Diabetes Prediction Project Report

1. Problem Statement

Diabetes affects millions, and early detection can make a huge difference in managing the condition. In this project, I set out to build a machine learning model that can predict whether a patient is likely to be diabetic. Using the Pima Indians Diabetes dataset, which includes key measurements like Glucose, Blood Pressure, BMI, Insulin, and Age, the goal was to create a tool that could help healthcare professionals flag potential risks early on.

2. Data Preprocessing & Insights

What I Did with the Data

I started by cleaning the dataset. I checked for missing values and handled them by filling in with the mean or median, so no data was left hanging. Outliers were spotted using box plots and histograms, and then I capped them using a clipping method to avoid skewing the analysis.

What I Found

- **Data Patterns:**

The data showed clear differences between diabetic and non-diabetic patients, especially when looking at variables like Glucose and BMI. These features displayed distinct distributions that hinted at their importance.

- **Feature Relationships:**

A correlation analysis helped me understand which variables were closely linked. This was important because some features were too similar, so I removed the redundant ones to keep the model focused on what really matters.

- **The Chosen Features:**

After testing various methods, I selected Glucose, Blood Pressure, BMI, Insulin, and Age as the top predictors. This decision wasn't just about numbers—it also aligned with what we already know about diabetes risk factors.

3. Model Training & Evaluation

Building the Models

I decided to use two different models:

- **Logistic Regression:** A simple, easy-to-understand model that works well as a starting point.

- **Random Forest:** An ensemble method that's great at handling complex, non-linear relationships in data.

Both models were fine-tuned using grid search to get the best settings for each one. For example, with Logistic Regression, I adjusted the regularization strength and penalty type. For Random Forest, I played with the number of trees, tree depth, and how many samples were needed to split a node.

How They Performed

- **Logistic Regression:**

Achieved an accuracy of around 77%. It struck a good balance by catching most of the true diabetic cases while keeping false positives at a reasonable level.

- **Random Forest:**

Came in with roughly 76% accuracy. Its performance was quite similar, though there were some differences in how it handled precision and recall.

I used metrics like precision, recall, and the F1-score, along with confusion matrices and precision-recall curves, to really understand the strengths and weaknesses of each model. In the end, Logistic Regression seemed to edge out slightly in recall, which is crucial for ensuring that no diabetic cases are missed.

4. Challenges & Future Improvements

What Was Tricky

- **Data Imbalance:**

The number of diabetic cases was lower than non-diabetic ones, making it a bit challenging to ensure the model wasn't biased toward the majority.

- **Feature Selection:**

Stripping away too many features risked losing valuable insights, so it was a balancing act between simplifying the model and keeping the important details.

- **Interpretability vs. Complexity:**

While more complex models like Random Forest can capture more nuance, they're also harder to explain. Keeping the model interpretable was important, especially for something as critical as healthcare.

What Could Be Better

- **More Advanced Techniques:**

In the future, I might try more sophisticated ensemble methods or gradient boosting to see if I can get even better performance.

- **Addressing Imbalance:**

Techniques like SMOTE or tweaking class weights could further improve the recall without compromising precision too much.

- **Explainable AI:**

Adding tools like SHAP or LIME could help break down how each feature contributes to the prediction, which would be very useful for building trust in the model.

- **User Interface Enhancements:**

While the basic Streamlit app works well, making it more user-friendly and integrating feedback from healthcare professionals would be great next steps.

5. Conclusion

In summary, this project successfully built a machine learning model to predict diabetes risk using just five key features. The approach involved careful data cleaning, smart feature selection, and testing with two different models. The final model, especially the Logistic Regression version, shows promising performance and has been deployed as a simple Streamlit app for real-time predictions.

The journey wasn't without challenges, but each hurdle taught valuable lessons that pave the way for future improvements. I'm excited about the potential of this work to contribute to early diabetes detection and hope to build on it even further.

Methodology and Snapshots

Data Preprocessing

- **Data Cleaning:** Missing values were identified and imputed using mean/median values. Outliers were detected via box plots and histograms and then capped using the IQR method.
- **Feature Selection:** A correlation analysis helped remove redundant features. SelectKBest with mutual information was applied to select the top five features, aligning both with statistical significance and clinical relevance.
- **Final Feature Set:** The model was trained using the following features:
  - Glucose
  - Blood Pressure
  - BMI
  - Insulin
  - Age

Model Development

Two models were built and compared:

- **Logistic Regression:** Chosen for its simplicity and interpretability. Hyperparameters (e.g., regularization strength and penalty type) were tuned using grid search.
- **Random Forest Classifier:** An ensemble method used to capture non-linear relationships. Hyperparameters like the number of trees, max depth, and minimum samples split were optimized through grid search.

