*Q: “Can interpretable ML models predict stimulus spending behavior using financial vulnerability indicators?”*

*Title: Predicting COVID-19 Stimulus Payment Usage: A Machine Learning Approach with Interpretable Policy Insights*

*Great — since you're following a traditional CS/ML research workflow and already have the NFCS dataset, I’ll outline a detailed step-by-step process aligned with best practices for ML research (e.g., NeurIPS, ICML, AAAI-style papers).*

*This will include: data labeling, preprocessing, modeling, evaluation, interpretability, paper structuring, and preparation for submission. I’ll get back to you shortly with a comprehensive project plan.*

***Project Roadmap: Classifying Stimulus Payment Usage with NFCS 2021***

***1. Dataset Familiarization and Feature Engineering***

***Understanding the NFCS 2021 Dataset:*** *Start by reviewing the 2021 National Financial Capability Study (NFCS) State-by-State survey documentation and data dictionary. The 2021 NFCS is a large survey of over 25,000 U.S. adults (about 500 per state, plus D.C. and Puerto Rico) covering financial behaviors, attitudes, and demographics (*[*Using the NFCS Data | Finra Foundation*](https://finrafoundation.org/using-nfcs-data#:~:text=The%20State,ethnicity%2C%20education%2C%20and%20Census%20Division)*). Key sections include demographics (Section A), financial status and behaviors, and a special section (Section J) on COVID-19 impacts and stimulus payments. Become familiar with the survey questions and coding (e.g. codes for “Don’t know” or “Prefer not to say” are 98 and 99 in most variables). Pay particular attention to question* ***J51****, which asks those who received a federal COVID-19 stimulus payment: “What did you use the money for? (Select all that apply)”. This is the basis for our classification labels.*

***Identifying the Target Variable (Stimulus Usage):*** *Extract the responses to J51 for each respondent. J51 allows multiple selections: options include “Made purchases or paid bills,” “Paid down debt,” “Added to savings,” “Invested in the stock market,” “Donated to individuals/organizations,” and “Other,” plus “Don’t know/Prefer not to say.” Our goal is to classify* ***stimulus usage behavior*** *into three main categories:* ***Spending****,* ***Saving****, or* ***Debt Repayment****. To create a target label for each individual:*

* ***Filter respondents:*** *First, limit the dataset to respondents who actually* ***received at least one stimulus payment*** *(78% of the sample received one ()). Those who did not receive a payment will be excluded from the classification task (since they have no usage to classify).*
* ***Derive usage labels:*** *For each recipient, determine which of the three target categories apply. We may need to handle multi-response carefully: many respondents indicated using the money for multiple purposes (e.g. both paying bills and adding to savings). For simplicity in a multi-class classification, one strategy is to assign* ***one primary category per respondent****. For example, if an individual selected more than one category, we might assign the category corresponding to where the majority of their stimulus was used (if such information is available) or use a priority rule (e.g. if “Paid down debt” was selected, label as Debt, else if “Added to savings” selected (but not debt), label as Saving, else if “Made purchases or paid bills” selected, label as Spending). This ensures each person has a single label:* ***Spending****,* ***Saving****, or* ***Debt Repayment****. Another approach is to restrict the analysis to respondents who selected* ***exactly one*** *of the three main options for clarity. We will document whatever labeling rule we choose. (If the project allows multi-label classification, we could model it as three binary labels, but a single-label multi-class approach is more straightforward for an ML conference paper, so we proceed with exclusive classes.)*
* ***Encode the target:*** *Create a new column for stimulus usage category as a categorical variable with three values (e.g., Usage = {Spending, Saving, Debt}).*

***Feature Selection and Engineering:*** *Identify the* ***socioeconomic and financial features*** *in the NFCS data that might predict stimulus usage. These include:*

* ***Demographic variables:*** *age, gender, education level, employment status, income bracket, marital status, number of dependents, etc. These often influence financial behavior and will serve as input features. For example, income and employment could strongly affect whether someone could afford to save the stimulus or needed to spend it.*
* ***Financial situation variables:*** *Indicators of financial stress or stability, such as whether the person experienced a job loss or furlough due to COVID (question J52), whether they had an emergency fund, difficulty paying bills, outstanding debts, credit score self-rating, etc. The NFCS includes questions on financial fragility and credit management which can be highly relevant. For instance, someone who was laid off or had a large income drop during the pandemic might be more likely to spend the stimulus on essentials or debt, whereas someone financially secure might invest or save it.*
* ***Financial literacy and attitudes:*** *The NFCS typically includes questions testing financial knowledge and assessing attitudes (like risk tolerance or future orientation). We can derive a financial literacy score (e.g., count of correct answers on standard financial literacy quiz questions) and include attitude responses (e.g., comfort with risk, or self-reported financial anxiety) as features. These could influence behavior (a more financially savvy individual might be more inclined to save or invest the money).*
* ***Geographic and contextual features:*** *Since the survey is state-by-state, each respondent’s state (or region) could be included to capture any regional economic differences. However, using state as a feature might raise complexity (50+ categories) and policy audience might find state-level results interesting. We might one-hot encode region (e.g., Northeast, Midwest, etc.) or include state with caution. Also, consider time-related context if any (though the survey was largely one time period in 2021).*

*As we explore the dataset, we may perform* ***additional feature engineering*** *such as: combining related features (for example, creating a “financial vulnerability” index by combining several survey items), bucketing continuous variables into categories (e.g., age groups or income tiers if not already grouped), or creating dummy variables for categorical responses. Ensure to label and document each engineered feature. By the end of this step, we should have a clear understanding of the available data and a* ***feature matrix X and target vector y*** *ready for preprocessing. It’s also useful at this stage to note any quirks (like many “Prefer not to say” in income) that will need addressing in the next step.*

***2. Data Cleaning and Preprocessing***

*Before modeling, we must clean the data and transform it into a usable form for machine learning:*

***Handling Missing and Invalid Responses:*** *The NFCS dataset uses specific codes for non-responses (e.g., 98 = “Don’t know”, 99 = “Prefer not to say”). We will convert these codes to actual missing values (NaN) in our dataset for any features we plan to use. This applies to both features and the target: for example, if some respondents answered J51 with “Don’t know” or refused (code 98/99), those entries should be dropped or marked as missing for the target, since we cannot classify them. We will likely* ***drop respondents with no valid usage label*** *(98/99 or those who didn’t receive a stimulus), as noted in Step 1. For feature variables, if a particular respondent has a few missing feature values, we can impute or exclude those fields as appropriate. If certain variables have a large proportion of missing or “Prefer not to say”, we might decide to omit those features entirely to avoid bias or excessive imputation.*

***Data Imputation:*** *Decide on a strategy to fill or handle missing values in features:*

* *For numeric features (e.g., age, income if given as a number, amount of emergency savings, etc.), we can impute missing values with a sensible statistic such as the median or mean, or create a special category/bin if the number is coded (for instance, if income is categorical brackets, treat “Prefer not to say” as its own category or impute based on similar respondents). Alternatively, since tree-based models can handle missing values to some extent, we might also leave them as missing if using XGBoost which has built-in handling. But for clarity and consistency, explicit imputation is preferable.*
* *For categorical features (e.g., education level, employment status), one approach is to treat “Don’t know/No answer” as an additional category level (“Unknown”). However, if the count of unknowns is small, we might drop those records or impute based on mode of that feature. We will document how many data points are lost or altered due to cleaning to ensure transparency.*

***Encoding Categorical Variables:*** *Transform categorical features into a numerical format suitable for ML algorithms:*

