

LLM vs Data Scientist Feature Selection

Team 16 - Gemini

Numan Suri, Hamza Asad, Asad Kamal

Introduction (Project Statement)

Choosing the right features is an important part of building strong machine learning models, especially when it comes to health data analytics. Proper feature selection increases model interpretability, simplifies computations, and lowers the risk of overfitting by focusing on the most important features. This overall process improves the accuracy of predictions. The Centers for Disease Control and Prevention (CDC)'s National Health Interview Survey (NHIS) is a large, ongoing poll that collects a lot of information about a lot of different health-related factors. With such a large dataset, it's important to choose the right features to find the best predictors for health results like heart disease.

In recent years, advancements in Large Language Models (LLMs) have opened new avenues for automating and enhancing various aspects of data science workflows. Google's Gemini 2.5 Pro, for instance, represents a significant leap in AI capabilities, offering enhanced reasoning and coding functionalities (Olteanu, 2025). This study aims to evaluate and compare traditional human-guided methods of feature and model selection with those driven by Gemini 2.5 Pro's recommendations, focusing specifically on predicting heart disease using the NHIS dataset. The main research questions guiding our project are: can Gemini 2.5 Pro's feature recommendations match or exceed the predictive performance of features selected through conventional statistical and domain-driven data science techniques? How does model performance differ when using Gemini 2.5 Pro's suggested ML model versus traditionally selected ML models? What implications do Gemini-driven recommendations have for interpretability, model efficiency, and practical deployment of predictive models?

Methods

The NHIS questionnaire dataset includes responses from approximately 29,522 adult respondents across diverse demographic and socioeconomic backgrounds in the United States. Specifically, this dataset includes a large amount various variables, such as demographics (like age, gender, and ethnicity), lifestyle behaviors (like smoking, dieting, and exercise habits), socioeconomic indicators (like income and education level), and detailed medical histories (like diabetes, high blood pressure, and cancer diagnoses).

Our study's main variable is a binary sign of heart disease that was made by adding together peoples' self-reported diagnoses of coronary heart disease (CHDEV_A), angina (ANGEV_A), and myocardial infarction (MIEV_A). If respondents said they had any of these conditions, they were considered to have heart disease. If they didn't, they were considered to not have heart disease, which gave our predictive modeling a clinically meaningful base.

To begin with we conducted traditional feature selection techniques to determine which columns had a ($p < 0.05$) we decided on the following 27 columns:

1. HYPEV_A: ever been told you have high blood pressure
2. CHLEV_A: ever told you had high cholesterol
3. DIBLAST1_A: last time blood sugar test, if never told had diabetes
4. EMPLASTWK_A: worked for pay last week
5. ARTHEV_A: ever had arthritis
6. COPDEV_A: ever been told you had COPD, emphysema, or chronic bronchitis
7. STREV_A: ever been told you had a stroke
8. CANEV_A: ever been told you had cancer
9. PRDEDUC1_A: sample adults 18+ with private health insurance - plan 1
10. SMKCIGST_A: cigarette smoking status
11. PLN1PAY4_A: health insurance hierarchy 65+
12. PLN1PAY6_A: sample adults 18+ with private health insurance - plan 1
13. PLN1PAY5_A: paid for by Medicaid - plan 1
14. EDUCP_A: educational level of sample adult
15. DEPFREQ_A: how often depressed
16. RACEALLP_A: single and multiple race groups
17. NOTCOV_A: coverage status as used in Health United States
18. SEX_A : sex
19. URBRL: 2013 NCHS Urban-Rural Classification Scheme for counties
20. BMICAT_A: categorical Body Mass Index, Public Use
21. DEPEV_A: ever had depression
22. ANXFREQ_A: how often feel worried, nervous, or anxious
23. REGION: household region
24. ANXMED_A: take medication for worried/nervous/anxious feelings
25. ANXEV_A: ever had anxiety
26. ASEV_A: ever had asthma
27. AGEV_A: age (years, 18-85 top coded)

In contrast Gemini selected the following features:

1. AGEV_A: age (years, 18-85 top coded)
2. SEX_A : sex
3. HISPALLP_A: single and multiple race groups with Hispanic origin
4. EDUCP_A: educational level of sample adult
5. BMICAT_A: categorical body mass index, public use
6. SMKCIGST_A: cigarette smoking status
7. HYPEV_A: ever been told you have high blood pressure
8. CHLEV_A: ever told you had high cholesterol
9. DIBEV_A: ever had diabetes
10. PHSTAT_A: General health status
11. DEPEV_A: ever had depression
12. COPDEV_A: Ever been told you had COPD, emphysema, or chronic bronchitis.
13. STREV_A: ever been told you had a stroke

For initial feature selection techniques we decided to perform chi-square tests for categorical variables and t-tests for numerical variables in order to determine which features are statistically associated with heart disease. Chi-square tests assess independence between categorical features and their outcome and because the NHIS dataset had several categorical variables (i.e. whether someone has high cholesterol or not) we decided that Chi squared is an appropriate means of determining which factors were relevant predictors. In contrast T-tests compare the means of continuous variables between control groups which provides insights into which values are due to chance and which are statistically significant and meaningful. Our methodology is laid out below:

Chi Squared:

- Created a list of categorical features called `cat_feats`.
- Ran a chi-square test of independence for each of the categorical variables determining if the distribution of values differs between those with and without heart disease.
- Collected the chi-square statistic and p-values.
- Sorted variables by the p-value to prioritize variables that are most strongly associated with heart disease.

