Machine Learning Project

In this project we will build a credit risk model and design a strategy to assign credit.

Assessment: The project has 30 points. Each of you will be asked 2 questions. If you answer both correctly, you will get 30. If answer one you will get 20. If you miss both, you will get 10. There will be no make-up. Make sure you understand all the steps, and you will be fine.

1. Download the data from https://www.kaggle.com/competitions/amex-default-prediction/data. We will work with "train_data.csv" and "train_labels.csv".
   "train_labels.csv" shows default status of customers as of April 2018 (target variable). "train_data.csv" shows their activity in the last 13 months (April 2017 to April 2018). These characteristics will be used to define independent variables.
2. Read project's description, type of attributes, …
3. Read the data. Since the data is large you may have to read it in Chunks. This is an example: https://stackoverflow.com/questions/25962114/how-do-i-read-a-large-csv-file-with-pandas
4. Due to data size, you may get memory error while doing the project; so we will use only 20% of observations. Randomly choose 20% of observations from the "**train_labels.csv**". Merge this sample with "train_data.csv". This will be your development sample. **Save this data**, so in the future you don't have to read the original large file again.
   Often there are mistakes in this step. Make sure you do it correctly. Read this step carefully.
5. Perform One-Hot encoding on categorical variables.
6. Next, we define the independent variables. As mentioned, we have historical data for up to 13 months for each customer. We need to aggregate these data to define features as of April 2018.
   For **Numerical** features, aggregation can be done by: Average, Sum, Min, Max,… Also I suggest you include feature's value as of April 2018, which is the most recent value.
   Here are some examples for some aggregated features based on feature X_1:
   - X_1_Ave_6: Average X_1 in the last 6 months
   - X_1_Ave_12: Average X_1 in the last 12 months
   - X_1_Min_6: Minimum X_1 in the last 6 months
   - X_1_Max_9: Maximum X_1 in the last 9 months
   - X_1_Sum_3: You know
   - X_1_Apr_2018
   - You name it: (X_1_Apr_2018 – X_1_Apr_2017)/ X_1_Apr_2017
   - …

   Note: For some observations you have less months of data. So the above features may be calculated with less months. For example, for an application with 4 months of data, X_1_Ave_6 will be calculated based on average of X_1 in the last 4 months.

   For **Categorical** features, some examples for aggregation are as following. Note that you have already done one-hot-encoding and your categorical features are binary (0/1).

   - X_1_Response_Rate_6: Percentage of times X_1 equals 1 in the last 6 months.

- X_1_Ever_Response_12: Whether X_1 is response at least once in the last 12 months
- X_1_April_2018
- …

7. Split data into 70% as Train sample, 15% as Test1, and 15% as Test2.
8. Next we want to reduce number of features, and keep only features which have high predictive power. To do so we build an XGB model and will keep features with Feature Importance higher than 0.5%.
   Make sure all missing values are stored as NaN, so XGBoost can work with them.
   Run an XGBoost model on the train sample, with default parameters. Don't forget to drop unnecessary columns if any. Calculate feature importance and save the feature importance as a CSV file.
   Run another XGBoost model, which has 300 trees, 0.5 as learning rate, maximum depth of trees is 4, uses 50% of observation to build each tree, uses 50% of features to build each tree, and assigns a weight of 5 to default observations. Save the feature importance as a CSV file.
   Keep features that have feature importance of higher 0.5% in any of the two models. We will use only these features after this.
9. Next we run Grid Search for the XGBoost model (using only features we chose in step 10). Use the following combinations in the grid search:
   - Number of trees: 50, 100, and 300
   - Learning Rate: 0.01, 0.1
   - Percentage of observations used in each tree: 50%, 80%
   - Percentage of features used in each tree: 50%, 100%
   - Weight of default observations: 1, 5, 10

   Create the following table. **Update the table after each iteration of grid search and save the table**, so in case you got memory error or any other issues, you don't need to re-run that part of Grid.