* ***One-Hot Encoding:*** *For nominal categorical variables with no intrinsic order (e.g., race/ethnicity, state, marital status), apply one-hot encoding (creating binary indicator columns for each category). This avoids imposing any ordinal relationship where none exists. We need to be careful to avoid the dummy variable trap (though tree-based models are not sensitive to multicollinearity, one-hot is fine).*
* ***Ordinal Encoding:*** *For ordinal categories (e.g., education level if coded from 1 = less than high school up to, say, 7 = postgraduate degree, or self-reported financial status on a scale), we can map them to integers reflecting their order. For example, education might be encoded as 0,1,...n in increasing level. Ordinal encoding preserves the rank information. If the model is tree-based, it will treat these as numeric splits which is acceptable. (If we were using linear models, we’d consider one-hot even for ordinal if the relationship is non-linear, but tree models can find splits on the ordinal values appropriately.)*
* ***Binary Variables:*** *Some features are yes/no questions (e.g., “Have a retirement account?”, “Experienced income drop in 2020?”). These can be directly encoded as 0/1 indicators (with 1 = “yes”, 0 = “no”). Ensure that any “Not applicable” or skipped cases are handled (e.g., if a question was only asked to a subset, those outside the subset might be coded differently).*

*We will create a consistent preprocessing pipeline using tools like pandas or scikit-learn’s ColumnTransformer to apply these encodings systematically. This pipeline will help ensure the same transformations are applied during model training and future inference.*

***Normalization/Scaling:*** *For many machine learning models, scaling features to a common range is important.* ***However, for tree-based models like Random Forest and XGBoost, scaling is generally not required*** *– these models are not distance-based and are invariant to monotonic transformations of features. Therefore, we may skip normalization for tree models. If we include any model that is distance-based or uses gradient descent on feature values (like logistic regression or neural network as a baseline), we will apply standardization (zero mean, unit variance) or min-max scaling to the numerical features. In our case, since the focus is on tree ensembles, we can simply leave numeric features in their original units, but it’s a good practice to at least ensure no feature has extreme outliers or an unduly large scale that could cause numeric instability. We will inspect distributions for outliers (e.g., an extremely high income value or age) and possibly cap or transform highly skewed data (like take log of income if income were a continuous variable; in survey data it might already be categorical).*

***Dealing with Class Imbalance:*** *It’s likely the three usage categories (Spending, Saving, Debt) are not evenly distributed. From NFCS reports, we know many people spent the money on bills (a majority) while a substantial portion saved or paid debt (). For example, about 59% reported spending on purchases/bills, 38% added to savings, 33% paid down debt (note: these percentages sum >100% because of multiple responses) (). After assigning a single label per person, we might find “Spending” is the largest class and “Saving” or “Debt” smaller. We will check the balance: say our labeled data has X% Spenders, Y% Savers, Z% Debtors. If the classes are imbalanced (which they are likely to be, e.g., maybe more spenders than savers), we will plan for this in modeling and evaluation. At the preprocessing stage, one option is to* ***resample*** *the training data to balance classes (e.g., using techniques like oversampling the minority classes, such as duplicating or SMOTE, or undersampling the majority class). However, we must do resampling carefully only on the training folds to avoid data leakage (not on the entire dataset before cross-validation). We will likely use* ***stratified splits*** *for cross-validation and train/test to maintain the class distribution, and consider using model algorithms that can incorporate class weights (both RandomForest and XGBoost can accept class weight or scale parameters to handle imbalance). We might hold off on actual resampling until we see if the models can handle the imbalance via weighting. The plan is to be mindful of this issue and address it during model training if needed (discussed further in Step 5).*

*By the end of this step, we will have a cleaned and transformed dataset ready for analysis: a feature matrix (with all features numeric) and a clean target vector. We should also split the data into training and testing sets (e.g., 80% train, 20% test)* ***before*** *extensive analysis to avoid leaking information from test set. We will use a* ***stratified split*** *to ensure the test set has a representative distribution of the three usage categories. The training set will be used for EDA, model training, and cross-validation, while the test set will be held out for final evaluation.*

***3. Exploratory Data Analysis (EDA)***

*With the data prepared, perform a thorough exploratory data analysis to understand patterns and relationships in the dataset. This step helps inform feature selection, spotting anomalies, and providing context for modeling results:*

***Target Variable Exploration:*** *Examine the distribution of the stimulus usage categories in our labeled data. Count and plot the number (and percentage) of respondents classified as* ***Spending****,* ***Saving****, and* ***Debt Repayment****. A bar chart can illustrate the class frequencies. This will confirm the class imbalance extent. For instance, we expect “Spending” to be the largest category. If the data labeling followed a priority rule, double-check that distribution against known multi-response rates (e.g., if a lot of people did both spending and saving, our single-label distribution might depend on our chosen rule). This step ensures we’re aware of any biases introduced in labeling. We can also compare our sample’s distribution to external reports as a sanity check (e.g., NFCS reported 59% spent on bills, 38% saved, 33% paid debt (); our single-label counts might show a somewhat similar pattern, albeit with each person in only one category).*

***Univariate Analysis of Features:*** *For each feature, examine its distribution and potential relationship with the target:*

* *For numeric features (age, etc.), look at histograms or boxplots. Possibly segment these by usage category: e.g., boxplots of age for spenders vs savers vs debt-payers, or the average age in each category. Do older respondents tend to save more? Does income differ among the groups?*
* *For categorical features (education, employment, etc.), create frequency tables or bar charts by stimulus usage category. For example, proportion of each usage category by education level: this could reveal if higher-educated individuals were more likely to save or invest their stimulus, whereas those with less education (which might correlate with lower income) perhaps mostly spent it. Similarly, examine employment status vs usage: e.g., those unemployed or furloughed might have a higher tendency to spend on immediate needs or debt.*
* *For binary features (e.g., whether someone had emergency savings, whether they felt financially anxious, etc.), compare percentages between the groups. If 80% of “savers” had an emergency fund vs 50% of “spenders”, that’s a notable pattern.*

*These analyses can be visualized for clarity: e.g., stacked bar charts for categorical vs target, or violin plots for numeric vs target, etc. The goal is to identify which features show promising separation between the classes. We will note any* ***correlations*** *or trends. For example, we might observe that individuals with higher incomes or higher education were disproportionately represented in the* ***Saving*** *category, while those with income disruptions or higher financial stress skew towards* ***Spending*** *or* ***Debt****. These insights not only guide our modeling (which features to include or focus on) but also provide context for our results and might be reported in the paper’s descriptive statistics or motivation.*

***Bivariate Relationships and Feature Correlation:*** *Compute correlation matrices (or contingency tables for categorical features) to see how features relate to each other, and possibly to avoid multicollinearity issues. While tree-based models don’t require us to remove correlated features, understanding feature relationships is still useful. For example, income and education might be highly correlated; including both is fine, but we should be aware interpretation-wise that they convey overlapping information. If using any models that assume independence or if we later use logistic regression, we might consider dimensionality reduction or dropping one of a correlated pair. Also, check if any features have very low variance (near-constant) or a lot of missing data, which might warrant dropping them before modeling.*

***Initial Insights and Hypotheses:*** *Based on EDA, formulate hypotheses about what drives different stimulus usage behaviors. For instance: “Perhaps younger respondents or those with lower financial literacy tended to spend the stimulus, whereas older or wealthier respondents saved it.” Or “People who had high pre-existing debt or were laid off might prioritize debt repayment.” We may not know if these will hold true in the model, but these observations can motivate features in the model and provide a storyline for the paper’s results. Additionally, EDA might reveal if the three classes have overlapping characteristics or are quite separable. If we find that “Savers” and “Debtors” look very similar on most features (just as an example), then distinguishing them might be challenging for the model – this can set expectations about model performance.*