T-Tests:

- Identified and isolated continuous variables.
- Ran a Welch t-test to compare mean values between the heart disease and non-heart disease groups.
- Stored and sorted those results.

In terms of overall data preparation, categorical columns were mapped to human-readable labels using the NHIS codebook, and ambiguous survey responses (e.g., “refused,” “don’t know,” 7–9) were treated as missing values. We dropped any columns with over 50% missingness, and imputed remaining missing values using the mode for categorical features and the median for numeric ones. For categorical variables with two or more categories, we applied one-hot encoding using a column transformer. We then split the data using an 80:20 stratified train-test split to preserve the proportion of heart disease cases. Model performance was evaluated using the F1-score due to class imbalance, and we used classification reports and confusion matrices for interpretation. Where applicable, hyperparameter tuning was performed using grid search with 5-fold cross-validation.

On the Generative AI side of the project, we prompted Gemini 2.5 Pro in Google AI Studio to act as a data scientist and select features from the same dataset. The prompt was: *"Hello, you are to act as a data scientist. Please pick features from this CSV file that would give the best binary classification result for predicting heart disease. Data preview: """,* followed by the first two rows of the survey CSV and the full NHIS codebook for context. The objective was to test Gemini’s ability to independently recommend both predictive features and an appropriate column to transform into our binary `heart_disease` target variable. Now, because Gemini provided a

special subset of features, we did not perform manual filtering for missingness on the generative AI-selected features.

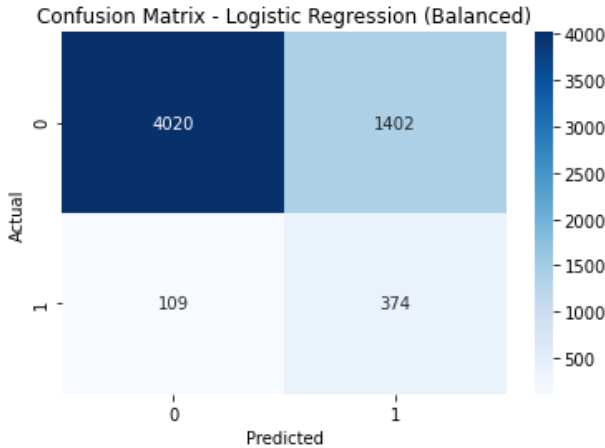
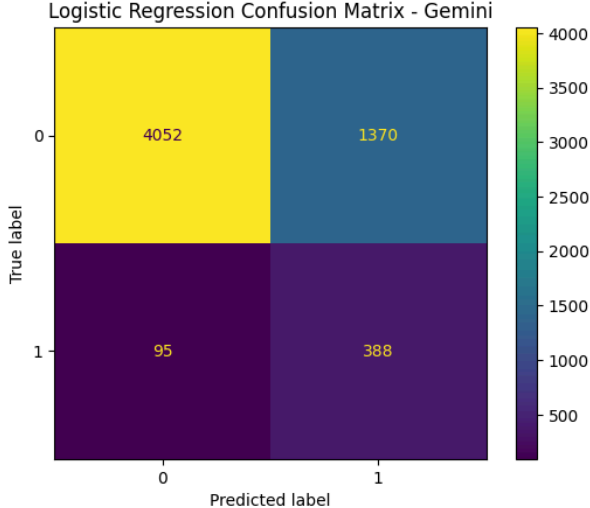
For our machine learning models used on both the human-selected features, we selected Logistic Regression along with three decision tree–based models: Random Forest, CatBoost, and XGBoost. We began with one of the most commonly used machine learning models for binary classification: logistic regression (LR). LR works by estimating the probability that a given input belongs to a particular class, and produces predictions between 0 and 1. In our case, the target variable is heart disease which we transformed into a binary variable, so the classification became choosing “heart disease” or “no heart disease”. Another reason to start with logistic regression is its prior use in medical studies, such as a 2022 paper in which researchers similarly aimed to predict heart disease—albeit using a dataset from the University of California, Irvine and experimenting with various train-test splits including 90:10, 80:20, 70:30, 50:50, and 40:60 (G et al., 2022).

Up next, we chose Random Forest (RF) as the first of our decision tree models due to how well it handles high dimensional data, its robustness to outliers, and generally produces high accuracy results. RF works as an ensemble learning method that combines the outputs of multiple decision trees—using majority voting for classification tasks and averaging for regression tasks—to produce a single, robust prediction (Sruthi, 2025). RF’s ensemble technique helps mitigate overfitting while also improving generalization performance. RF has also been used in medical research; for instance, a 2023 study used it to predict cardiovascular disease based on a dataset sourced from Kaggle (Sumwiza et al., 2023)

Then, CatBoost was chosen because it is one of the most technically capable and interesting ML models to come out in the last few years. It is a gradient boosting algorithm that boasts an exceptional ability to handle categorical data natively, eliminating the need for extensive preprocessing such as one-hot encoding. This was really helpful for us, given the vast majority of our columns were categorical in nature. Additionally, it uses ordered boosting, a technique designed to prevent target leakage by using a specific ordering principle when updating residuals, thereby reducing overfitting (Chepenko, 2019).