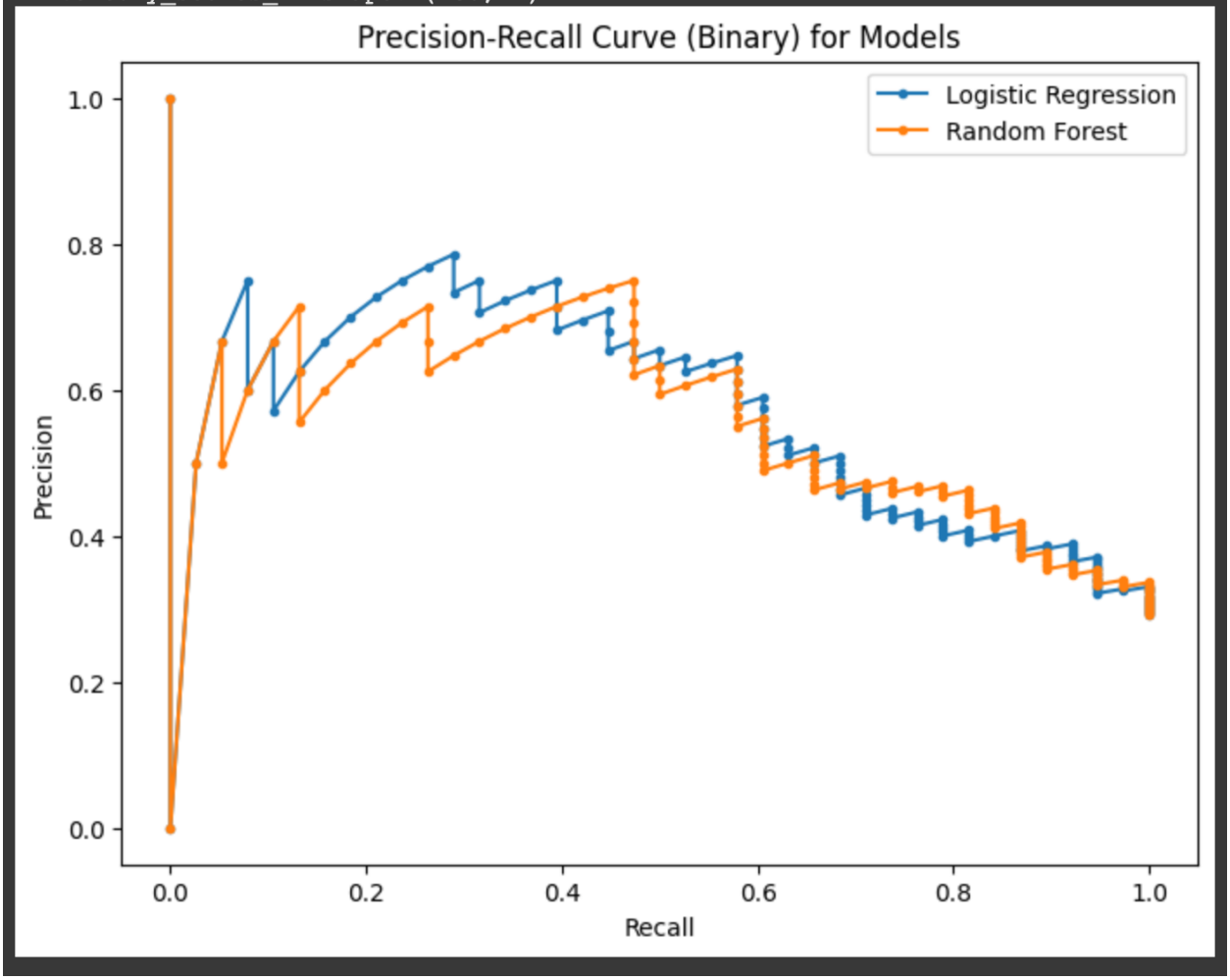
Model Evaluation

- **Evaluation Metrics:** Accuracy, Precision, Recall, and F1-Score were computed. Confusion matrices were generated to understand model errors.
- **Visualization:** Precision-Recall curves were plotted to analyze the trade-offs between precision and recall, especially critical in healthcare settings where false negatives are particularly problematic.