*Throughout EDA, generate clear* ***visualizations*** *(figures) that could potentially be used in the paper to illustrate the data. For example: a bar chart of stimulus usage categories overall, or a histogram showing income distribution colored by usage type, etc. Save these plots for reference. They might be included in an “Experiments” or “Data” section of the paper to provide context to readers (especially since policy-interested readers will want to see some descriptive stats of who did what with their stimulus).*

***4. Model Design and Training***

*With a solid understanding of the data, we proceed to designing the classification model pipeline and training models. The plan is to treat this as a* ***supervised multi-class classification*** *problem (three classes as defined). Key steps in this stage:*

***Defining the Modeling Pipeline:*** *We will set up a machine learning pipeline that includes our preprocessing (from Step 2) and the classifier. Using a pipeline ensures that any data transformations are systematically applied during cross-validation and future predictions. For example, using scikit-learn, we can create a pipeline that first applies the encoding/imputation and then fits the model. This prevents data leakage and keeps the process reproducible. Since we have already performed and saved preprocessing steps, we may implement the preprocessing in code (with classes for imputers and encoders) and attach it to the model training process.*

***Baseline Model for Benchmarking:*** *As a baseline, start with a simple classifier like* ***Logistic Regression*** *or a basic* ***Decision Tree****. This gives a point of comparison for the more sophisticated models (XGBoost, Random Forest). A logistic regression (with appropriate regularization) will provide a linear model benchmark – its coefficients can also offer an initial sense of feature importance (though in a linear way). A decision tree (depth-limited) can serve as another baseline, being an interpretable model. Training these baseline models on the training set (with cross-validation for logistic to tune regularization strength, for instance) will establish a baseline accuracy or F1-score to beat. We don’t expect the baseline to be highly accurate if the problem is complex, but it is important to include in a research paper to show the value added by more complex models.*

***Advanced Models – Random Forest and XGBoost:*** *The core models we plan to use are ensemble tree methods:*

* ***Random Forest (RF):*** *An ensemble of many decision trees trained on bootstrapped samples of the data and feature subsets, which tends to improve generalization by reducing variance. We will use an RF classifier (e.g., from scikit-learn) and tune key hyperparameters like the number of trees (n\_estimators), max depth of trees, and maybe minimum samples per leaf, etc. Random Forests can handle categorical features (via the one-hot encoding we provide) and are robust to outliers and irrelevant features. They also provide a straightforward feature importance measure (the Gini importance) which we can examine later (though we will rely more on SHAP for nuanced importance).*
* ***XGBoost:*** *eXtreme Gradient Boosting is a powerful gradient-boosted tree algorithm known for its performance and speed improvements (* [*Interpretable machine learning with tree-based shapley additive explanations: Application to metabolomics datasets for binary classification - PMC*](https://pmc.ncbi.nlm.nih.gov/articles/PMC10159207/#:~:text=correcting%20errors%2C%20with%20the%20base,penalize%20complexity%20and%20prevent%20overfitting) *). XGBoost builds trees sequentially, each trying to correct errors of the previous, and includes regularization to prevent overfitting (* [*Interpretable machine learning with tree-based shapley additive explanations: Application to metabolomics datasets for binary classification - PMC*](https://pmc.ncbi.nlm.nih.gov/articles/PMC10159207/#:~:text=the%20other%20hand%2C%20XGBoost%20is,penalize%20complexity%20and%20prevent%20overfitting) *). We choose XGBoost because of its success in structured data competitions and its ability to handle mixed data types effectively. We’ll use the XGBoost classifier (likely via the xgboost Python library or through scikit-learn API) on our data. Important hyperparameters to tune include n\_estimators (number of boosting rounds/trees), max\_depth of each tree, learning\_rate (step size shrinkage), subsample (fraction of training samples to use per tree), colsample\_bytree (fraction of features per tree), and the regularization parameters (gamma, lambda, alpha for XGBoost’s internal regularization).*

***Hyperparameter Tuning:*** *We will perform systematic hyperparameter tuning for the models to optimize performance:*

* *Use* ***cross-validation on the training set*** *(for example, 5-fold or 10-fold stratified CV) to evaluate different hyperparameter combinations. We can use a grid search or random search strategy. For instance, for Random Forest, explore n\_estimators (e.g., 100, 200, 500), max\_depth (e.g., None or 5, 10, 15), and maybe min\_samples\_leaf (e.g., 1, 5, 10) to prevent overfitting on trivial splits. For XGBoost, we’ll explore combinations of learning rate (e.g., 0.01, 0.1, 0.2), max\_depth (3, 6, 9), and number of trees (50, 100, 200), along with perhaps subsample fractions. Given the search space can be large, a random search or Bayesian optimization (e.g., using libraries like Optuna or Scikit-Optimize) could be more efficient than an exhaustive grid. The goal is to find a reasonably good set of parameters for each model that yields strong cross-validated performance.*
* *Use* ***stratified CV*** *to ensure each fold has a balanced representation of usage classes. We’ll focus on metrics like balanced accuracy or macro-average F1 during tuning (rather than plain accuracy) to ensure the model isn’t just optimizing the majority class performance.*
* *The hyperparameter tuning process will be documented (which parameters tried, how we selected the best model). We may include a table in the paper’s appendix or main text summarizing the chosen hyperparameters.*

***Training Final Models:*** *After tuning, train the final Random Forest and XGBoost models on the* ***entire training set*** *using the best-found hyperparameters. We might also include the logistic regression baseline trained on the full training set for final comparison. Ensure to incorporate any class weighting if needed (e.g., XGBoost allows setting a scale\_pos\_weight for imbalance in binary classification; for multi-class, one can equivalently give higher weight to under-represented classes in the objective function or sample more of them – we will have addressed imbalance either by weighting or earlier resampling). The output of this step will be* ***trained model artifacts*** *ready to be evaluated on the hold-out test data.*

*Throughout training, monitor for any signs of overfitting. For example, compare cross-validation performance to training performance. If a model is significantly overfitting (e.g., huge depth trees memorizing training data), consider regularization (prune tree depth, add regularization in XGBoost, or increase min\_samples\_leaf in RF). Our hyperparameter tuning should handle this, but we will keep an eye on it. We may also consider an alternative model if needed (for instance, if both RF and XGBoost perform similarly, that’s fine – if one clearly outperforms, we’ll focus on that one for interpretation in the next steps). At this stage, we’ll also start thinking about how to interpret these models – since they are not simple linear models, we have planned to use SHAP, but Random Forest feature importances can be checked as a quick sanity check (which features are coming out on top? Do they make intuitive sense given the EDA?).*

***5. Model Evaluation***

*After training, we need to thoroughly evaluate the models to understand their performance, robustness, and any weaknesses. This involves both quantitative metrics and qualitative error analysis:*

***Performance on Test Set:*** *Use the hold-out test set (kept aside from training) to evaluate the final models (Random Forest and XGBoost, and the baseline for comparison). Since this is a multi-class classification, we will compute metrics such as:*

* ***Accuracy:*** *The overall fraction of correct classifications. This gives a quick sense of performance but can be misleading if classes are imbalanced (a model predicting everyone as “Spending” might get high accuracy if “Spending” is majority).*
* ***Precision, Recall, F1-Score for each class:*** *Compute the precision (percent of predicted X that were correct) and recall (percent of actual X that were correctly predicted) for each of the three classes, as well as the F1-score (harmonic mean of precision and recall) per class. This gives insight into how well the model does on each category. For example, is it struggling to identify “Debt” cases (perhaps due to fewer examples)? We will likely present a* ***classification report*** *with these values.*
* ***Macro-averaged F1 and Weighted F1:*** *Macro-average treats all classes equally, while weighted accounts for class frequencies. Macro F1 is a good indicator of overall model ability across classes (especially important if we care about minority classes performance). We aim for a high macro-F1 to ensure the model is learning all classes, not just the majority.*
* ***Confusion Matrix:*** *Construct a confusion matrix for the test results, showing how often each actual class was predicted as each class. This helps identify systematic errors (e.g., perhaps many “Debt” cases are misclassified as “Spending” – meaning the model confuses those two categories often). The confusion matrix can be visualized as a heatmap in the paper to illustrate where the model is doing well or making mistakes.*

