Appendix A - Preprocessing

June 24, 2023

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
accepted_df = pd.read_csv("data/accepted_2007_to_2020.csv", index_col=0)
# rejected_df = pd.read_csv("data/rejected_2007_to_2018Q4.csv")
```

1 Accepted Loans

Note: Looking at columns with the same number of nulls gives us an idea of which variables were "generated" together.

In [ ]:
```
accepted_df = accepted_df[~accepted_df.loan_amnt.isna()]
```

In [ ]:
```
df = accepted_df.set_index("id")

# Handle missing values
df.dropna(
    axis=1, thresh=len(df) * 0.9, inplace=True
)  # Drop columns with more than 90% missing values
df.dropna(inplace=True)  # Drop rows with any missing values

# Feature engineering
df["issue_d"] = pd.to_datetime(df["issue_d"])  # Convert issue date to datetime
df["year"] = df["issue_d"].dt.year  # Extract year from issue date
df["month"] = df["issue_d"].dt.month  # Extract month from issue date


# Convert int_rate to numerical
df["int_rate"] = df["int_rate"].str.rstrip("%").astype("float") / 100.0
df["term"] = df["term"].apply(lambda x: int(x.split()[0]))
df["emp_length"] = df["emp_length"].str.extract(r"(\d+)")
df["emp_length"] = pd.to_numeric(df["emp_length"], errors="coerce")
# Feature engineering - Extract year and month from issue date
df["issue_d"] = pd.to_datetime(df["issue_d"])
df["issue_year"] = df["issue_d"].dt.year
```

```python
# Calculate credit history length
df["earliest_cr_line"] = pd.to_datetime(df["earliest_cr_line"])
df["credit_history_length"] = df["issue_year"] - df["earliest_cr_line"].dt.year
# Convert revol_util to numeric
df["revol_util"] = df["revol_util"].str.rstrip("%").astype("float") / 100.0

# Calculate the difference between last payment and issue date in years
df["last_pymnt_d"] = pd.to_datetime(df["last_pymnt_d"])
df["last_pymnt_issue_diff"] = (df["last_pymnt_d"] - df["issue_d"]).dt.days // 365

# Calculate the difference between last credit pull and issue date in years
df["last_credit_pull_d"] = pd.to_datetime(df["last_credit_pull_d"])
df["last_credit_pull_issue_diff"] = (
    df["last_credit_pull_d"] - df["issue_d"]
).dt.days // 365

# Convert "debt_settlement_flag" and "hardship_flag" to numeric
df["debt_settlement_flag"] = (df["debt_settlement_flag"] == "Y").astype(int)
df["hardship_flag"] = (df["hardship_flag"] == "Y").astype(int)


# I made issue year categorical to capture trends, e.g. loans back in 2008 might behav
categorical_cols = [
    "grade",
    "sub_grade",
    "home_ownership",
    "verification_status",
    "purpose",
    "addr_state",
    "initial_list_status",
    "loan_status",
    "issue_year",
    "application_type",
]
df = pd.get_dummies(df, columns=categorical_cols)


columns_to_drop = [
    "emp_title",
    "url",
    "title",
    "zip_code",
    "issue_d",
    "last_pymnt_d",
    "last_credit_pull_d",
    "pymnt_plan",
    "earliest_cr_line",
]
```

```
        df.drop(columns=columns_to_drop, inplace=True)
```

In [ ]:
```
# check that everything is numeric
df.dtypes.value_counts()
```

In [ ]:
```
# check that there are no null values
df.isna().sum().value_counts()
```

In [ ]:
```
df.shape
```

In [ ]:
```
df.to_pickle("data/preprocessed_df.pickle")
```