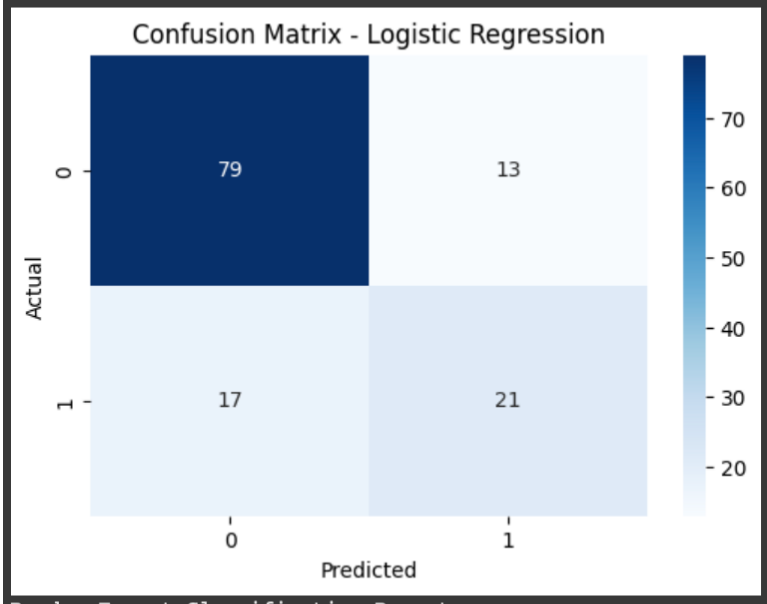
Snapshots

```
Logistic Regression Evaluation:
Accuracy: 0.7769230769230769
Confusion Matrix:
[[79 13]
 [16 22]]
Classification Report:
              precision    recall  f1-score   support

         0.0       0.83      0.86      0.84        92
         1.0       0.63      0.58      0.60        38

    accuracy                           0.78       130
   macro avg       0.73      0.72      0.72       130
weighted avg       0.77      0.78      0.77       130


Random Forest Evaluation:
Accuracy: 0.7615384615384615
Confusion Matrix:
[[81 11]
 [20 18]]
Classification Report:
              precision    recall  f1-score   support

         0.0       0.80      0.88      0.84        92
         1.0       0.62      0.47      0.54        38

    accuracy                           0.76       130
   macro avg       0.71      0.68      0.69       130
weighted avg       0.75      0.76      0.75       130
```

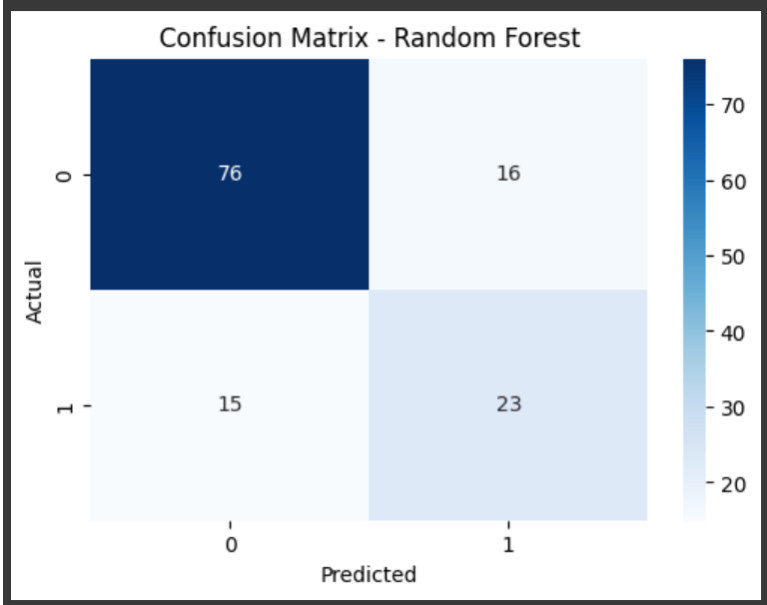
```
Filtered y_scores_log shape: (130, 2)
Filtered y_scores_rf shape: (130, 2)
```



```
Logistic Regression Classification Report:
              precision    recall  f1-score   support

         0.0       0.82      0.86      0.84        92
         1.0       0.62      0.55      0.58        38

    accuracy                           0.77       130
   macro avg       0.72      0.71      0.71       130
weighted avg       0.76      0.77      0.77       130
```