***Cross-Validation Results and Variance:*** *We will also look at the cross-validation results from training to ensure the model’s performance was consistent across folds. If there was high variance, we might report the standard deviation of performance across folds, indicating the model’s robustness. If one model (say XGBoost) consistently outperformed another (RF) in cross-val and test, we will primarily use that model for our interpretation in the next steps. However, if they are close, we might include both or choose the simpler one for discussion.*

***Addressing Class Imbalance in Evaluation:*** *If the results show that one class has much lower recall than others (common with imbalanced data), we will mention how we handled it. For instance, if “Saving” cases were few and the model’s recall on “Saving” is low, we might go back and try to improve that by adjusting the classification threshold or using oversampling in training. One technique could be to adjust the probability threshold for each class (especially if using a one-vs-rest approach or analyzing predicted probabilities) to maximize a metric like macro-F1. However, with tree ensembles in multi-class, typically we take the argmax class. Instead, we might incorporate* ***class weights****: e.g., give a higher weight to the loss of misclassifying a “Saving” instance to force the model to pay more attention to that class. We will document if we employ such weighting and check if it improves the balanced performance.*

***Error Analysis:*** *Dive deeper into the errors the model makes. We’ll take some misclassified examples and analyze why the model might have gotten them wrong. Perhaps there are certain subgroups for which predictions are frequently incorrect. For example, maybe the model confuses* ***Saving vs Investing*** *(though we didn’t explicitly classify “Investing” as a separate class, if some savers also invested, the model might mislabel some savers as spenders or vice versa). We will see if errors have any pattern, such as if younger savers are mispredicted as spenders, etc. This can hint at either feature interactions we missed or the inherent noise in the data (some people’s behavior might just be hard to predict). This kind of analysis not only strengthens the paper (showing we did a thoughtful evaluation) but also guides the interpretability analysis – we might specifically explain some of these tricky cases with SHAP/LIME in the next step to see what the model was thinking.*

***Model Comparison:*** *Compare the performance of Random Forest vs XGBoost vs baseline. We expect XGBoost to perhaps edge out RF in accuracy due to better optimization, but if the difference is small, the more interpretable model (RF or even logistic) could be favored for simplicity. If one model significantly outperforms, that becomes our* ***primary model*** *for explanation and deployment. In the paper, we’ll likely present a table of results showing all models and their metrics (accuracy, macro F1, etc.), highlighting the best model’s performance. We will also test for* ***statistical significance*** *of differences if applicable (for example, using a paired test on errors or confidence intervals from cross-val). This is more advanced, but including a statement like “XGBoost outperforms the logistic regression baseline by X% in accuracy and Y points in macro-F1, indicating the non-linear relationships captured by the tree-based model are important (we performed a McNemar’s test which indicated the improvement is statistically significant at p<0.05).” adds rigor.*

***Dealing with Overfitting/Underfitting:*** *If the test performance is much lower than training, we have overfitting. We would then consider going back to adjust hyperparameters (e.g., more regularization, simpler model). If performance is uniformly low, maybe the features aren’t predictive enough or the problem is inherently noisy – we might consider adding more features or using an ensemble of models. Given this is a well-studied survey, it’s likely we can get decent predictive power, but we will calibrate expectations. Realistically, predicting human financial behavior might yield moderate accuracy, not extremely high, but that’s acceptable if we can still extract insightful patterns. We will clearly state the achieved performance and acknowledge any limitations.*

*By the end of this step, we should have a* ***validated model*** *(or models) with known performance levels. This sets the stage for interpretation: now that we trust the model’s predictions to some extent, we can explore why it made those predictions using interpretability tools.*

***6. Interpretability and Explainability***

*To ensure our model’s predictions are transparent and to extract insights for policy and research, we will apply interpretability techniques, primarily* ***SHAP*** *and* ***LIME****, to explain the model’s behavior both globally and locally:*

***Shapley Additive Explanations (SHAP):*** *We will use SHAP to understand feature importance and effects. SHAP assigns each feature a contribution value for each individual prediction, based on the concept of Shapley values from cooperative game theory. One advantage of SHAP is that it provides* ***consistent global and local interpretability****, unifying ideas from other methods like LIME with solid theoretical guarantees (* [*Interpretable machine learning with tree-based shapley additive explanations: Application to metabolomics datasets for binary classification - PMC*](https://pmc.ncbi.nlm.nih.gov/articles/PMC10159207/#:~:text=This%20class%20of%20models%20is,the%20unique%20properties%20of%20Shapley) *). In practice, we’ll use the TreeSHAP implementation (which is optimized for tree-based models) on our trained Random Forest or XGBoost model. Key steps with SHAP:*

* ***Global Feature Importance:*** *Calculate the mean absolute SHAP value for each feature across all instances to rank features by overall impact on the model’s output. This will produce a ranking of which features the model found most influential in classifying stimulus usage. For example, we might find that* ***income*** *or* ***job loss status*** *are top features. This global importance is more reliable than the raw feature importance from the Random Forest (which can be biased by feature cardinality or collinearity), and it also tells us directionality (via the SHAP value sign in each class context).*
* ***SHAP Summary Plot:*** *We will create a SHAP summary plot (beeswarm plot) that shows the distribution of SHAP values for each feature, colored by feature value. This visualization will let us see not only importance but how each feature affects the prediction. For instance, we might observe that for the feature* ***Income****, higher income has positive SHAP values toward the “Saving” class (indicating it pushes predictions towards classifying the person as a saver), whereas low income has negative SHAP values (pushing towards “Spending”). Similarly, a feature like* ***Financial hardship (difficulty paying bills)*** *might have positive SHAP values for “Spending” (meaning if you had trouble paying bills, the model is more likely to predict you spent the stimulus, which makes intuitive sense). These patterns provide a* ***global explanation*** *of the model’s learned relationships.*
* ***Class-Specific Analysis:*** *Since this is multi-class, we can condition SHAP analysis on output classes. We might compute SHAP values for each class or look at the SHAP values difference when considering one class vs the rest. Another approach is to binarize (one-vs-all) for explanation purposes: e.g., train separate models or use SHAP interaction values. But likely, analyzing the one model’s SHAP can still reveal, for example, which features contribute to someone being classified as “Saver” vs “Spender”. We will interpret accordingly: e.g., “Feature X has a high positive impact on identifying savers, while Feature Y is most important for distinguishing debt repayers.”*
* ***Local Explanations with SHAP:*** *Choose a few representative individuals (or interesting test cases) from each class and examine their SHAP values. For an individual who was predicted (correctly) as a* ***Saver****, list the top 3 features that had the highest positive SHAP values pushing the model towards “Saving” for that person, and features that pushed against it. We might find, for example, “Person A was predicted to save their stimulus because they had a high income, stable job, and already had emergency savings (these features contributed strongly to the ‘Saving’ prediction), whereas the model was pushed slightly away from ‘Saving’ by the fact that they had some credit card debt (which might have suggested a need to pay debt).” Likewise, do this for a* ***Spender*** *(maybe features: low income, high financial anxiety, no emergency fund contributed to the model predicting they’d spend the money on immediate needs) and a* ***Debtor*** *(features like high existing debt, or perhaps older age with mortgage could push towards using stimulus for debt). These narrative explanations, backed by SHAP values, provide intuitive understanding of what the model has learned.*