Lastly, the final model we used was XGBoost which is generally known to have a strong performance with structured/tabular data, scalability to large datasets (our dataset contained nearly 30,000 records), and robustness to imbalanced classes. Like CatBoost, XGBoost also handles missing values natively, making it particularly suitable for medical datasets and survey-based inputs where incomplete responses are common (Sriraman, 2024). One of the ways it is able to do this is that both L1(Lasso) and L2 (Ridge) regularization are integrated into its objective function (Ultralytics, n.d.).

Results

Traditional	Gemini 2.5																																																
Logistic Regression																																																	
<table><tr><td></td><td>precision</td><td>recall</td><td>f1-score</td></tr><tr><td>0</td><td>0.97</td><td>0.74</td><td>0.84</td></tr><tr><td>1</td><td>0.21</td><td>0.77</td><td>0.33</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.74</td></tr><tr><td>macro avg</td><td>0.59</td><td>0.76</td><td>0.59</td></tr><tr><td>weighted avg</td><td>0.91</td><td>0.74</td><td>0.80</td></tr></table> <p>Table 1a: Logistic Regression - Classification Report</p>		precision	recall	f1-score	0	0.97	0.74	0.84	1	0.21	0.77	0.33	accuracy			0.74	macro avg	0.59	0.76	0.59	weighted avg	0.91	0.74	0.80	<table><tr><td></td><td>precision</td><td>recall</td><td>f1-score</td></tr><tr><td>0</td><td>0.977</td><td>0.747</td><td>0.847</td></tr><tr><td>1</td><td>0.221</td><td>0.803</td><td>0.346</td></tr><tr><td>accuracy</td><td>0.752</td><td>0.752</td><td>0.752</td></tr><tr><td>macro avg</td><td>0.599</td><td>0.775</td><td>0.597</td></tr><tr><td>weighted avg</td><td>0.915</td><td>0.752</td><td>0.806</td></tr></table> <p>Table 1b: Logistic Regression - Classification Report</p>		precision	recall	f1-score	0	0.977	0.747	0.847	1	0.221	0.803	0.346	accuracy	0.752	0.752	0.752	macro avg	0.599	0.775	0.597	weighted avg	0.915	0.752	0.806
	precision	recall	f1-score																																														
0	0.97	0.74	0.84																																														
1	0.21	0.77	0.33																																														
accuracy			0.74																																														
macro avg	0.59	0.76	0.59																																														
weighted avg	0.91	0.74	0.80																																														
	precision	recall	f1-score																																														
0	0.977	0.747	0.847																																														
1	0.221	0.803	0.346																																														
accuracy	0.752	0.752	0.752																																														
macro avg	0.599	0.775	0.597																																														
weighted avg	0.915	0.752	0.806																																														
<p>Confusion Matrix - Logistic Regression (Balanced)</p>  <p>Figure 1a: Logistic Regression Confusion Matrix</p>	<p>Logistic Regression Confusion Matrix - Gemini</p>  <p>Figure 1b: Logistic Regression Confusion Matrix</p>																																																

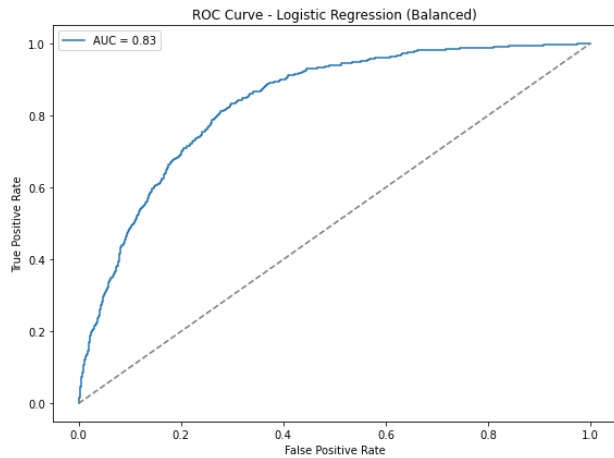


Figure 2a: Logistic Regression ROC Curve Plot

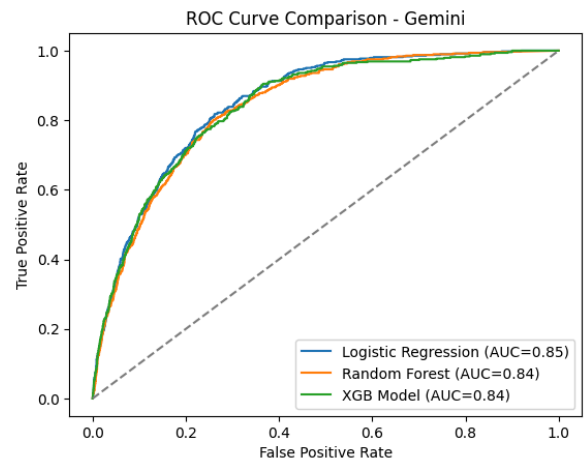


Figure 2b: Logistic Regression ROC Curve Plot

Random Forest

	precision	recall	f1-score
0	0.96	0.79	0.87
1	0.22	0.67	0.34
accuracy			0.78
macro avg	0.59	0.73	0.60
weighted avg	0.90	0.78	0.83

