IEOR 142 Project - Predicting diabetes based on health factors

Maria-Theresa Licka, Toshani Khanna, Ke Ma, Sinchana Srinivas and Mario Schweikert

Diabetes is a critical public health issue in the United States, affecting millions of individuals annually and resulting in severe health complications and a significant economic burden. Diabetes occurs when the body either fails to produce sufficient insulin or cannot utilize it effectively. This leads to elevated blood sugar levels, which, if unmanaged, may result in complications such as: Heart disease, Vision impairment, Kidney damage and Amputations of lower limbs. While diabetes has no cure, adopting strategies like a healthy diet, physical activity, weight management, and medical intervention can mitigate its effects. Early detection plays a pivotal role in preventing the progression of the disease, making predictive models essential tools for identifying at-risk populations. The **scale of the problem** is substantial:

- **34.2 million Americans** had diabetes in 2018, and **88 million** had prediabetes.
- **1 in 5 diabetics** and **8 in 10 pre diabetics** are unaware of their condition.
- The economic burden is immense, with annual costs estimated at **$327 billion** for diagnosed diabetes and around **$400 billion** when including undiagnosed cases and prediabetes.

The dataset used in this analysis stems from the **Behavioral Risk Factor Surveillance System (BRFSS)**, a telephone survey conducted by the CDC since 1984, capturing over **400,000 health-related responses annually**.
The dataset used is derived from the Behavioral Risk Factor Surveillance System (BRFSS), a health-related telephone survey conducted annually by the CDC. The data includes 253,680 responses from the 2015 survey, with 21 feature variables capturing participants' health behaviors, chronic conditions, and preventative healthcare use. The target variable, **Diabetes_012**, categorizes individuals into three classes: 0 (no diabetes or diabetes only during pregnancy), 1 (prediabetes), and 2 (diabetes). The dataset is imbalanced, with a larger proportion of respondents in the "no diabetes" category. It represents a cleaned subset of the original dataset, which initially contained 441,455 responses and 330 variables

Processing Data:
Binary Model: To simplify the problem and focus on distinguishing between individuals with no diabetes (label 0) and those with diabetes (label 1), we excluded pre-diabetes cases. This approach enables the development of a balanced binary dataset for more accurate classification and aligns with real-world needs where identifying confirmed diabetes is critical.
Multi-Class Model: Retaining pre-diabetes as a separate class (labels 0, 1, and 2) allows for nuanced predictions across all three stages (no diabetes, pre-diabetes, and diabetes). This model provides comprehensive insights for interventions targeting different risk groups.
Handling Dataset Imbalance
The original dataset is imbalanced, with significantly more respondents classified as having no diabetes compared to those with diabetes or pre-diabetes. To address this we tried different approaches like Oversampling or Undersampling, ensuring an equal number of samples in both classes. This prevents bias in model training. For the multi-class model, we retained the imbalanced nature while using stratified splits to maintain proportional representation of all classes during training and testing.
Note: Code File: Initial Data Analysis has some graphs and descriptions to further understand the dataset.

Analytics Model
Our notebook is divided into two sections: one for the binary models and one for the multi-class models. Each section is further divided into Data Preprocessing, Training, and Testing. Since our dataset is highly imbalanced, we tested different approaches to address this issue for each problem. Consequently, every model has its own preprocessing pipeline. To ensure consistency in evaluating the models, we performed a train-test split at the beginning of the notebook and created two new CSV files from the initial dataset. For reproducibility a fixed seed was used during splitting and training.

IEOR 142 Project - Predicting diabetes based on health factors

Maria-Theresa Licka, Toshani Khanna, Ke Ma, Sinchana Srinivas and Mario Schweikert

Binary Models

Logistic Regression: For this model, we undersampled and balanced the dataset to 33.932 data points. After calculating the VIFs, we confirmed that all values were below 5, indicating no multicollinearity issues. We began by training an initial logistic regression model using all features. Iteratively we retrained the model, removing the feature with the highest p-value at each step until all remaining features had statistical significance (p-value smaller than 0.05). Our final logistic regression model only excludes the following features: Veggie, Smoker, NoDocbcCost (no money for doctor) and Physical Activity. The signs of all factors make sense.

Gradient Boosting: Our Gradient Boosting Model was trained using two sampling techniques to handle class imbalance: NearMiss[1], which removes majority class samples farthest from the minority class and SMOTE, which generates synthetic data for the minority class. Hyperparameters for NearMiss were fine-tuned using GridSearchCV. While Oversampling shows a medium TPR (42.5%), the FPR (0.08%) is really low. Undersampling has a high TPR (84.3%) and a high FPR (65.51%).

LDA: Our binary Linear Discriminant Analysis (LDA) models were trained using NearMiss and SMOTE to address class imbalance. With the default threshold (0.5), the baseline model achieved an accuracy of 0.8609, while SMOTE and NearMiss models had TPRs of 0.7779 and 0.7694, respectively, but higher FPRs. After optimizing thresholds using the ROC curve, all models showed improved TPRs (e.g., 0.8296 for baseline) at the cost of increased FPRs, highlighting the trade-offs in balancing sensitivity and specificity. However, threshold optimization did not improve overall accuracy across models.

