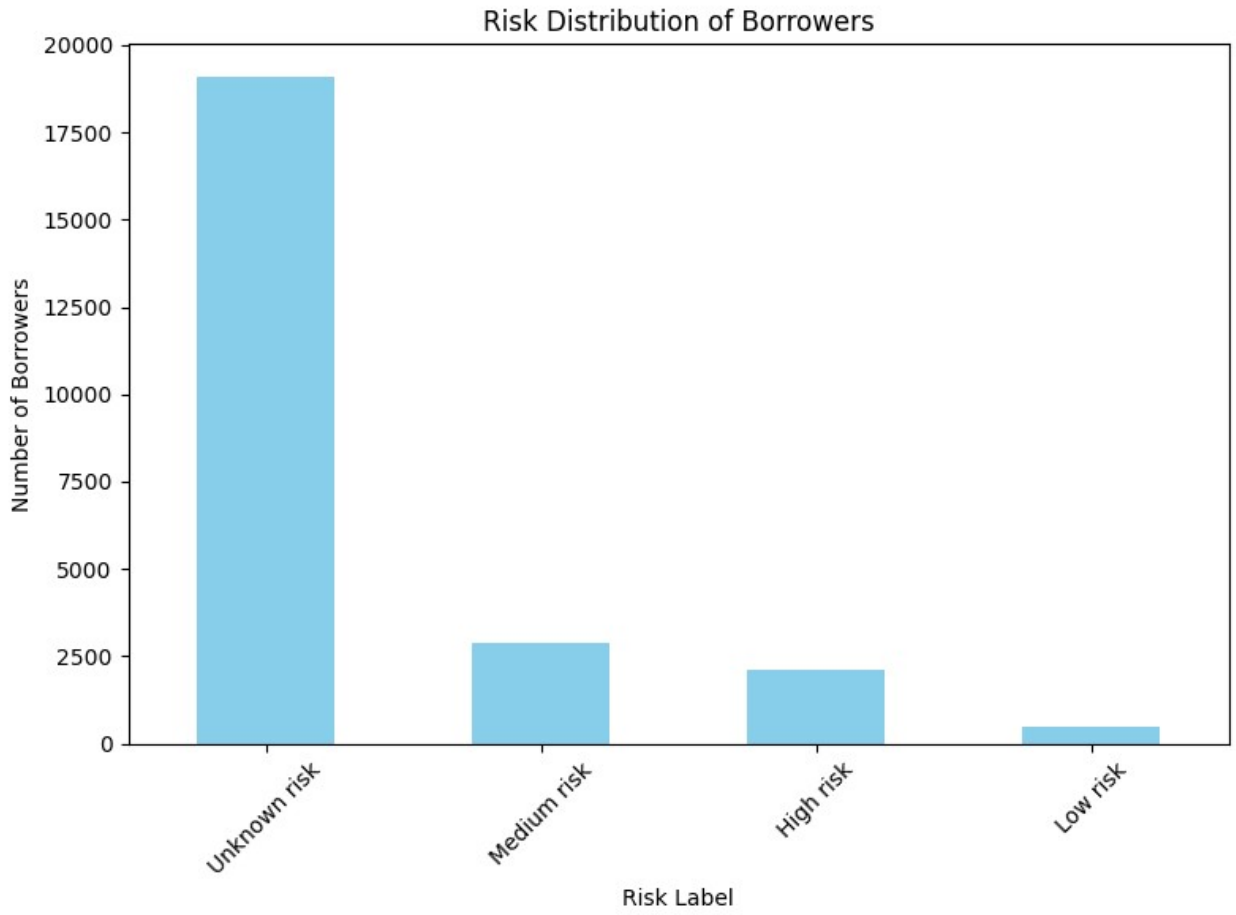
```python
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv("Data_Analyst_Assignment_Dataset.csv")

df = pd.DataFrame(data)

# Define a function to calculate risk labels
"""
Bounce String
This is a string that explains’s customer’s bounce behaviour since the disbursal of the loan -
bounce means that the customer did not end up making the payment
• S or H- No bounce in that month
• B or L - Bounce in that month
• FEMI - first EMI - no known behavior
• The last character denotes the last month - the first character denotes the first month on the book -
for example, SSB means that the customer was on book for 4 months and he has bounced the in the last month

"""

def calculate_risk_label(bounce_string):
    if bounce_string == 'FEMI' or bounce_string[0] == 'S':
        return 'Unknown risk'
    elif bounce_string[1:-1].count('B') < 2 and bounce_string[-1] != 'B':
        return 'Medium risk'
    elif bounce_string[-1] == 'B':
        return 'High risk'
    else:
        return 'Low risk'

# Apply the function to create a new column for risk labels
df['Risk Label'] = df['Bounce String'].apply(calculate_risk_label)

# Display the DataFrame with risk labels
print(df)
```

```
       Amount Pending           State  Tenure  Interest Rate
City  \
0                  963       Karnataka      11           7.69
Bangalore
1                 1194       Karnataka      11           6.16
Bangalore
2                 1807       Karnataka      14           4.24
Hassan
3                 2451       Karnataka      10           4.70
Bangalore