Table 2a: Random Forest - Classification Report

	precision	recall	f1-score
0	0.950	0.911	0.930
1	0.314	0.460	0.373
accuracy	0.874	0.874	0.874
macro avg	0.632	0.685	0.651
weighted avg	0.898	0.874	0.884

Table 2b: Random Forest - Classification Report

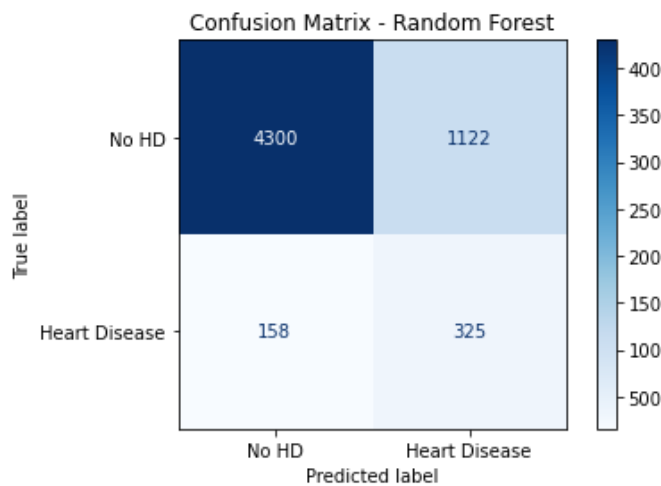


Figure 3a: Random Forest Confusion Matrix

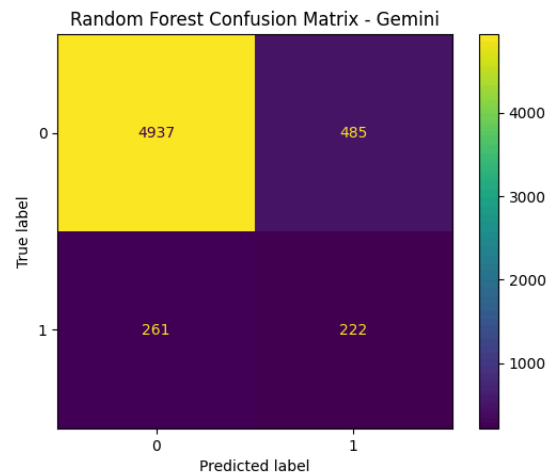


Figure 3b: Random Forest Confusion Matrix

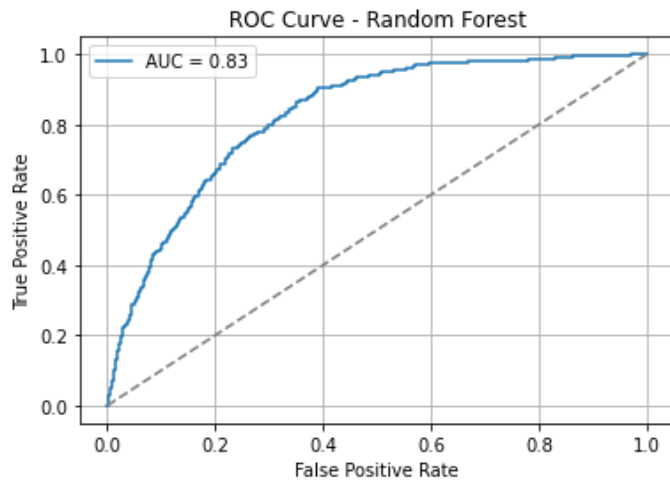


Figure 4a: Random Forest ROC Curve Plot

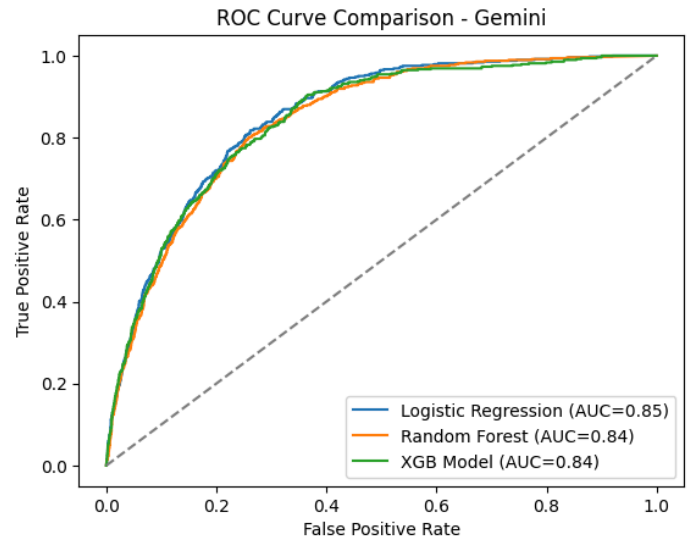


Figure 4b: Random Forest ROC Curve Plot

XGBoost

	precision	recall	f1-score
0	0.97	0.74	0.84
1	0.21	0.79	0.33
accuracy			0.74
macro avg	0.59	0.76	0.58
weighted avg	0.91	0.74	0.80

Table 3a: XGBoost - Classification Report

	precision	recall	f1-score
0	0.972	0.773	0.862
1	0.228	0.754	0.351
accuracy	0.772	0.772	0.772
macro avg	0.600	0.763	0.606
weighted avg	0.912	0.772	0.820