Cart: This model implements 10-fold cross-validated CART using GridSearchCV and DecisionTreeRegressor. We then set the 0.0005 for ccp_alpha and plot the results to improve the generalization ability of the model. We then use the best output from the cross-validation process moving forward, to reduce the risk of overfitting. The test accuracy of CART model is high (85.95%) but has a low sensitivity (FPR: 16.7%). One strength is that we can explain the model and see that high Blood Pressure is the root node.

Random Forest: This model implements 5-fold cross-validated CART using GridSearchCV and RandomForestClassifier. Then we used CV to find the best max features value (3).The test accuracy of Random Forest is (86.58%),Have good model performance.

PyTorch[2]: Our PyTorch model includes an input layer and a hidden layer (both activated by Softmax), followed by a final layer with a single neuron activated by Sigmoid to output a probability between 0 -1. We used binary cross-entropy loss[3] and the Adam optimizer. After training on the SMOTE oversampled dataset with a learning rate of 0.001 for 15 epochs, we saw that the loss converged. HP tuning was performed manually.

Evaluation Binary Models: To evaluate our binary models, we created a baseline model as a ground truth for comparison. This baseline model always predicts the most common class,which is healthy (0). While calculating accuracy provides an initial understanding of performance, it does not adequately reflect the model's effectiveness on our highly imbalanced dataset. To gain deeper insights, we also calculated the **True Positive Rate** and **False Positive Rate**, visualized classification errors using a **confusion matrix**, and

---

[1]https://www.comet.com/site/blog/resampling-to-properly-handle-imbalanced-datasets-in-machine-learning/
[2]https://machinelearningmastery.com/building-a-binary-classification-model-in-pytorch/
[3]https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a?gi=d1679b2052cc

plotted the **ROC curve**. For models that return probabilities, we manually further evaluated their performance by adjusting thresholds and finding a trade-off between TPR and FPR.

	Model	Accuracy	TPR	FPR
0	Baseline Model	0.978818	NaN	NaN
1	Logistic Regression	0.726360	0.787129	0.283692
2	LDA (Original)	0.699600	0.829600	0.321900
3	LDA (Oversampling)	0.699600	0.809800	0.302900
4	LDA (Undersampling)	0.699600	0.741200	0.345500
5	CART	0.859500	0.167900	0.028200
6	Gradient Boosting (Oversampling)	0.848700	0.425000	0.081200
7	Gradient Boosting (Undersampling)	0.432700	0.843000	0.635100
8	Random Forest	0.865800	0.152800	0.016300
9	Torch	0.829600	0.512700	0.117900

Logistic Regression provides a balanced tradeoff between sensitivity and specificity, making it a strong candidate for identifying diabetic patients while minimizing unnecessary tests. Gradient Boosting (Oversampling) is also a solid option, offering improved detection but with higher complexity. Models like CART and LDA, despite their individual strengths, fall short overall due to low sensitivity (CART) or high false positives (LDA). CART would have a good explainability but fails in reliable recognition.
Given the context of early diabetes intervention, Logistic Regression is recommended for practical use, while Gradient Boosting (Oversampling) can be considered for a more nuanced approach if additional complexity is manageable. Both models help achieve the project goal of effectively predicting diabetes risk and offering valuable insights for targeted healthcare interventions.

Multi Class Models:
Gradient Boosting: SMOTE was used for oversampling and NearMiss for undersampling. During hyperparameter tuning with GridSearch, we used the F1 score as the evaluation metric to better evaluate model performance on the unbalanced dataset. Neither Undersampling nor Undersampling reliably detect all classes.

LDA: We built our multi classification LDA model separating our target variable (diabetes_type) from the features and did feature scaling using StandardScaler. To address class imbalance, we used both oversampling and undersampling (SMOTE/NearMiss). The LDA models were then trained on the baseline, SMOTE-oversampled, and NearMiss-undersampled datasets.

Cart: When using oversampling and undersampling: SMOTE overall performance was better than NearMiss on CART classification model. We achieved an accuracy of 82.60%, an average recall of 70% and a F1-score of 68%

Random Forest: When using oversampling and undersampling, Random Forest has good performance on the SMOTE oversampled dataset. CART and Random Forest models rely on sufficient data to construct accurate segmentation boundaries. Decision trees may construct inaccurate boundaries when a lot class data is NearMIss.

PyTorch[4]: This time, we increased the complexity of our model by adding more hidden layers to capture the difficult differentiation between the three classes. The input and hidden layers are still activated with ReLU, while the unactivated output layer only produces a vector of length 3, from which we select the largest value. The model is optimized using the multiclass loss function CrossEntropyLoss.

Evaluation Multi Class Models: Again a baseline model with the most common class was created. Besides calculating the accuracy and creating a 3x3 confusion matrix we used the classification_report[5] to calculate Precision, Recall, F1-Score and Support for every class.