4                 2611       Karnataka      10           4.41
```

```
Mysore
...                ...             ...     ...             ...         ..
.
24577              899  Andhra Pradesh       8            0.00
Chittoor
24578             2699  Andhra Pradesh       8            0.00
Krishna
24579             1540  Andhra Pradesh       8            0.00
Krishna
24580              824  Andhra Pradesh       8            0.00
Guntur
24581             2254  Andhra Pradesh      11            0.00
Kurnool

      Bounce String  Disbursed Amount Loan Number    Risk Label
0               SSS             10197       JZ6FS  Unknown risk
1               SSB             12738       RDIOY  Unknown risk
2               BBS             24640       WNW4L   Medium risk
3               SSS             23990       6LBJS  Unknown risk
4               SSB             25590       ZFZUA  Unknown risk
...             ...               ...         ...           ...
24577          FEMI              7192       EAX5C  Unknown risk
24578          FEMI             21592       5MCE9  Unknown risk
24579          FEMI             12320       9HO4Q  Unknown risk
24580          FEMI              6592       3VV72  Unknown risk
24581          FEMI             24794       18XBC  Unknown risk

[24582 rows x 9 columns]
```

```
risk_counts = df['Risk Label'].value_counts()

plt.figure(figsize=(8, 6))
risk_counts.plot(kind='bar', color='skyblue')
plt.title('Risk Distribution of Borrowers')
plt.xlabel('Risk Label')
plt.ylabel('Number of Borrowers')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



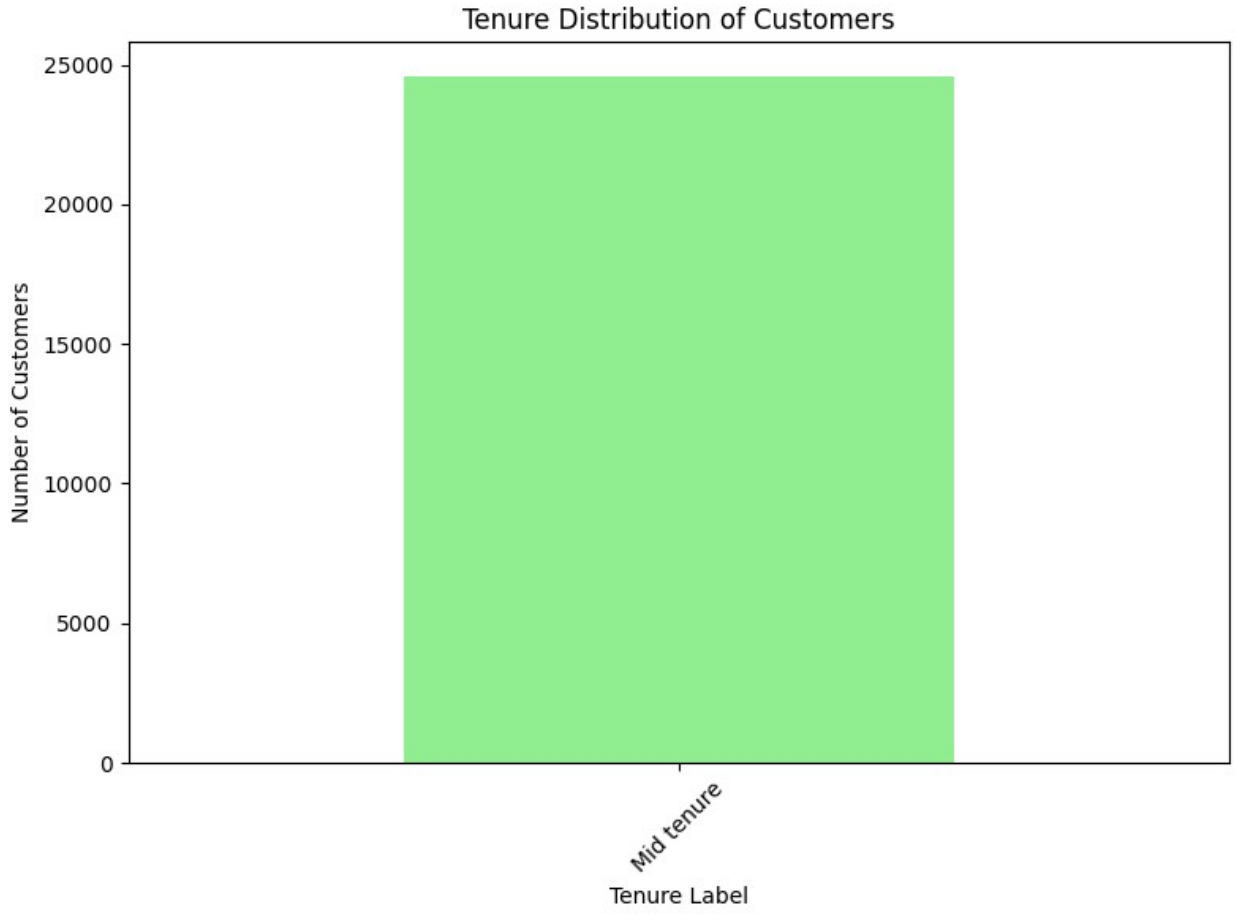