Table 3b: XGBoost - Classification Report

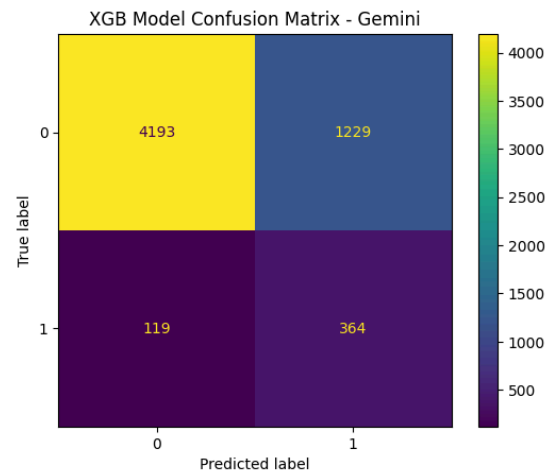
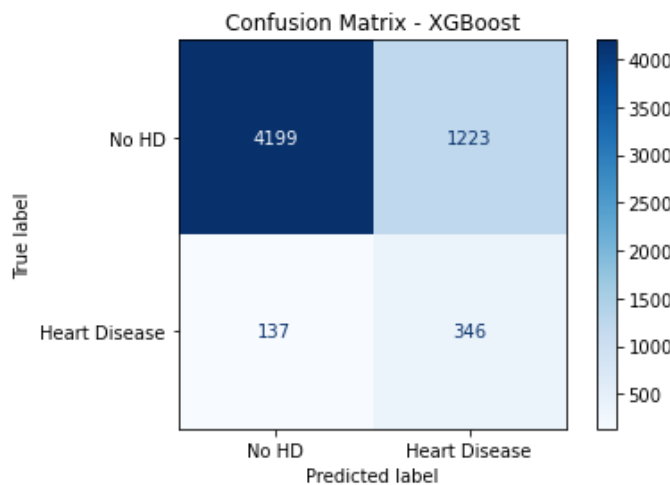


Figure 5a: XGBoost Confusion Matrix

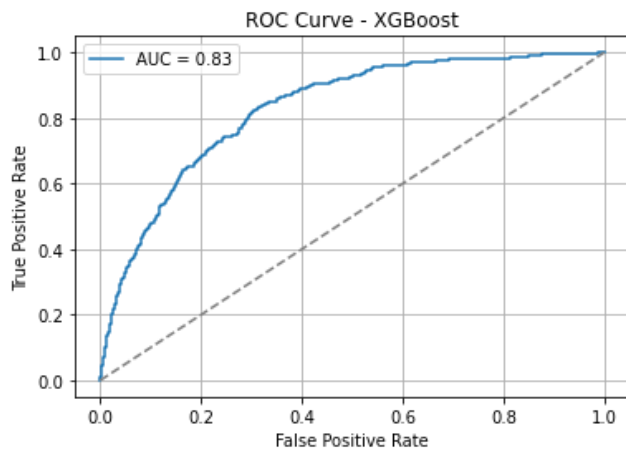


Figure 6a: XGBoost ROC Curve Plot

Figure 5b: XGBoost Confusion Matrix

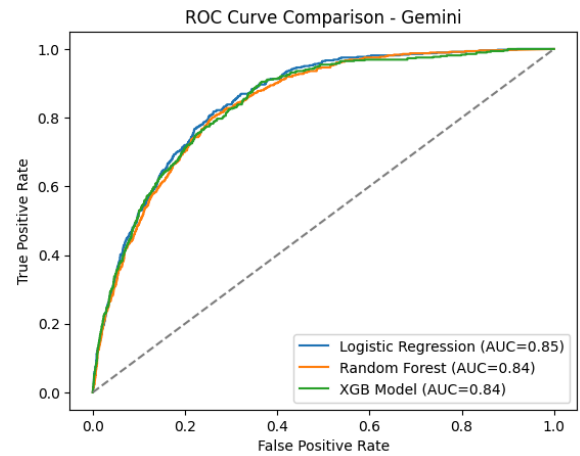


Figure 6b: XGBoost ROC Curve Plot

CatBoost

	precision	recall	f1-score
0	0.96	0.83	0.89
1	0.25	0.64	0.36
accuracy	0.81		
macro avg	0.61	0.73	0.62
weighted avg	0.90	0.81	0.85

Table 4a: CatBoost - Classification Report

	precision	recall	f1-score
0	0.926252	0.989118	0.956654
1	0.486957	0.115942	0.187291
accuracy	0.917697	0.917697	0.917697
macro avg	0.706604	0.552530	0.571972
weighted avg	0.890320	0.917697	0.893724