# Trees	LR	Subsample	% Features	Weight of Default	AUC Train	AUC Test 1	AUC Test 2
50	0.01	50%	50%	1	…	…	…
…	…	…	…	…	…	…	…

10. Choose the best model, based on bias and variance. Re-run the model with optimum parameters, and save the final XGB model.
11. Next, grid search for Neural Network. We first need to process the data. We have already done one-hot encoding. We need to do Missing Value Imputation, Outlier Treatment, and Normalization. **We will use only features that we chose in step 10**. As mentioned, probably there is no need for outlier treatment and feature scaling; but to practice, cap and floor observations at 1 and 99 percentiles. Use StandardScaler for normalization (standardization). Replace missing values with 0.
12. Next we run Grid Search for the Neural Network model. Use the following combinations in the grid search:
    - Number of hidden layers: 2, 4

- # nodes in each hidden layer: 4, 6
- Activation function for hidden layers: ReLu, Tanh
- Dropout regularization for hidden layers: 50%, 100% (no dropout)
- Batch size: 100, 10000

Use Adam for optimizer, Cross Entropy for Loss function, and 20 for number of Epochs. For everything else, use default parameters.

Note you would need to run separate For Loops for different number of Hidden Layers.

Create the following table. **Update the table after each iteration of grid search and save the table**, so in case you got memory error or any other issues, you don't need to re-run that part of Grid.

# HL	# Node	Activeation Function	Dropout	Batch Size	AUC Train	AUC Test 1	AUC Test 2
2	4	ReLu	50%	100	…	…	…
…	…	…	…	…	…	…	…

13. Choose the best model, based on bias and variance. Re-run the model with optimum parameters, and save the final NN model.
14. Choose the best model among NN and XGB (models of step 11 and step 14)

Strategy:

Next, you want to define two strategies: a conservative and an aggressive. For each strategy, you define a threshold to accept/reject applicants based on the model's output. Applicants with probability of default (model's output) lower than threshold, will be accepted, and those with PD higher than threshold will be rejected. The conservative strategy has a lower threshold compared with the aggressive one; hence accepts less applicants.

We will estimate Portfolio's default rate, and Revenue based on each strategy, show it to management, and let them decide which strategy is better.

Estimate Portfolio's Default Rate: You already know how to calculate default rate for a strategy; you just need to calculate actual default rate (Y = 1) among applications that will be accepted based on the strategy.

Estimate Portfolio's Revenue: Revenue on a credit card depends on two factors: how much the customer spends, and how much of monthly balance the customer does not pay (roll over to the next month). Credit Card companies, charge a small amount for each dollar you spend. Also they charge an interest rate on the remaining monthly balance that you do not pay (revolving balance).

For example, assume a CC charges 0.1% on each dollar spent, and charges 24% (annually) on balances. If a customer spend $1000 in a month, company's revenue from spend of this customer will be 1000×0.001=$1. If customer pays back $200 out of $1000, company will charge 2% monthly interest (24% annually) on the remaining $800, which means $16 interest revenue in that month.

To have a prediction of spend and balance for a customer, we use their historical spend and balance. In the data, features that start with S_ are spend variables, and features that start with B_ are balance variables. Choose one spend and one balance feature (any feature of your choice). Calculate average of these two features for the last 6 months (i.e. November 2017 to April 2018). If we show these two averages with S_Ave and B_Ave, monthly revenue for a customer would be calculated as:

$$Monthly\ Revenue\ for\ 1\ Customer = B\_Ave \times 0.02 + S\_Ave \times 0.001$$

And Expected Revenue in the next 12 months would be 12 multiplied by the above value.

To estimate portfolio's expected revenue based on a strategy, calculate sum of the above revenue among customers who are accepted based on the strategy. Assume a revenue of 0 for those who default.