***Local Interpretable Model-Agnostic Explanations (LIME):*** *While SHAP provides consistent and global-local interpretation, we will also employ LIME for additional local explanation, as a check and for illustration. LIME works by perturbing features of an instance and training a simple interpretable model (like a linear model) around that instance to approximate the complex model’s behavior in that local region. For one or two specific data points (perhaps the same ones we used for SHAP local analysis), we will apply LIME to see which features it highlights. For instance, using LIME on a particular individual who the model predicted as “Debt Repayment” might yield a simple explanation like “because their income is medium, they have high debt, and they did not report difficulty paying bills (suggesting they had some capacity to use extra money to pay off debt).” We will compare the LIME explanation with SHAP for consistency. Ideally, they should agree on the main influential features for that prediction. If they do, it boosts confidence that our interpretation is not an artifact of one method. If there’s discrepancy, that’s worth noting (often due to LIME’s local linear approximation vs SHAP’s exact Shapley value). According to literature, SHAP’s approach satisfies certain desirable properties and unifies the local surrogate idea of LIME with game-theoretic values (* [*Interpretable machine learning with tree-based shapley additive explanations: Application to metabolomics datasets for binary classification - PMC*](https://pmc.ncbi.nlm.nih.gov/articles/PMC10159207/#:~:text=This%20class%20of%20models%20is,the%20unique%20properties%20of%20Shapley) *), so we expect SHAP to be our primary tool. LIME, however, is a nice pedagogical addition and can be easier to present to non-technical audiences (as it gives a straightforward linear equation or weighting for a single prediction).*

***Global vs Local Explanations:*** *We will clearly distinguish between global importance (which features generally matter most) and local explanations (why a specific prediction was made). Both are important: the former for understanding overall trends and key drivers, and the latter for trust and anecdotal validation on individual cases. For global, besides SHAP, we can also mention the Random Forest feature importance rankings and see if they line up with SHAP’s findings (often they will for the top features, but SHAP may shuffle the order if some features are used in interactions). If there are partial dependence effects that are interesting, we could include Partial Dependence Plots (PDPs) or Individual Conditional Expectation (ICE) curves for top features to illustrate the marginal effect of a feature on the prediction. For instance, PDP for income might show that as income increases, probability of “Spending” goes down and “Saving” goes up, plateauing after a point – a complementary view to SHAP. These are not explicitly requested, but they serve as additional explainability tools if needed.*

***Validating Interpretations:*** *We should ensure the interpretations align with domain understanding. If the model suggests something counter-intuitive (say, it found that higher financial literacy* ***increases*** *chance of spending – which would be surprising), we will double-check if that’s a genuine pattern or a spurious correlation. This might involve slicing the data and verifying the relationship. Our aim is to use interpretability not just to explain the model but to* ***glean insights about real-world behavior****. For policy implications, these insights are gold – they tell us what characteristics associate with different uses of stimulus money. We will highlight such findings in the results/discussion.*

*By the end of this step, we will have a set of* ***explanatory results****: feature importance rankings, SHAP plots (global and possibly some local visualization like force plots or decision plots), and possibly LIME explanations. These will feed directly into the creation of figures for the paper and the narrative of which factors influence stimulus usage behavior according to our model. We’ll ensure to save the SHAP values and any LIME outputs so we can integrate them into the analysis and writing seamlessly.*

***7. Visualization and Insight Extraction***

*This step focuses on creating clear visualizations of our results and distilling key insights from them, which will be crucial for communicating in the paper (especially to an audience that may include policy-makers who appreciate intuitive visuals and explanations):*

***Key Plots to Generate:***

* ***Feature Importance Bar Chart:*** *A bar chart (or horizontal bar chart) of the top 10 (or 15) features by importance. This could be based on SHAP average absolute values (preferred for accuracy) or Random Forest Gini importances for simplicity. Each bar’s length shows the importance magnitude. For example, it might show “Income level” as the longest bar, followed by “Job Loss in 2020”, “Emergency Savings”, etc. This plot gives readers a quick snapshot of what factors were most predictive in the model.*
* ***SHAP Summary (BeeSwarm) Plot:*** *This plot shows each feature’s impact distribution on the model output. We will include this in the paper to provide a more nuanced view than the bar chart. Each feature appears on the y-axis, and dots representing individual respondents are plotted on that feature’s line, with color indicating the feature value (e.g., low to high). Position on the x-axis shows the SHAP value (impact on model output toward one class or the other). We might need to adapt it for multi-class; one way is to plot SHAP for a specific class (say “Saving” vs others) or use separate plots per class. To keep things simple for the paper, we could transform the multi-class problem into a binary for visualization (for instance, look at SHAP values distinguishing “Spending” vs “Not Spending”, etc.), or focus on one class of interest at a time. Alternatively, use the SHAP library’s support for multi-class which can produce such plots by specifying an output dimension. This visualization yields insights like, “Red (high) income points are mostly on the right (positive SHAP for saving), whereas blue (low income) points are on the left (negative SHAP for saving), indicating higher income strongly increases likelihood of the saving class”. We will annotate and describe such observations.*
* ***Confusion Matrix Heatmap:*** *A matrix diagram showing predicted vs actual classes with color intensity. This will highlight, for example, that most spenders were correctly identified (the cell [Spending actual, Spending predicted] is bright), and where the model errs (maybe a somewhat bright off-diagonal at [Debt actual, Spending predicted] indicating debt-users often got misclassified as spenders). Such a figure is useful in the paper to discuss performance and which categories are harder to distinguish.*
* ***Example Explanation Figure:*** *We might include a specific example’s explanation as a figure. This could be a SHAP force plot for one individual (which visually shows feature contributions as forces pushing the prediction towards one class or another). Force plots are a bit complex for print, but perhaps a simplified table or bar chart of top feature contributions for an example might be clearer. Alternatively, a LIME explanation bar chart for one instance (LIME often outputs something like: feature X contributed +0.8 to probability of class, feature Y -0.3, etc). We will decide on one illustrative example – likely a typical scenario – to include as a case study in the paper. For instance, “Case Study: A thirty-year-old unemployed respondent with no emergency savings was predicted to spend their stimulus. The explainability analysis (Figure X) shows that the lack of savings and unemployment status heavily swung the model’s prediction toward Spending, outweighing other factors like education.” This makes the results tangible.*

***Insights Extraction:*** *With these visuals, extract* ***meaningful insights*** *that answer the research questions and could inform policy:*

* *Identify the* ***most influential factors*** *for stimulus usage. From the feature importance, suppose we find factors like income, job loss, education, and financial anxiety are top. The insight might be: “Income and employment disruptions were the strongest predictors of how people used their stimulus funds. Those with higher incomes were far more likely to save the money, whereas those who lost jobs or had lower incomes overwhelmingly used it for immediate expenses or paying down debt.” This can be backed by our SHAP analysis showing those trends.*
* *Understand* ***behavioral segments:*** *Using the combination of features, we can characterize typical profiles. For example:* ***Spenders*** *tended to be individuals with signs of financial strain (lower income, lost income in pandemic, lower savings, perhaps younger or with dependents),* ***Savers*** *tended to be financially comfortable (higher income, less hardship, maybe already had some investment accounts), and* ***Debtors*** *might be a mix (often middle-income with significant debts like credit cards or loans, who chose to reduce those debts). These profiles come directly from seeing which features push predictions towards each class. We might create a table or simply describe these profiles in text.*
* *Check for any* ***surprises****: Did any feature behave unexpectedly? For instance, if we included a feature like “received financial advice” or “financial literacy score”, did it significantly influence usage? If the model found a subtle pattern (like say high financial literacy correlates with investing the stimulus, which we didn’t explicitly classify but might overlap with saving), note that. For policy, if something like financial literacy or anxiety is a strong factor, it suggests educational interventions or addressing financial anxiety could influence how people handle windfalls.*
* ***Visualization for Policy Audience:*** *Ensure that at least some visuals are accessible to a broader audience. The confusion matrix and SHAP summary might be more technical, so in a presentation or appendix we might keep those. The paper can include the feature importance chart and a simpler SHAP-like chart illustrating one feature’s effect (for example, a partial dependence plot of income vs probability of saving could be easily understood: it might show a curve rising with income). We will decide on the final set keeping the narrative in mind.*