Table 4b: CatBoost - Classification Report

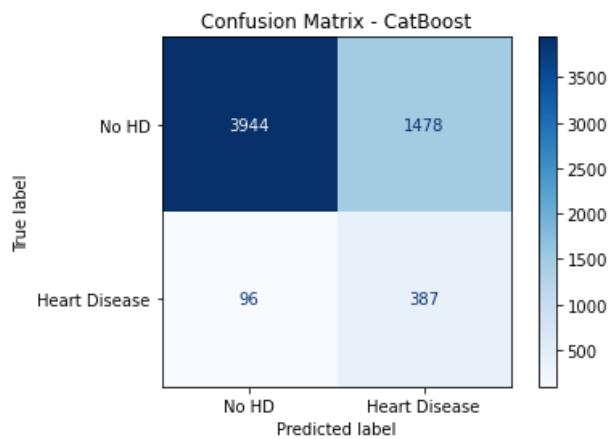


Figure 7a: CatBoost Confusion Matrix

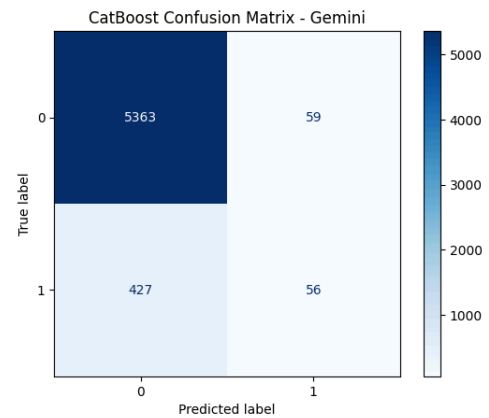


Figure 7b: CatBoost Confusion Matrix

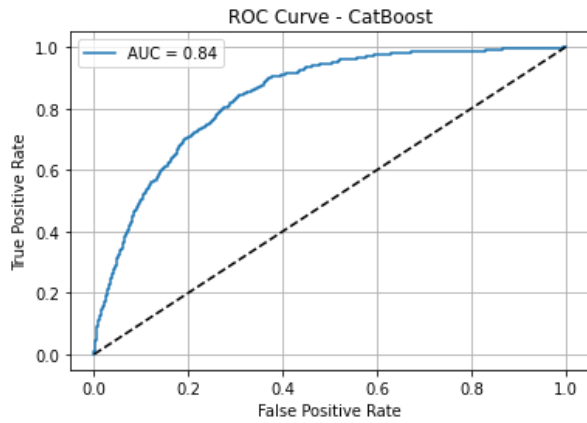


Figure 8a: CatBoost ROC Curve Plot

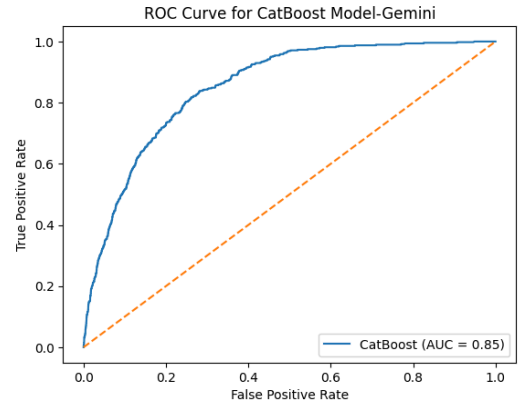


Figure 8b: CatBoost ROC Curve Plot

Gemini Evaluation Metrics

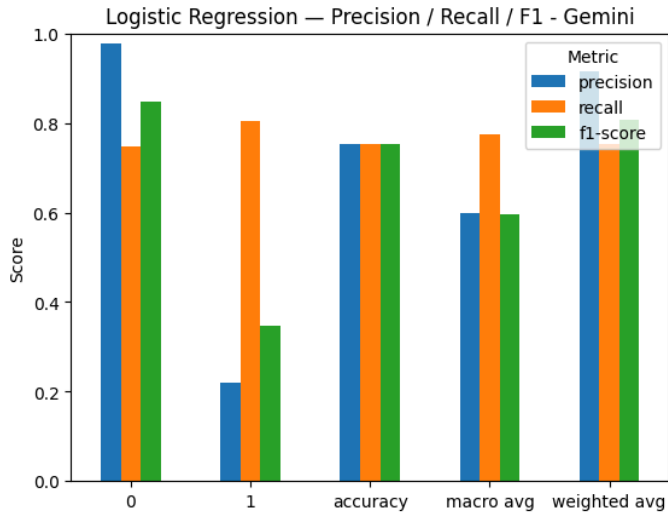


Figure 9a: Logistic Regression Metrics Plot

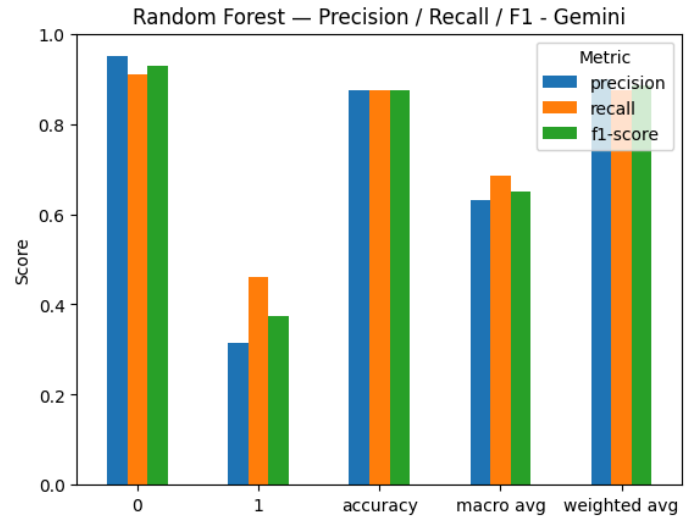
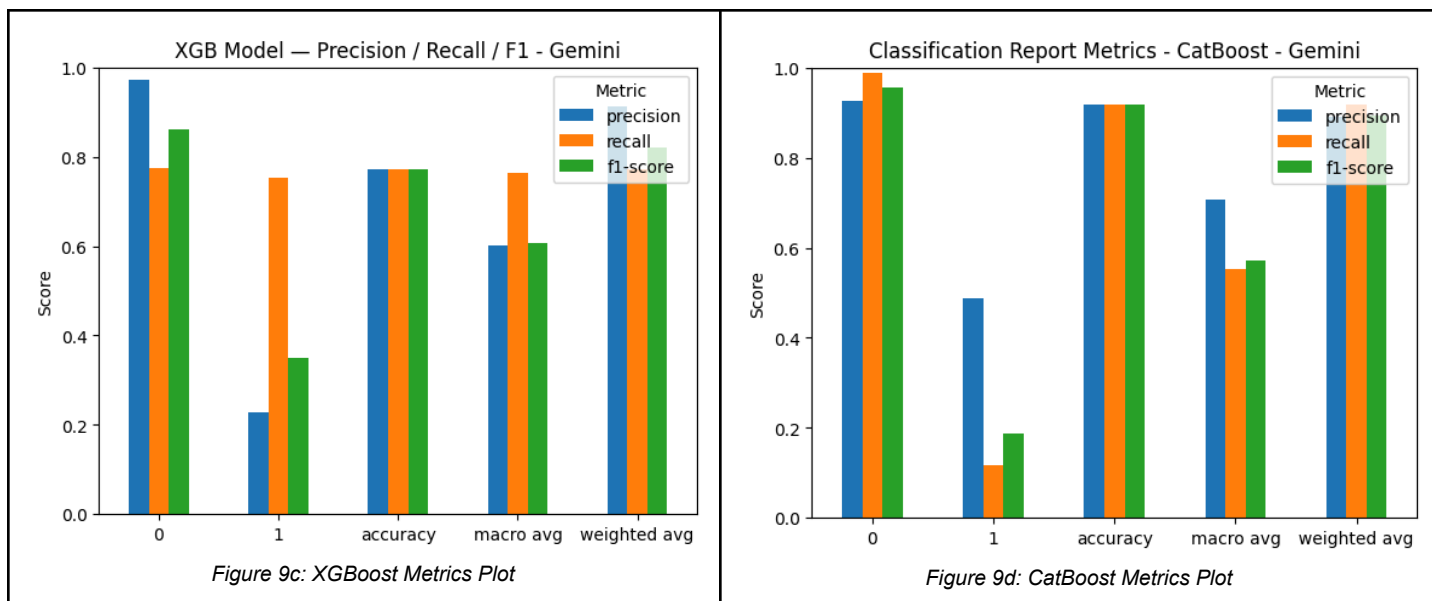


Figure 9b: Random Forest Metrics Plot



Discussion

Logistic Regression:

Our logistic model had a high accuracy of 0.74 and a recall at 0.77. Our earlier models had an accuracy of 0.92 but we tuned the parameters to increase recall. In a real world context this trade off is significant as it reflects the priority of identifying true heart disease cases. It's better to have a higher recall (fewer false negatives) than a higher precision (fewer false positives) due to the real world implications and we did a good job of capturing this important nuance.

Overall, Gemini's selected features had a marginally higher accuracy, macro F1 and weighted F1. It also has a slightly better recall and F1 for the positive class (heart disease), and uses fewer features which reduces model complexity and makes it more interpretable. It's possible that the traditional selected features perform worse due to the extra features which may be adding noise as opposed to useful information and could possibly be leading to overfitting, especially since our data is quite large to begin with. The Gemini selected features are therefore preferred as you would want to catch as many true cases as possible. Gemini's advanced reasoning and multimodal capabilities that it has become famous for allowed it to identify relevant features effectively while simultaneously reducing the feature set without sacrificing predictive accuracy making it a powerful tool for future use in the medical industry.