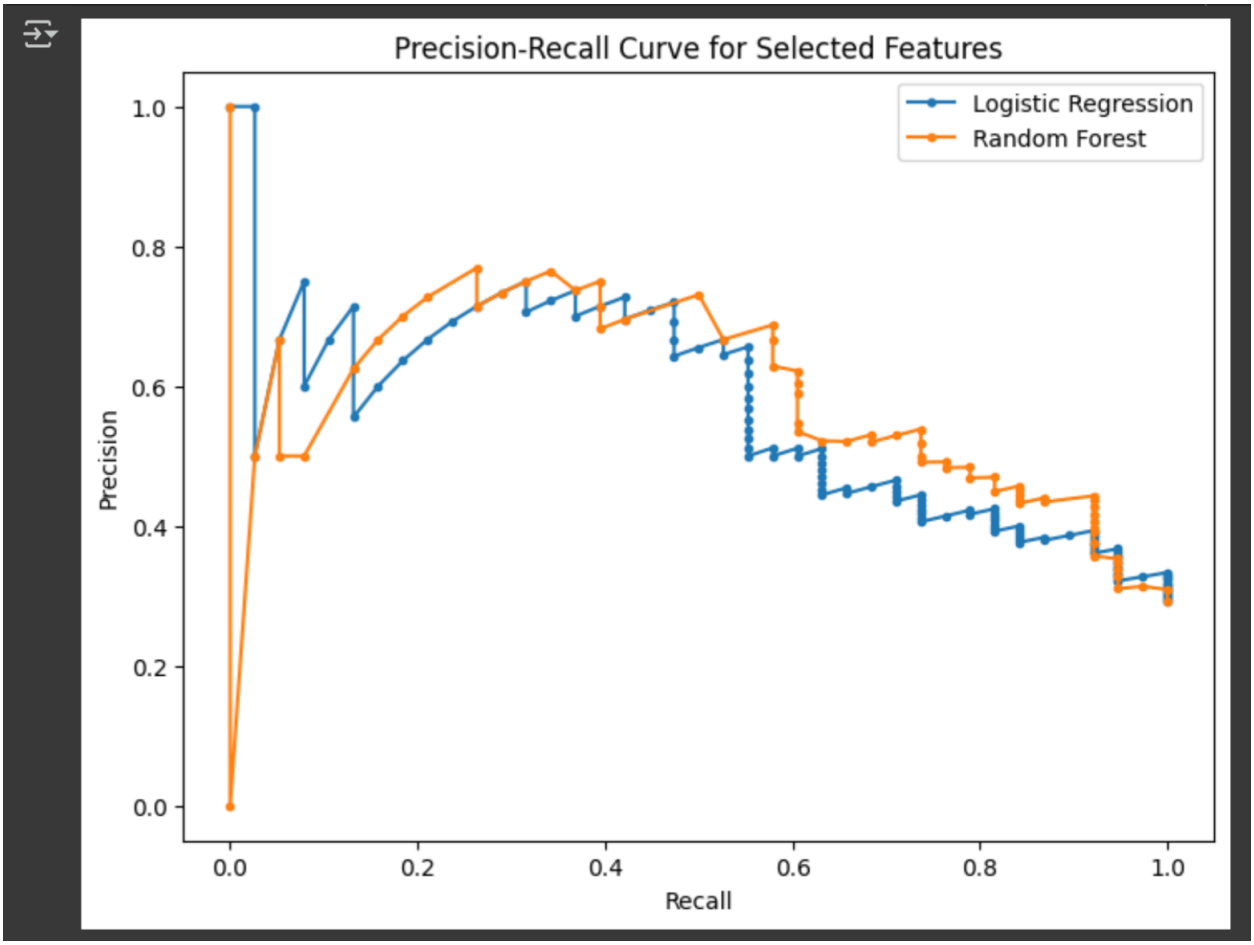



```
Random Forest Classification Report:
              precision    recall  f1-score   support

         0.0       0.84      0.83      0.83        92
         1.0       0.59      0.61      0.60        38

    accuracy                           0.76       130
   macro avg       0.71      0.72      0.71       130
weighted avg       0.76      0.76      0.76       130
```