15. Write a function that calculates default rate and revenue based on a threshold. Function gets six inputs:
    - Data with four columns: Target Variable (Default indicator), Default model's output (PD), Estimated Monthly Balance, Estimated Monthly Spend
    - Name of Target Variable (as a text/string)
    - Name of default model's output (as a text/string)
    - Name of Estimated Monthly Balance variable (as a text/string)
    - Name of Estimated Monthly Spend variable (as a text/string)
    - Threshold (a number between 0 and 1)

    And will return two outputs: portfolio's default rate, and portfolio's expected revenue.

    Use only train sample to try a few thresholds, and choose one conservative and one aggressive strategy. It is up to you how to choose the thresholds. Imagine you want to present it to senior management and want to impress them with your work/results. The only constraint is that company does not want the default rate to be higher than 10%.

Prepare the presentation:

General Guidelines:

1. Create pretty slides
2. Don't use any background
3. Format numbers, use 1000 separators. Decimal numbers with 2 decimal places (in case of very small numbers with 3 decimal places)
4. Don't use small fonts that can not be seen
5. Don't put too much material in a slide
6. Each slide should be self explanatory. While you don't want to put too much material, put enough material that explain the stuff in the slide
7. Format tables. Assign appropriate titles to tables and figures
8. Don't copy paste from your code
9. Have a good story to tell
10. Format everything. Standard fonts …
11. Use colors, but don't overuse

In general, remember a presentation is like presenting a product. Both packaging and functionality matter. **You need to wrap your good model in a pretty package.**

Note 1: In the following steps, feel free to change the format of tables to make the slides easier to follow and understand.
Note 2: I have proposed the minimum items to be included in the slides. Feel free to add additional explanations, …
Note 3: Due to computational constraints, you may need to simplify the project, read less rows and observations, … Adjust the following tables based on your final sample.

Fill the attached deck with your results.

Credit Risk
Project.pptx

Slide #1. Executive Summary. This is where you sell your model. Show the results of your strategies, and add any explanation that can attract people. In this slide imagine you are a seller. Include the following table. Talk about the project, project's goal, why this project is important, how it helps the company, and anything that might be interesting (like these days people get excited when they hear AI …)
Propose the strategy that you think help the company better. Explain why you think this is a better strategy.

	Train			Test 1			Test 2		
	#Total	Default Rate	Revenue	#Total	Default Rate	Revenue	#Total	Default Rate	Revenue
Conservative Strategy									
Aggressive Stratgey									

Slide #2. Data. Explain your data (data of step 3). Explain why you chose April 2018 originations (come up with a story). Include the following table, explain why you decided to use this data, explain your target variable (you can generate a story for what default means), …

Category	**# Observations**	**Default Rate**
All Applications		
Applications with 13 months of historical data		
Applications with 12 months of historical data		
Applications with 11 months of historical data		
Applications with 10 months of historical data		
Applications with 9 months of historical data		
Applications with 8 months of historical data		
Applications with 7 months of historical data		
Applications with 6 months of historical data		
Applications with 5 months of historical data		
Applications with 4 months of historical data		
Applications with 3 months of historical data		
Applications with 2 months of historical data		
Applications with 1 month of historical data		

All Applications	# Obs	Default Rate
13 Months	77359	23.38%
12 Months	2019	38.68%
11 Months	1164	45.10%
10 Months	1268	46.61%
9 Months	1328	46.54%
8 Months	1220	44.75%
7 Months	1038	43.93%
6 Months	1095	37.35%
5 Months	958	38.83%
4 Months	926	40.93%
3 Months	1154	34.84%
2 Months	1227	30.97%
1 Month	1027	33.40%

Slide #3. Features. Talk about categories of independent variables used in the development process (data of step 3). Use raw features; i.e. features as they are in the raw data, and before defining new features in step 5.

Category	# of features

Slide #4. Feature Engineering. Talk about type of features you have created (step 5). You can talk about categories, such as Average, Median, Min, Max, …

Add a table like table of slide 3, this time not for raw features, but for features you have defined based on raw features.

Category	# of features

Also show summary statistics for the top 5 features with highest SHAP values in the best XGBoost model (Step 12). Note that at this point you don't need to talk about the XGBoost model and SHAP. You can just mention that based on your analyses these are among the most important attributes.