Gemini's chosen features led to slightly higher total accuracy and F1-scores (macro and weighted), as well as better recall and F1 for the positive class, even though they used fewer features. This made the model simpler, which probably made it easier to understand while lowering the risk of overfitting.

Random Forest:

Using traditionally chosen features to evaluate the Random Forest's performance, the model had a good accuracy of 78% and a pretty high recall of 67% for the minority class (heart disease), with an F1-score of 0.34. It was very good at finding real cases of heart disease, but it had a low level of accuracy (0.22), which means it produced a lot of false positives. In health care, where missing a diagnosis can have very serious consequences, this trade-off is usually seen as fair. It may be more work in the short term to flag healthy people for extra screening, but that is usually better than not finding patients who are really at risk. Usually, statistical relevance guides feature selection, and this recall-focused method seems to support it by putting sensitivity over specificity.

The Random Forest model using Gemini-selected features achieved higher overall accuracy (87.4%) and performed well on the majority class (F1-score of 0.93). However, its recall for the minority class (heart disease) was lower at 46%, with an F1-score of 0.37—slightly better than the traditional feature selection model, but it missed more true positives. This reflects a trade-off between precision and sensitivity. Gemini's smaller feature set appears optimized for overall performance and ease of use, making it ideal. In contrast, the traditional feature set had a higher recall (0.67) but at the cost of very low precision (0.22), resulting in many false positives. While this could lead to unnecessary follow-ups, increased anxiety, etc. it's often a small price to pay in the field of health care. It is much more preferable to have false positives than false negatives especially in relation to the medical industry where early treatment can mean the difference between life and death. Ultimately, the Gemini model offers better balance and higher accuracy, making it the preferable choice for detecting heart disease even if it has a lower recall rate and more false positives.