*We will also extract numbers from the model for the write-up, such as: “All else equal, our model indicates that a respondent who was laid off in 2020 had a 20% higher likelihood of using the stimulus to pay down debt or bills rather than save it, compared to an otherwise similar respondent who wasn’t laid off” – such a statement can be derived from model what-if analysis or SHAP differences. These kinds of insights tie the model back to real-world interpretation.*

*Finally, we compile the most important insights to emphasize in the* ***Discussion*** *section of the paper. For example: “This analysis suggests the stimulus payments primarily served as relief for those in financial need (who spent it on necessities or debt), while for others it became extra savings. Policy-wise, this supports the idea that broad cash transfers both help those in crisis and allow others to bolster their finances, potentially increasing overall financial resilience (*[*Most Stimulus Payments Were Saved or Applied to Debt | NBER*](https://www.nber.org/digest/oct20/most-stimulus-payments-were-saved-or-applied-debt#:~:text=,used%20to%20pay%20down%20debt)*) ().” We’ll ensure to connect such insights with existing literature or expectations (the citation from NBER showed similar patterns in 2020: ~40% spent, ~60% saved or debt (*[*Most Stimulus Payments Were Saved or Applied to Debt | NBER*](https://www.nber.org/digest/oct20/most-stimulus-payments-were-saved-or-applied-debt#:~:text=,used%20to%20pay%20down%20debt)*), aligning with what we find). All visualizations will be saved in publication-quality formats (PNG/SVG for inclusion in LaTeX, etc.) and properly labeled and referenced in the text.*

***8. Writing and Structuring the Paper***

*With analysis complete, the next step is writing the research paper in a standard ML/AI conference format. We will follow the typical structure of NeurIPS/ICML/AAAI papers and ensure each section is well-crafted:*