---

[4]https://machinelearningmastery.com/building-a-binary-classification-model-in-pytorch/
[5]https://scikit-learn.org/1.5/modules/generated/sklearn.metrics.classification_report.html

IEOR 142 Project - Predicting diabetes based on health factors

Maria-Theresa Licka, Toshani Khanna, Ke Ma, Sinchana Srinivas and Mario Schweikert

As shown in the overview of all results, Random Forest (Oversampling) and CART (Oversampling) achieve the highest precision at 66%. However, the recall for both models is also relatively high at 70%, which could result in many false positive alarms and unnecessary hospital visits. We conclude that our models are not sufficient to detect all three classes simultaneously with satisfactory precision and recall. A rolled out model in production should focus on the binary approach to differentiate between diabetes and no diabetes.

	Model	Precision	Recall	F1-Score	Support
0	Baseline Model	0.45	0.39	0.41	10148.0
1	LDA (Original)	0.49	0.41	0.41	10148.0
2	LDA (Oversampling)	0.45	0.52	0.42	10148.0
3	LDA (Undersampling)	0.39	0.42	0.26	10148.0
4	CART (Oversampling)	0.66	0.70	0.68	9962.0
5	CART (Undersampling)	0.55	0.61	0.44	9962.0
6	Gradient Boosting (Oversampling)	0.46	0.46	0.38	10148.0
7	Gradient Boosting (Undersampling)	0.38	0.37	0.26	10148.0
8	Random Forest (Oversampling)	0.66	0.70	0.68	9962.0
9	Random Forest (Undersampling)	0.55	0.61	0.44	9962.0
10	Torch	0.46	0.43	0.44	10148.0

Impact

Diabetes is a rapidly growing health concern affecting millions of individuals in the U.S. and worldwide. This trend highlights the increasing importance of early detection and preventive care. Unfortunately, many people neglect their health by skipping early and regular wellness screenings, often due to cost concerns. This delay in diagnosis can lead to prolonged undetected suffering. Research shows that reducing the cost of preventive care can significantly boost its accessibility and effectiveness[6]. Our ML model addresses this challenge by providing a fast, cost-effective diagnostic solution that potentially requires only a nurse instead of a doctor. This can also address the problem of long waiting times to get doctor appointments. Patients can independently complete a questionnaire (if they smoke, eat healthy), while a nurse performs basic health measurements. The result of the questionnaire and the health parameters are the feature vector which is inputted to our model. We have developed a UI in Python with Tkinter to illustrate how an input mask can look and uploaded the code and screenshot in the Repository. As smart devices are rapidly developing and are capable of measuring our vitals, it is only a matter of time before compact chips are capable of measuring relevant health indicators for our model such as insulin levels[7], enabling continuous health monitoring without seeing a doctor. After mentioning cost reduction and the potential usage of smart devices for continuous screening, we want to raise awareness in underserved rural areas around the world. Our model could be integrated into telemedicine approaches, allowing individuals to input vital signs taken by a layperson into the system for analysis. Considering the growing population in developing countries, our goal is to offer cost-effective, decentralized, and scalable healthcare solutions. During our dataset search, we discovered that there are ethnic differences and varying risk factors for diabetes. It is essential to ensure that our final model represents all ethnic groups fairly and our data is unbiased.To address this, we could use an ensemble learning approach, combining our general model with specialized models trained on data from specific ethnic populations. For example, an initial step could involve training a model on a dataset of females over the age of 21 with Pima Indian heritage, where we already found a dataset[8]. While considering the **benefits**, it is equally important to examine potential risks. One key strength of our ML model is its ability to operate in the digital world. However, entering sensitive patient data in digital forms carries the **risk** of data breaches and data privacy is a critical concern for medical software. Additionally, our model is not perfect and has a false negative rate greater than zero. This means it may misclassify an unhealthy human as healthy, potentially delaying necessary medical steps and leading to long-term health complications.

Also we have to bridge the gap between just and vision to have an impact on diabetes detection and having an impact on the real world. FDA approvals are a long-term process and cost a lot of money. To overcome this challenge we have to convince different stakeholders and potential investors of our idea.

---

[6] https://www.michiganmedicine.org/health-lab/what-happens-when-preventive-care-becomes-free-patients
[7] https://www.healthline.com/diabetesmine/wrist-smartwatches-and-diabetes-tech
[8] https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset

Appendix

Code:

We have created a GitHub repository consisting of

- Data Analysis Notebook (.ipnyp + .pdf)
- Machine Learning Model Notebook (.ipnyp + .pdf)
- UI Mockup (Screenshot + .py)

The link is: https://github.com/MareSeestern/Indeng142A_Project

Alternative Colab Versions:

- Data Analysis: https://colab.research.google.com/drive/1MTVT76DyKEtfjFVvdo1PtGywtA3vSiI7?usp=sharing
- Model: https://colab.research.google.com/drive/1omtDPKvPzFcyHI8jH7c2ckIXZRZnqFEt?usp=sharing

Additionally, all notebooks can be found under this Appendix as PDF print: