LLM Implementation Directions

1. Ensemble models
2. Synthetic data
3. Web scraping
4. Feature explanation/Generation

Progress for week 3/22-3/28

Sherry (Web Scraping/LLM)

- Used OpenAI API to get the industry type & company size for emp_title
- Used googlesearch to get official websites

	emp_title_clean	title_type	industry	company_url	company_size
0	ryder	company	Transportation	https://www.ryder.com/en-us	Large
1	air resources board	company	Government	None	Large
2	university medical group	company	Healthcare	https://www.ubmd.com/	Large
3	veolia transportaton	company	Transportation	https://www.veolianorthamerica.com/who-we-serv...	Large
4	southern star photography	company	Other	https://www.instagram.com/southernstarphoto/?h...	Small
5	mkc accounting	company	Finance	https://www.yelp.com/biz/mkc-accounting-and-in...	Small
6	starbucks	company	Food & Beverage	https://www.starbucks.com/	Large
7	southwest rural metro	company	Healthcare	https://www.ruralmetrofire.com/	Medium
8	ucla	company	Education	https://uclabruins.com/	Large
9	va dept of conservationrecreation	company	Government	https://www.facebook.com/VirginiaDCR/	Large

- Next step
    a. Enrich more variables with GPT - fast to implement; need to control token cost
    b. Web scraping (company website/crunchbase) - free; but slow and our data too sparse
    c. Generate natural language borrower profile and pass to gpt - can use as soft label; low interpretability and potential high cost

Yijiao (Synthetic Data)

- Created a new column called description, which gathers the information from borrowers in the dataset including loan_amnt, purpose, emp_title, emp_length, annual_inc, home_ownership, open_acc, and total_acc.
- Next step: Use a Large Language Model (LLM) to generate a risk score based on the synthetic description field. This score will then be integrated with structured pre-loan features to train an xGBoost model, allowing evaluation of whether LLM-generated features improve predictive performance.

Xintong (Synthetic Data)

- SMOTE generates new, synthetic examples by interpolating between a data point and its nearest neighbors. (use temporal splitting) In our case, the minority class (Loss loans) made up only about 13% of the data, which caused the model to largely ignore them. After applying SMOTE, the training data became balanced, and the model's ability to detect bad loans improved significantly—recall for class 0 increased from 0.05 to 0.30, and the F1 score became more balanced across classes.

- While SMOTE is effective, it also has limitations: it may amplify noise if the minority class is noisy, and it can increase training time due to a larger dataset.

Jiaxuan (Feature Explanation)

- Based on current prediction results and feature importance from the temporal-splitting model. Choose top importance features to import into LLM
- Clearly construct instruction prompts that ask LLM to intuitively explain the relationship between each important feature and loan risk.

  Example:

  Feature: loan_amnt

  Prompt: Explain clearly why a larger loan amount typically increases the risk of default on loans
- Implementation: Through OpenAI's api to get access to GPT-4.

  Code:

```python
import openai

openai.api_key = "your-api-key"

feature_prompts = {
    "loan_amnt": "Explain clearly why a larger loan amount typically increases the risk of default on loans.",
    "installment": "Intuitively explain how higher monthly installment payments might impact the borrower's ability to repay the loan.",
    "emp_length": "Clearly explain why longer employment length might reduce the risk of loan default.",
    "annual_inc": "Explain intuitively why higher annual income typically decreases loan default risk.",
    "dti": "Clearly explain why a high debt-to-income ratio increases the likelihood of default.",
    "fico_range_high": "Intuitively explain why borrowers with higher FICO scores generally have lower loan default risk."
}

feature_explanations = {}

for feature, prompt in feature_prompts.items():
    response = openai.ChatCompletion.create(
        model="gpt-4",
        messages=[
            {"role": "system", "content": "You are a financial expert explaining predictive features clearly and intuitively."},
            {"role": "user", "content": prompt}
        ],
        temperature=0.2,
        max_tokens=250
    )
    explanation = response.choices[0].message.content.strip()
    feature_explanations[feature] = explanation
    print(f"\nFeature: {feature}\nExplanation: {explanation}\n")

```

- Based on the feature explanations provided by LLM, create a clear, intuitive, and stakeholder-friendly summary table or report.
- Analyze LLM-generated Explanations for Insights (Use LLM)
  Example:

  LLM Explanation:
  "High monthly installments limit borrowers' disposable income, making them vulnerable to default, especially if coupled with high debt-to-income ratios."

  Insight: **Interaction** between `installment` and `dti` may significantly influence default risk.

- Based on the insights from explanations, create new interaction features to enhance predictive power:

  Example: df['installment_dti_interaction'] = df['installment'] * df['dti']

  Example: Insights: "Borrowers with very high bankcard utilization are often financially strained, causing exponential risk increases."

    Feature Transformation: df['bc_util_squared'] = df['bc_util'] ** 2

- Use LLM explanations clearly and practically to identify and possibly remove redundant or irrelevant features, improving generalization.
    - Features with unclear or weak LLM explanations might have limited predictive relevance.

    - Clearly test accuracy after feature removal to confirm.

- Retrain and Optimize the baseline model based on the newly modified features based on the interpretations of original features from LLM (Get an ensemble model)

Andy (TabNet neural network for ensembling)

- Continued implementation of the TabNet neural network
- Perform train/validate/test
- Assess performance of the XGBoost and TabNet models individually
- Careful consideration to be given to ensure that the XGBoost component of the ensemble model is not overshadowed by the neural network
- Next steps: think about ensembling process, further hyperparameter tuning of TabNet, and other neural networks potentially