* ***Title and Abstract:*** *Create a concise, informative title (e.g., “Classifying Stimulus Payment Usage via Machine Learning: An Analysis of the 2021 NFCS Survey”). The Abstract (usually ~200 words) will summarize the problem context (COVID-19 stimulus and interest in usage behavior), the data and methods (NFCS 2021 dataset, classification with ML models, interpretability via SHAP), and highlight key findings (e.g., which factors predict usage and an interesting result or two) and implications. It should entice readers from both ML and policy backgrounds by mentioning both the technical approach and the real-world significance.*
* ***Introduction:*** *Introduce the problem of stimulus payments usage and why it matters. We will cite statistics or prior work to ground the reader (for example, mention that studies found about 40% of stimulus checks were spent and 60% saved or used for debt (*[*Most Stimulus Payments Were Saved or Applied to Debt | NBER*](https://www.nber.org/digest/oct20/most-stimulus-payments-were-saved-or-applied-debt#:~:text=,used%20to%20pay%20down%20debt)*), highlighting the variability in usage). State that understanding these behaviors is important for economic policy (did the money provide relief vs get saved) and how predictive modeling can add value by identifying patterns and key drivers. Then clearly state our approach: “In this paper, we use machine learning to classify individuals’ usage of their 2020–2021 stimulus payments (spend vs save vs debt repayment) based on their socioeconomic and financial characteristics. Using the 2021 FINRA NFCS dataset, we train ensemble models (XGBoost, Random Forest) to predict usage and employ SHAP values for interpretation.” Also outline contributions: e.g., 1) we demonstrate an ML approach to a novel financial behavior classification problem, 2) we provide insights into the determinants of stimulus usage with interpretability techniques, and 3) this work bridges ML and public policy interests by analyzing a large national survey with state-of-the-art AI methods. The intro should be accessible and motivate both the technical reader (interesting ML application) and the policy reader (important questions answered in a data-driven way).*
* ***Related Work:*** *Include a section that discusses relevant literature in two areas: (a)* ***Studies on stimulus payments usage or financial behavior*** *– e.g., economic surveys (NBER or Fed studies) about how people used stimulus checks, to show what is already known (like “Most households either saved or paid debt with the stimulus (*[*Most Stimulus Payments Were Saved or Applied to Debt | NBER*](https://www.nber.org/digest/oct20/most-stimulus-payments-were-saved-or-applied-debt#:~:text=,used%20to%20pay%20down%20debt)*)”). This sets context and might highlight what our work adds (most prior studies might be descriptive; we add a predictive modeling angle). (b)* ***Machine learning in consumer finance / behavioral classification*** *– discuss any similar ML applications (perhaps ML used on survey data or financial decision predictions). Also reference interpretability in ML, especially for social science applications, citing works that used SHAP or LIME in finance or economics if available. If few direct references, we can cite generic ones (e.g., Lundberg and Lee (2017) for SHAP, Ribeiro et al. for LIME, etc., though in-text we can cite our earlier reference explaining SHAP (* [*Interpretable machine learning with tree-based shapley additive explanations: Application to metabolomics datasets for binary classification - PMC*](https://pmc.ncbi.nlm.nih.gov/articles/PMC10159207/#:~:text=This%20class%20of%20models%20is,the%20unique%20properties%20of%20Shapley) *)). The related work section situates our study at the intersection of financial capability research and ML methodology. It shows we’re aware of what’s been done and how our approach is novel (maybe no one has applied ML to NFCS stimulus question yet, making our work relatively unique).*
* ***Data and Methodology:*** *This section (sometimes split into “Data” and “Methods”) will detail the dataset and our modeling approach.*
  + *Data: Describe the NFCS 2021 dataset (sample size, methodology briefly, what variables it contains) and specifically detail how we constructed the target variable from J51 (explain the “select all that apply” nature and how we derived the spend/save/debt categories). Mention any filtering (we exclude those who didn’t receive stimulus, etc.) and give final sample size used. Provide a table of descriptive statistics: e.g., percent in each class, and maybe means of key features by class (to give readers a sense of differences).*
  + *Feature processing: Summarize the feature set we used (perhaps a table listing the features or groups of features). Explain any encoding or transformations.*
  + *Modeling approach: Explain that we formulated it as a multi-class classification. List the algorithms (logistic regression, random forest, XGBoost) and justify them (e.g., widely used for tabular data, allow interpretation). Describe how we did hyperparameter tuning via cross-validation (mention maybe the use of stratified 5-fold CV and what we optimized). Also note the class imbalance and how we addressed it (say we used class weights or oversampled – whatever we did, as decided earlier). Essentially, this section should be detailed enough that another researcher could replicate our pipeline from data to model training. We’ll ensure to cite any packages or tools (for instance, “we used the XGBoost library” or “SHAP Python library for explanations”).*
  + *Optionally, we might include a brief* ***Mathematical formulation*** *if suitable: defining the classification problem and the objective function of XGBoost, etc., but since this is an applied paper, heavy equations aren’t necessary. A concise description in words suffices.*
* ***Experiments and Results:*** *In a conference paper, we then present the outcomes. This can possibly be split into* ***Results*** *and* ***Discussion****, but given space, often combined. We will likely structure it as:*
  + *Predictive Performance: Present the performance metrics of our models (perhaps in a table). Highlight that XGBoost (for example) achieved the best performance, with [accuracy = X%, macro-F1 = Y]. Discuss how this compares to baseline (e.g., “the XGBoost model improves macro-F1 from 0.60 to 0.75 compared to logistic regression, indicating the importance of non-linear relationships”). If any interesting observation like one class remains harder to predict, mention that.*
  + *Feature Importance & Global Explanations: Present the global interpretation (feature importance plot and/or SHAP summary). Describe the key drivers as found by the model. This is where we answer which features matter: e.g., “Income was the most influential feature, with higher income strongly associated with the model predicting the Saving outcome. Job loss during the pandemic was the next most important factor, pushing predictions towards Spending or Debt when true.” We will cite evidence (the SHAP analysis) to back each statement, possibly referencing our figures. We connect this with intuition or prior work: “This aligns with expectations that those with financial hardship use windfalls for immediate needs, whereas those financially secure treat them as savings ().”*
  + *Local Explanations & Case Studies: Provide a couple of examples to illustrate model decisions. This could be a small sub-section or paragraph. For example: “We examine a respondent who our model predicts as a Saver. She is a high-income earner who did not experience income loss and has a college degree. The model’s explanation (using SHAP) shows that her high income and education contributed positively to the Saving prediction, while her moderate credit card debt slightly pulled towards Debt repayment. Ultimately, the model confidently classifies her as Saving. In contrast, another example of a Spender: a young male with a high school diploma, who lost his job in 2020 and has no emergency fund, was predicted to have spent the stimulus. SHAP highlights the job loss and lack of savings as the top contributors to this prediction.” These concrete examples make the results more relatable.*
  + *(If space) Comparative Analysis: If we tried multiple approaches (e.g., multi-label vs multi-class, or including vs excluding certain features), we might briefly note those and their outcomes. But likely, to keep focus, we stick to the chosen approach.*
  + *Emphasize any particularly novel finding. Maybe we discover something like “financial literacy score was not as important as expected” or “people with moderate incomes were most likely to pay down debt, even more than very low-income folks” etc. Such findings can be mentioned to show the depth of analysis.*
* ***Discussion:*** *This section interprets the results in a broader context and discusses implications, limitations, and future work. We will highlight what the results mean for policy or understanding consumer behavior. For instance: “Our findings suggest that stimulus payments functioned both as relief and as savings enhancement. For financially vulnerable groups (e.g., those who lost jobs or lacked savings), the funds were likely spent on necessities or used to reduce debt – indicating the payments helped ease financial stress in the short term. Meanwhile, more secure households added the funds to savings, which could have longer-term benefits for their financial stability or could indicate the stimulus was more of a windfall than needed income for them. This heterogeneity is important for policymakers: a one-size-fits-all stimulus may have varied impacts on economic demand versus personal balance sheets.” We link this back to any related economic discussions (maybe cite a policy paper or two). Also, in discussion, mention the usefulness of ML here: “The ML model not only achieved good predictive accuracy but also highlighted key predictors of behavior, some of which (like financial anxiety level) might be harder to isolate in standard analyses.”*
  + ***Limitations:*** *Acknowledge limitations: e.g., the NFCS data is self-reported and cross-sectional (not causal), our classification had to simplify a multi-response into single labels (some nuance lost), the model’s accuracy, while good, is not perfect (some unpredictability in human behavior). Also mention that our model might not generalize beyond the survey population or time (conditions unique to the pandemic). If there were important categories like investing or donating that we left out, mention that focusing on the three main categories was a simplification for modeling.*
  + ***Future Work:*** *Suggest future research avenues: using multi-label classification to capture multiple uses of funds, applying the model to other datasets or combining with time-series spending data, incorporating more features like psychological factors, or extending the analysis to see if these behaviors correlate with outcomes (like financial health improvements). Also, perhaps using more advanced ML or causal inference to see the effect of stimulus on outcomes. Since the audience is ML, one might also propose exploring more sophisticated models or fairness analysis (e.g., ensure the model is not biased against certain demographic groups in prediction).*
  + *We might also note potential improvements like collecting more granular data (how the stimulus was split) or using our model for targeted surveys (just suggestions).*
* ***Conclusion:*** *A short final section summarizing the work and its significance. Restate the problem and that we provided an ML solution, highlight one or two main findings again, and conclude with a statement about the contribution (e.g., “This work demonstrates how interpretable machine learning can be applied to nationally representative financial surveys to uncover patterns in economic behavior, providing a tool for both prediction and insight that can inform policy design.”). End on a positive forward-looking note.*
* ***References:*** *We will include a bibliography of all cited works in the appropriate format. This will cover sources like the NFCS documentation, any papers or articles about stimulus usage (like the NBER digest (*[*Most Stimulus Payments Were Saved or Applied to Debt | NBER*](https://www.nber.org/digest/oct20/most-stimulus-payments-were-saved-or-applied-debt#:~:text=,used%20to%20pay%20down%20debt)*)), and ML interpretability references. Ensure the reference list is formatted as required by the conference (likely numeric citations or author-year depending on template).*
* ***Appendices (if allowed):*** *If the conference allows, we might include extra material such as more detailed tables, the exact survey question wording (J51) for clarity, additional plots from EDA, etc. We might also include a link to code or data in the appendix if appropriate (though code release might also just be mentioned in the reproducibility section or a footnote).*

*The writing should be clear and succinct due to page limits (NeurIPS ~9-10 pages). We will iterate to make sure the story flows: from introduction (why we care) to methodology (what we did) to results (what we found) to implications. Also, since the audience includes ML researchers and possibly policy folks, we will avoid overly technical jargon without explanation and equally avoid heavy policy jargon – strike a balance. We’ll likely have someone not involved in the project read a draft to ensure it’s understandable. Finally, we’ll format the paper in the required LaTeX template (ensuring figures and tables are appropriately placed and referenced). Early drafting of each section can be done in parallel with analysis; we’ll refine it after finalizing results.*

***9. Reproducibility and Code Packaging***

*Given the increasing emphasis on reproducibility in ML research, we will make sure our code and data processing are organized and shared to allow others (and reviewers) to replicate our findings:*

***Organizing the Codebase:*** *The analysis code will be structured into clear sections, likely in Jupyter notebooks or Python scripts, corresponding to the steps above (data prep, EDA, modeling, interpretation). To make it reproducible:*

* *We will fix a random seed for any randomness (splitting data, model initialization) to ensure results can be reproduced exactly. For example, using numpy.random.seed(42) and appropriate seed arguments in scikit-learn and XGBoost.*
* *Use a consistent environment (e.g., Python 3.x, with specific library versions: pandas, numpy, scikit-learn, xgboost, shap, lime, etc.). We will create a requirements.txt or environment.yml listing these versions. This ensures anyone setting up the environment can match our setup.*
* *Write a* ***README*** *that explains how to run the code and what order to run things in (e.g., first run the data preprocessing script to generate the cleaned dataset, then the training script, etc.).*

***Jupyter Notebooks and Scripts:*** *During development, we might use notebooks for EDA and trying out ideas (since they’re great for mixing code and visuals). For final packaging, we can clean these notebooks (remove extraneous code, add markdown explanations) and/or convert critical parts to Python scripts for automated runs. We might provide both the notebook (for easy understanding) and a script (for quick execution). Key notebooks could be: EDA.ipynb, Modeling.ipynb, Interpretation.ipynb. In each, we’ll ensure the outputs (plots, metrics) are saved to files (like PNG images for plots, CSV for any important intermediate results).*