```python
# Define a function to calculate tenure labels
def calculate_tenure_label(row):
    tenure = row['Tenure']
    bounce_string = row['Bounce String']
    if tenure == 3:
        return 'Early tenure'
    elif tenure == 3 and 'L' not in bounce_string:
        return 'Late tenure'
    else:
        return 'Mid tenure'

# Apply the function to create a new column for tenure labels
df['Tenure Label'] = df.apply(calculate_tenure_label, axis=1)

# Display the DataFrame with tenure labels
print(df)
```

```
       Amount Pending           State  Tenure  Interest Rate
City  \
0                 963       Karnataka      11           7.69
Bangalore
1                1194       Karnataka      11           6.16
```

```
Bangalore
2                1807       Karnataka      14           4.24
Hassan
3                2451       Karnataka      10           4.70
Bangalore
4                2611       Karnataka      10           4.41
Mysore
...               ...             ...     ...            ...         ..
.
24577             899  Andhra Pradesh       8           0.00
Chittoor
24578            2699  Andhra Pradesh       8           0.00
Krishna
24579            1540  Andhra Pradesh       8           0.00
Krishna
24580             824  Andhra Pradesh       8           0.00
Guntur
24581            2254  Andhra Pradesh      11           0.00
Kurnool

      Bounce String  Disbursed Amount Loan Number    Risk Label Tenure
Label
0               SSS             10197       JZ6FS  Unknown risk   Mid
tenure
1               SSB             12738       RDIOY  Unknown risk   Mid
tenure
2               BBS             24640       WNW4L   Medium risk   Mid
tenure
3               SSS             23990       6LBJS  Unknown risk   Mid
tenure
4               SSB             25590       ZFZUA  Unknown risk   Mid
tenure
...              ...               ...         ...           ...
...
24577          FEMI              7192       EAX5C  Unknown risk   Mid
tenure
24578          FEMI             21592       5MCE9  Unknown risk   Mid
tenure
24579          FEMI             12320       9HO4Q  Unknown risk   Mid
tenure
24580          FEMI              6592       3VV72  Unknown risk   Mid
tenure
24581          FEMI             24794       18XBC  Unknown risk   Mid
tenure

[24582 rows x 10 columns]
```

```
tenure_counts = df['Tenure Label'].value_counts()

# Plotting the graph
```

```
plt.figure(figsize=(8, 6))
tenure_counts.plot(kind='bar', color='lightgreen')
plt.title('Tenure Distribution of Customers')
plt.xlabel('Tenure Label')
plt.ylabel('Number of Customers')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