CatBoost:

When we used the CatBoost model to compare traditional and Gemini 2.5 Pro-selected features, we found that the performance is very different. With traditional feature selection, the model got an F1-score of 0.36 for the positive class (heart disease) and a total success rate of 81%. When Gemini-selected traits were used, on the other hand, the model was more accurate overall (91.8%), but it was much less effective at finding positive cases—its F1-score for the heart disease class was only 0.19. Gemini's chosen traits made the model better at finding the majority class, but this meant that it found fewer real heart disease cases.

This shows an important issue with health-related classification tasks: total accuracy can be wrong when there is an imbalance between classes. In real healthcare, not finding patients who are at risk could have major effects, so recall and F1-score for the minority class are often more important than overall accuracy. Still, Gemini's suggestion of a feature subset that produced high performance on the majority class, without any statistical testing or domain-specific tuning, shows that LLMs could be useful for exploring data early on, especially when making prototypes or working with new datasets.

XGBoost:

It was very interesting to see how similar the XGBoost model's success was across both traditional and Gemini 2.5 Pro-selected features. The model got an F1-score of 0.34 for the positive class (heart disease) and a 77% total success rate using traditional feature selecting. With the features chosen by Gemini, the model got a slightly higher F1-score of 0.35 for heart disease and the same total accuracy of 77.2%. Both times, recall for the positive class was high (72% for the traditional class and 75.4% for the Gemini class), but accuracy stayed low because of the imbalance between the classes.

Based on these results, it looks like Gemini's feature selection was able to come close to the human-guided statistical selection in the XGBoost setting. Even though the model still had trouble with low accuracy on the minority class, both methods were very good at finding cases of heart disease, which is an important trait for tasks that involve predicting clinical risk. These findings show that LLM-driven feature selection may be a good option to manual statistical methods in some model architectures, especially when speed and automation are important.

In conclusion, Gemini 2.5 Pro selected features worked well and sometimes even did a better job than the traditionally selected features, but there are still some issues. The results may have been changed by variations in hyperparameter tuning, how sensitive they were to class imbalance, and how the features were shown in different processes. Additionally, Gemini was also only asked once with the prompt I gave; more repeated or situation-aware prompting could result in a better output. Standardizing tuning, looking into class imbalance techniques like SMOTE, and using this method on other datasets are all things that should be done in the future. Overall, Gemini shows a lot of promise as a data science assistant—it offers quick and small solutions—but it still needs to be supervised, especially in healthcare settings where fairness and ease of use are very important.

References

- CatBoost. (n.d.). *Missing values processing*. CatBoost.ai. Retrieved Apr 21, 2025, from <https://catboost.ai/docs/en/concepts/algorithm-missing-values-processing>
- Chepenko, D. (2019, Feb 13). *Introduction to gradient boosting on decision trees with Catboost*. Medium. <https://medium.com/data-science/introduction-to-gradient-boosting-on-decision-trees-with-catboost-d511a9ccbd14>
- G, A., Ganesh, B., Ganesh, A., Srinivas, C., Dhanraj, & Mensinkal, K. (2022). Logistic regression technique for prediction of cardiovascular disease. *Global Transitions Proceedings*, 3, 127-130. <https://www.sciencedirect.com/science/article/pii/S2666285X22000449>
- Olteanu, A. (2025, March 26). *Gemini 2.5 Pro: Features, Tests, Access, Benchmarks & More*. DataCamp. Retrieved April 6, 2025, from https://www.datacamp.com/blog/gemini-2-5-pro?utm_source=chatgpt.com
- Sriraman, I. (2024, September 11). *A Deep Dive into XGBoost: How It Works and Its Differences from GBM*. Medium. Retrieved April 21, 2025, from <https://ishwaryasriraman.medium.com/a-deep-dive-into-xgboost-how-it-works-and-its-differences-from-gbm-11b0b01f9714>
- Sruthi. (2025, February 28). *Random Forest Algorithm with Machine Learning*. Analytics Vidhya. Retrieved April 21, 2025, from <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest>
- Sumwiza, K., Twizere, C., Rushinabigwi, G., Bakunzibake, P., & Bamurigire, P. (2023). Enhanced cardiovascular disease prediction model using random forest algorithm. *Informatics in Medicine Unlocked*, 41. <https://www.sciencedirect.com/science/article/pii/S2352914823001624>

Ultralytics. (n.d.). *XGBoost*. Ultralytics Glossary. Retrieved Apr 21, 2025, from <https://www.ultralytics.com/glossary/xgboost>

Statement of Work

Numan Suri	Hamza Asad	Asad Kamal
Author of gemini_catboost_pipeline.ipynb and gemini_lr_rf_gb_pipeline.ipynb. Also worked on the method, results, and conclusion section of the report.	Author of capstone-2.ipynb (initial research, processing and exploration later incorporated in asad_ml_notebook), Discussion section of the report.	Author of asad_ml_notebook.ipynb, trad ML pipeline, results section of report.