***Version Control (GitHub):*** *We will use GitHub to host the project repository. This repository will contain: the code (notebooks/scripts), the README, the environment file, and possibly a subset of data or instructions to get the data (since the NFCS dataset might be too large or require terms, we likely won’t upload the raw data publicly if not allowed; instead we provide a link or instructions for users to obtain it from FINRA’s site (*[*Using the NFCS Data | Finra Foundation*](https://finrafoundation.org/using-nfcs-data#:~:text=The%20FINRA%20Foundation%20encourages%20researchers%2C,can%20also%20be%20separately%20accessed)*)). If possible under data sharing rules, we can include a processed anonymized dataset or at least a sample for demonstration. We’ll also include our paper (LaTeX source and PDF) in the repo once ready, to serve as documentation.*

***Reproducing Results:*** *We will test that a fresh clone of the repo can reproduce the key results (we might do this on a different machine or container). If possible, we might use Docker to containerize the environment, though that might be overkill for a conference paper. Alternatively, use binder or similar for notebooks. But at least with requirements.txt, it should be straightforward. We also plan to adhere to any conference-specific reproducibility checklist; for example, NeurIPS has a checklist requiring specifying things like how hyperparameters were chosen, whether code will be released, etc., which we will fill out truthfully (since we do intend to release code and details).*

***Documentation:*** *Besides the README, we will comment our code clearly, and possibly write a short* ***user guide*** *for replicating each step. For instance, in the repo wiki or README: “Step 1: Download NFCS 2021 data from FINRA (link). Step 2: Run preprocess\_data.py to generate clean\_data.csv. Step 3: Run train\_models.py which will output metrics and save models. Step 4: Run interpret\_model.ipynb to produce SHAP plots.” This way, even someone not deeply familiar with our work can follow along.*

***Packaging for Submission:*** *Some conferences encourage or require submitting code as supplemental material. We will prepare a zip of our code (ensuring no identifying info of authors if double-blind review). The code will be cleaned of any secrets or personal paths. If the dataset cannot be included, we will note that and possibly include a small synthetic dataset for illustration if needed.*

*By prioritizing reproducibility, we ensure that our results are credible. It also helps us internally; as we write the paper, we might need to regenerate a figure or compute an alternate metric, and having everything scripted means we can do so reliably. Post-publication, the reproducibility and open-source aspect allows others (e.g., policy analysts or other researchers) to extend our work, perhaps applying our pipeline to newer data or different cohorts.*

*Additionally, we’ll consider implementing tests or validations in code: for example, after preprocessing, assert that no illegal values remain (no 98/99 codes), or after splitting, assert that the class distribution in train and test are similar. These little checks act as regression tests to catch any issues if we update code.*

*Finally, we might prepare a* ***demo or visualization*** *for broader communication: e.g., an interactive plot or a small web app showing how changing certain features changes the prediction (using the model). This isn’t required for the paper, but it could be useful if presenting to a non-technical audience or even at a poster/demo session. Tools like SHAP have an interactive visualization in notebooks, but something like that could be shared. Regardless, the main focus is to have the code and analysis environment ready for anyone to rerun and verify our work.*

***10. Choosing and Preparing for Publication***

*With everything done, the last step is to decide where to publish and ensure the work meets the target venue’s requirements:*

***Selecting a Venue:*** *The ideal venues mentioned are top-tier ML conferences like* ***NeurIPS, ICML, AAAI****. These are competitive, and our work is an applied research with interpretability – it could fit if we emphasize the novel combination or insights. We should consider the fit: NeurIPS and ICML often look for methodological innovation, but they also have tracks or acceptance for impactful applications, especially if interpretability is highlighted. AAAI also accepts applied AI work. Another consideration: workshops or specialized conferences (e.g., ACM Conference on AI, Ethics, and Society, or workshops on AI for social good, or an economics/financial ML workshop) could be easier to get into and have the right audience. However, we can aim high for a main conference first. We should also consider if we want to target a journal instead (since policy interest is noted, maybe a journal like “Journal of Behavioral Finance” or an interdisciplinary journal could work). But given the instruction, we’ll assume aiming for an ML conference proceedings paper.*

***Formatting and Submission Guidelines:*** *Once we pick (say NeurIPS 2025 as a target), download their LaTeX paper template. Transfer our content into that template, adjusting to meet page limits and formatting rules (e.g., NeurIPS has 9 pages limit for main content, unlimited references; requires specific margin, font size, etc.). Ensure our figures are legible and have proper captions. Follow the citation style as required (likely numeric citations for conferences). We must also ensure the paper is* ***anonymous*** *for double-blind review: remove author names, acknowledgments, and any identifying information (like if our code repo is public, we might anonymize it for review or use a private link). Typically, we would say “Dataset was obtained from FINRA (citation)” rather than “we got the data via our collaboration with X” to not reveal if we had insider access (assuming we didn’t need any privileged access since NFCS is public). Also be cautious that none of our references or writing inadvertently expose us (e.g., “In our previous work we did X” – avoid that in the submission version).*

***Cover Letter and Submission:*** *For conferences, usually a cover letter isn’t needed, but for journals it is. If we go journal route, we’d write a cover highlighting why the paper is suitable for that journal. For a conference, we will just submit via their electronic system by the deadline. We’ll triple-check that our PDF meets all requirements (sometimes they check for fonts, non-identifiable metadata, etc.). We should also prepare a concise* ***keywords*** *or* ***supplementary material*** *if needed (some conferences allow a supplement where we can put extra experiments or the code).*

***Responding to Reviews:*** *Anticipate the kinds of questions or criticisms reviewers might have: e.g., “Is this just an application of known methods without novelty?”, “How do we know the model isn’t overfitting or reflecting spurious correlations?”, “Could simpler statistical methods have done the same thing?”, or “The policy implications could be drawn without ML – what does ML add?”. We should preemptively address these in the paper (in discussion or methodology). Emphasize what was non-trivial: maybe the interpretability approach on a new problem, the insights gained that weren’t obvious a priori, or the rigorous evaluation. If the conference has a rebuttal phase, we’ll be ready to answer any such points with additional analysis if needed.*

***Publication and Beyond:*** *If accepted, great – we will present at the conference (make slides or a poster focusing on the main findings and the interpretability angle, since that’s engaging for audiences). If not, we will consider the feedback and possibly submit to another venue or a journal. Even if the conference route doesn’t pan out, the work can be put on arXiv as a preprint to disseminate it. We would then target a relevant journal or workshop where the work can still reach the interested community (for instance, IEEE or ACM journals on data science, or even an economics journal if we focus more on the findings).*

*Additionally, since there’s policy interest, we might prepare a more accessible summary of the findings – perhaps a blog post or an op-ed style article – that highlights the insights in non-technical terms for policymakers. For example, a short piece titled “Who Spends vs. Saves Stimulus Checks? An AI Approach Yields Answers” that can be shared with think-tanks or on LinkedIn.*

*Finally, we will ensure to adhere to any* ***ethics and privacy considerations*** *required by the publication. Using survey data like NFCS is generally fine (it’s anonymized and public). We will note in the paper that the data is survey self-reported and all analysis was done on de-identified records. If an ethics review statement is needed, we’ll include one stating that this research was reviewed and found exempt (if we needed IRB, though likely not since it’s public data). This shows thoroughness in preparation for publication.*

*With these steps completed, we will have a comprehensive project from data to paper submission. Each step in this roadmap ensures that the research is systematic, rigorous, and well-communicated, increasing its chances of success in an ML research venue and its usefulness to the wider community. IS THIS ANY DIFFERENT THAN WHAT I HAVE DONE SO FAR?*