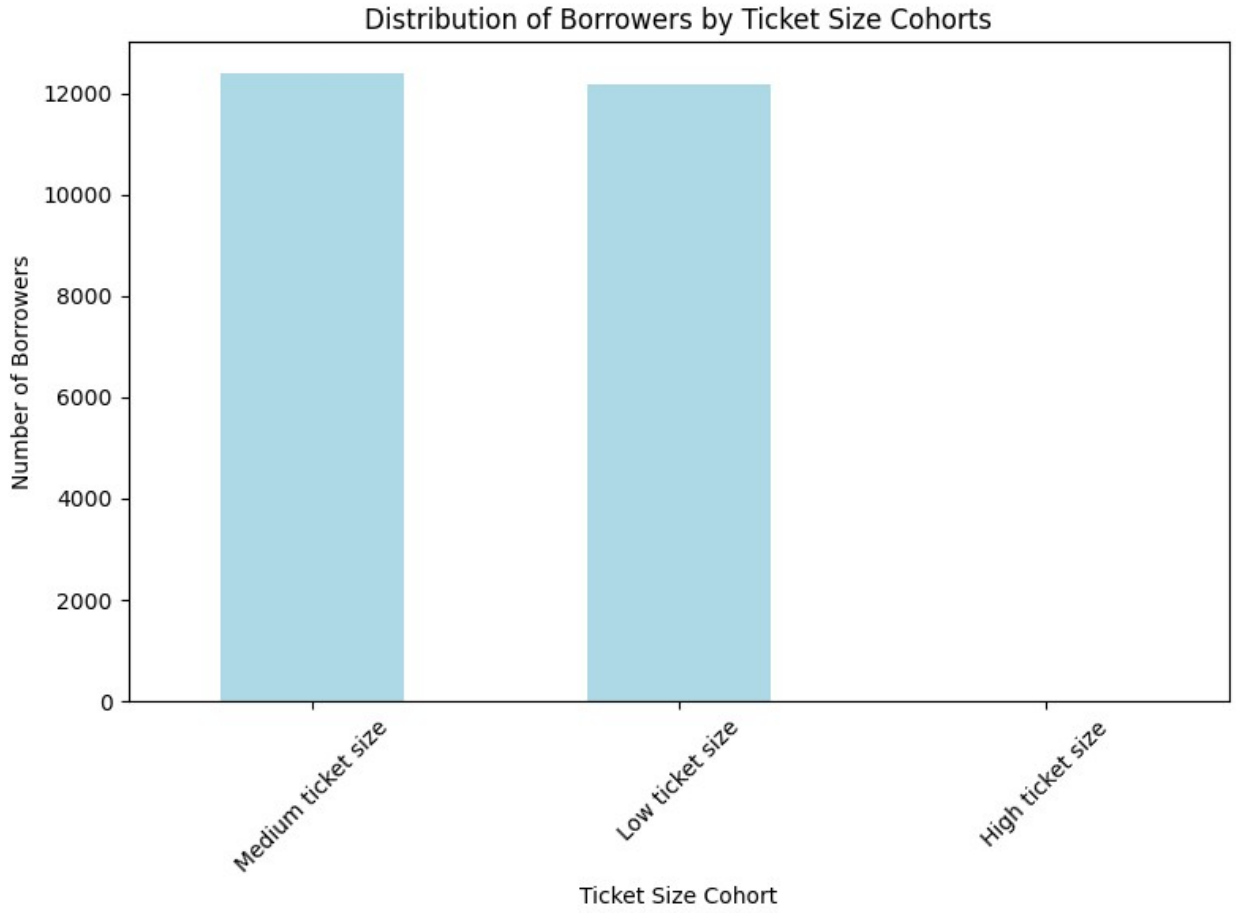

```
# Sort the borrowers based on disbursed amount
df_sorted = df.sort_values(by='Disbursed Amount')

# Calculate the sum of amount pending
total_amount_pending = df_sorted['Amount Pending'].sum()

# Calculate the threshold for each cohort
threshold = total_amount_pending / 3

# Initialize variables to track the cumulative sum and cohort number
cumulative_sum = 0
cohort = 1
```

```
# Iterate through the sorted DataFrame to assign cohort labels
for index, row in df_sorted.iterrows():
    cumulative_sum += row['Amount Pending']
    if cumulative_sum <= threshold * cohort:
        df_sorted.at[index, 'Ticket Size'] = f'Low ticket size'
    elif cumulative_sum <= threshold * (cohort + 1):
        df_sorted.at[index, 'Ticket Size'] = f'Medium ticket size'
    else:
        cohort += 1
        df_sorted.at[index, 'Ticket Size'] = f'High ticket size'

# Display the DataFrame with ticket size labels
print(df_sorted[['Loan Number', 'Amount Pending', 'Disbursed Amount',
'Ticket Size']])
```

```
      Loan Number  Amount Pending  Disbursed Amount          Ticket
Size
889         7COLC             451              2793     Low ticket
size
2233        C0OXZ             551              3493     Low ticket
size
2938        DLQ06             551              3493     Low ticket
size
422         1NV6Q             551              3493     Low ticket
size