Feature	Min	1 Percentile	5 Percentile	Median	95 Percentile	99 Percentile	Max	Mean	% Missing

Slide #5. Data Processing / One-Hot Encoding. Show the categorical variables, and show how you treated them. Show the results after One-Hot Encoding. Include your code to do one-hot encoding.

Slide #6. Feature Selection. Add a graph that explains your feature selection process (steps 7 to 10). Create a pretty graph. Attach an excel file with results of feature importance for two models (steps 8 and 89 Add a column to table of slide 4, that shows # features selected from each category to be used in grid search.

Category	# of features	# selected

Slide #7. XGBoost - Grid Search. Include your grid search code. Explain why you chose these parameters (don't say because you said …). Talk about your experience with grid search, how many models you trained, any lessons learned, …

Slide #8. XGBoost - Grid Search. In this slide, we create scatter plots for models of grid search, and will choose the best model based on the scatter plot. For each of the models of grid search, calculate average and standard deviation of AUC across three samples (train and tests). Then include 2 scatter plots in the slide:

- In the first one, X_Axis shows Average AUC, and Y-Axis shows Standard Deviation of AUC.
- In the second one, X-Axis is AUC of train sample and Y-Axis is AUC of Test 2 sample.

Explain which model you would choose based on each scatter plot.

Slide #9. XGBoost – Final Model. Show the parameters of the final model, also AUC of model on each sample. Also show how model Rank Orders on each of the three samples. Check the last part of XGBoost sample code, for rank ordering. Note you need to define score bins based on the train sample, and apply the same thresholds to test samples. Show rank orderings in a Bar-Chart, where each sample is one series in Bar Chart, X-Axis shows score bins (intervals), and Y-Axis shows default rate in each bin.

Slide #10. XGBoost – SHAP Analysis. Show Beeswarm Graph for the final model, based on Test 2 sample. Add some explanation of your choice. You can talk about ranking of attributes, correlation between attribute and the output, …

Slide #11. XGBoost – SHAP Analysis. Show Waterfall Graph for the final model, based on one observation in Test 2 sample. Add some explanation of your choice. You can talk about which attributes are driving the score, how to improve the score, …

Slide #12. Neural Network – Data Processing. Explain your data processing for Neural Network. Feel free to add code, tables, … Format this slide, so it is easy to follow and understand.

Slide #13. Neural Network - Grid Search. Include your grid search code. Explain why you chose these parameters (don’t say because you said …). Talk about your experience with grid search, how many models you trained, any lessons learned, …

Slide #14. Neural Network - Grid Search. In this slide, we create scatter plots for models of grid search, and will choose the best model based on the scatter plot. For each of the models of grid search, calculate average and standard deviation of AUC across three samples (train and tests). Then include 2 scatter plots in the slide:

- In the first one, X_Axis shows Average AUC, and Y-Axis shows Standard Deviation of AUC.

- In the second one, X-Axis is AUC of train sample and Y-Axis is AUC of Test 2 sample.

