scores are flagged as high-risk.    * Devices associated with frequent missed payments (e.g., mobile apps) show patterns that may indicate risky behavior. 2. **Session Duration Outliers**:    * Customers with extremely short or long session durations could indicate either lack of engagement or unusual activity.    * 10% of sessions were below 10 minutes, while 5% exceeded 500 minutes. 3. **Escalation Trends**:    * Customers with unresolved escalations and high composite risk scores are a priority for follow-up.    * Escalation counts correlate strongly with negative sentiment. 4. **Device Trends**:    * Mobile devices are the most common platform for missed payments, suggesting potential usability or trust issues. |
| **Recommendation** | 1. **Fraud Detection**:    * Implement additional verification for transactions flagged as high-risk (e.g., multiple missed payments or unusual session durations).    * Use device-level insights to identify suspicious activity. 2. **Customer Satisfaction**:    * Proactively engage with customers who show negative sentiment or unresolved escalations.    * Improve mobile app usability and ensure clear communication to reduce missed payments. 3. **Operational Improvements**:    * Introduce tailored educational campaigns for high-risk customers on payment options and escalation resolution.    * Monitor session duration patterns to detect unusual behavior and enhance the user experience. 4. **Technology Enhancements**:    * Use machine learning models to refine risk scores and identify hidden patterns.    * Invest in anomaly detection systems for real-time monitoring of high-risk actions. |