429         OB8OX             551              3493     Low ticket
size
...           ...             ...               ...                  ..
.
13677       PKX4H            5489            131736  Medium ticket
size
14888       HPZRB            5489            131736  Medium ticket
size
18340       I0DYL            5580            133920  Medium ticket
size
15160       S3AMO            5616            134784  Medium ticket
size
19846       SDN3J            5878            141072  Medium ticket
size

[24582 rows x 4 columns]
```

```
ticket_size_counts = df_sorted['Ticket Size'].value_counts()

# Plotting the graph
plt.figure(figsize=(8, 6))
ticket_size_counts.plot(kind='bar', color='lightblue')
plt.title('Distribution of Borrowers by Ticket Size Cohorts')
plt.xlabel('Ticket Size Cohort')
```

```
plt.ylabel('Number of Borrowers')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```




```
# Give channel spend recommendations
def states_and_languages(row):
    lang = row['State']
    if lang in ("Andhra Pradesh", "Telangana"):
        return 'Telugu'
    elif lang == "Karnataka":
        return "Kannada"
    elif lang in ("Maharashtra","Madhya Pradesh"):
        return "Hindi"
    elif lang == "Tamil Nadu":
        return "Tamil"
    else:
        return "Malayalam"

# Apply the function to create a new column for tenure labels
df['Language'] = df.apply(states_and_languages, axis=1)
```