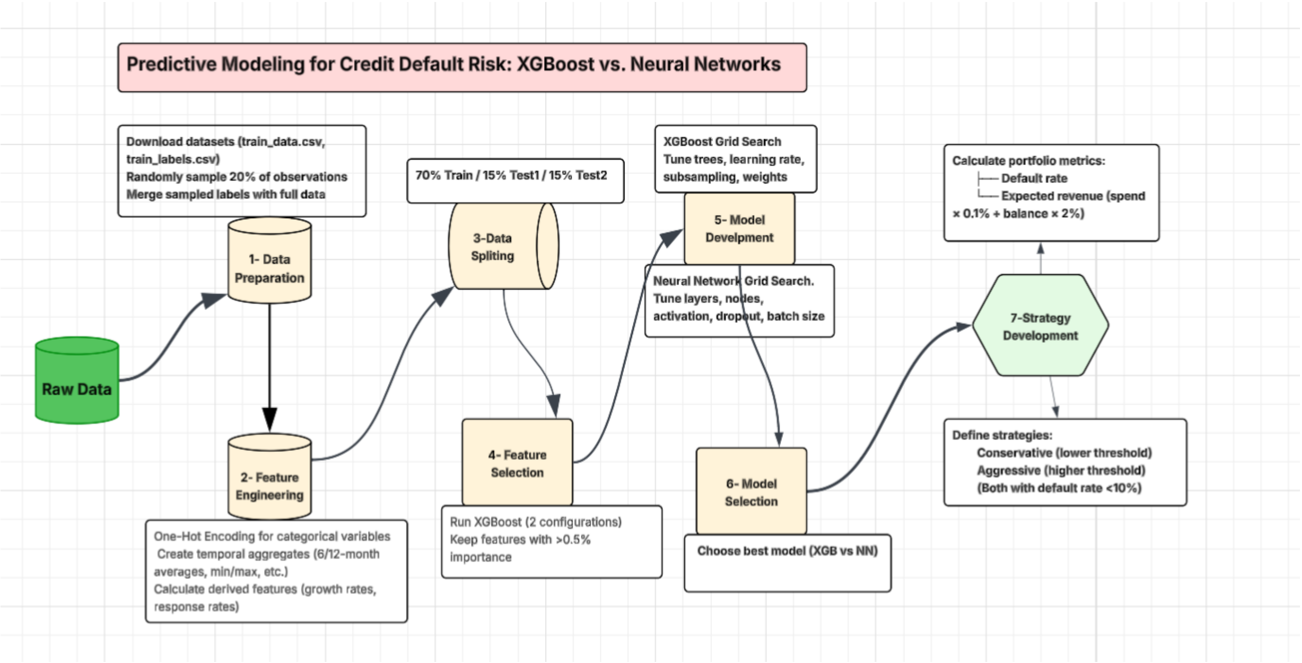
Explain which model you would choose based on each scatter plot.

Slide #15. Neural Network – Final Model. Show the parameters of the final model, also AUC of model on each sample. Also show how model Rank Orders on each of the three samples. Check the last part of XGBoost sample code, for rank ordering. Note you need to define score bins based on the train sample, and apply the same thresholds to test samples. Show rank orderings in a Bar-Chart, where each sample is one series in Bar Chart, X-Axis shows score bins (intervals), and Y-Axis shows default rate in each bin.

Slide #16. Final Model. Talk about the final model (XGBoost or Neural Net), and why you chose this one. Add tables or graphs from previous steps to support your reasoning …

Slide #17. Strategy. Include the function you have written in step 17. Also include the following table. Explain what thresholds you chose for conservative and aggressive strategy, and explain your rationale.

	Train			Test 1			Test 2		
Threshold	#Total	Default Rate	Revenue	#Total	Default Rate	Revenue	#Total	Default Rate	Revenue
0.1									
0.2									
0.3									
0.4									
0.5									
0.6									
0.7									
0.8									
0.9									
1									



Predictive Modeling for Credit Default Risk: XGBoost vs. Neural Networks

Raw Data

1- Data Preparation

Download datasets (train_data.csv, train_labels.csv)
Randomly sample 20% of observations
Merge sampled labels with full data

2- Feature Engineering

One-Hot Encoding for categorical variables
Create temporal aggregates (6/12-month averages, min/max, etc.)
Calculate derived features (growth rates, response rates)

3-Data Spliting

70% Train / 15% Test1 / 15% Test2

4- Feature Selection

Run XGBoost (2 configurations)
Keep features with >0.5% importance

5- Model Develpment

XGBoost Grid Search
Tune trees, learning rate, subsampling, weights

Neural Network Grid Search.
Tune layers, nodes, activation, dropout, batch size

6- Model Selection

Choose best model (XGB vs NN)

7-Strategy Development

Calculate portfolio metrics:
├── Default rate
└── Expected revenue (spend × 0.1% + balance × 2%)

Define strategies:
Conservative (lower threshold)
Aggressive (higher threshold)
(Both with default rate <10%)