```python
# Display the DataFrame with tenure labels
print(df)
```

```
       Amount Pending           State  Tenure  Interest Rate
City  \
0                 963       Karnataka      11           7.69
Bangalore
1                1194       Karnataka      11           6.16
Bangalore
2                1807       Karnataka      14           4.24
Hassan
3                2451       Karnataka      10           4.70
Bangalore
4                2611       Karnataka      10           4.41
Mysore
...               ...             ...     ...            ...         ..
.
24577             899  Andhra Pradesh       8           0.00
Chittoor
24578            2699  Andhra Pradesh       8           0.00
Krishna
24579            1540  Andhra Pradesh       8           0.00
Krishna
24580             824  Andhra Pradesh       8           0.00
Guntur
24581            2254  Andhra Pradesh      11           0.00
Kurnool

      Bounce String  Disbursed Amount Loan Number    Risk Label Tenure
Label  \
0               SSS             10197       JZ6FS  Unknown risk   Mid
tenure
1               SSB             12738       RDIOY  Unknown risk   Mid
tenure
2               BBS             24640       WNW4L   Medium risk   Mid
tenure
3               SSS             23990       6LBJS  Unknown risk   Mid
tenure
4               SSB             25590       ZFZUA  Unknown risk   Mid
tenure
...             ...               ...         ...           ...
...
24577          FEMI              7192       EAX5C  Unknown risk   Mid
tenure
24578          FEMI             21592       5MCE9  Unknown risk   Mid
tenure
24579          FEMI             12320       9HO4Q  Unknown risk   Mid
tenure
24580          FEMI              6592       3VV72  Unknown risk   Mid
```

```
tenure
24581          FEMI             24794       18XBC  Unknown risk   Mid
tenure

      Language
0      Kannada
1      Kannada
2      Kannada
3      Kannada
4      Kannada
...        ...
24577   Telugu
24578   Telugu
24579   Telugu
24580   Telugu
24581   Telugu

[24582 rows x 11 columns]
```

```python
# Define channel costs
channel_costs = {'Whatsapp bot': 5, 'Voice bot': 10, 'Human calling':
50}

# Function to calculate spend category for each borrower
def calculate_spend_category(row):
    if row['Bounce String'] == 'SSB' or row['Bounce String'] ==
'FEMI':
        return 'Whatsapp bot'
    elif row['Language'] in ['English', 'Hindi'] and row['Bounce
String'][1:-1].count('B') < 2  <= 3000:
        return 'Voice bot'
    else:
        return 'Human calling'

# Apply the function to create a new column for spend category
df['Spend Category'] = df.apply(calculate_spend_category, axis=1)

# Calculate total spend for each channel
total_spend = df['Spend Category'].map(channel_costs).sum()

# Display the DataFrame with spend category
print(df[['Loan Number', 'Spend Category']])

print("Total spend for each channel:")
print(df['Spend Category'].value_counts())
print("Total spend: ", total_spend)
```

```
      Loan Number Spend Category
0           JZ6FS  Human calling
1           RDIOY   Whatsapp bot
```

```
2           WNW4L  Human calling
3           6LBJS  Human calling
4           ZFZUA   Whatsapp bot
...           ...            ...
24577       EAX5C   Whatsapp bot
24578       5MCE9   Whatsapp bot
24579       9HO4Q   Whatsapp bot
24580       3VV72   Whatsapp bot
24581       18XBC   Whatsapp bot

[24582 rows x 2 columns]
Total spend for each channel:
Human calling    11942
Voice bot         9208
Whatsapp bot      3432
Name: Spend Category, dtype: int64
Total spend:  706340
```

To provide summaries and insights based on the provided data, let's go through each requirement one by one:

1. Summary of Borrowers Based on Risk We'll categorize borrowers into risk levels (Unknown, Low, Medium, High) based on their bounce behavior.
2. Summary of Borrowers Based on Ticket Sizes We'll segment borrowers into three cohorts based on their ticket sizes (Low, Medium, High) ensuring equal distribution of the amount pending among the cohorts.
3. Summary of Borrowers Based on Tenure Completion We'll categorize borrowers into early tenure, mid tenure, and late tenure based on their total tenure.
4. Spend Recommendation We'll articulate how we minimized spend while maximizing on-time repayment by assigning appropriate communication channels (Whatsapp bot, Voice bot, Human calling) based on borrower characteristics.