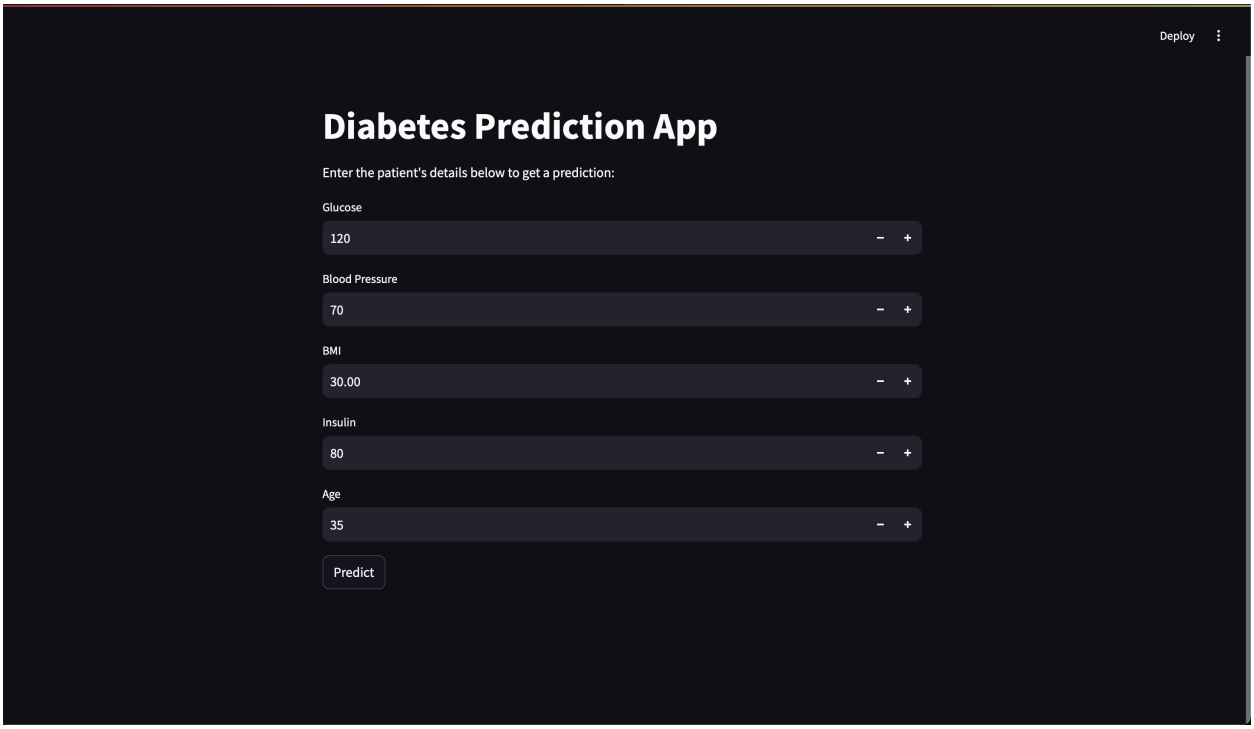


Final eval after re train with selected features

```
Logistic Regression with selected features:
Accuracy: 0.7692307692307693
Classification Report:
              precision    recall  f1-score   support

         0.0       0.82      0.86      0.84        92
         1.0       0.62      0.55      0.58        38

    accuracy                           0.77       130
   macro avg       0.72      0.71      0.71       130
weighted avg       0.76      0.77      0.77       130

Random Forest with selected features:
Accuracy: 0.7615384615384615
Classification Report:
              precision    recall  f1-score   support

         0.0       0.84      0.83      0.83        92
         1.0       0.59      0.61      0.60        38

    accuracy                           0.76       130
   macro avg       0.71      0.72      0.71       130
weighted avg       0.76      0.76      0.76       130
```


Precision-Recall Curve for Selected Features
Logistic Regression
Random Forest
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Working of stream-lit app



Diabetes Prediction App

Enter the patient's details below to get a prediction:

Glucose

110 − +

Blood Pressure

85 − +

BMI

33.60 − +

Insulin

90 − +

Age

50 − +

Predict

Prediction: The patient is not diabetic.

Diabetes Prediction App

Enter the patient's details below to get a prediction:

Glucose

156 − +

Blood Pressure

92 − +

BMI

34.70 − +

Insulin

90 − +

Age

34 − +

Predict

Prediction: The patient is likely diabetic.

Discussion on Model Performance as per Confusion Matrix

The confusion matrices for both models provide insights into their strengths and weaknesses:

- **Logistic Regression:**

The matrix shows that the model correctly identified a high percentage of non-diabetic patients (True Negatives) but had some difficulty with diabetic cases (False Negatives). This is critical because missing diabetic cases could lead to severe consequences.

- **Random Forest:**

While the performance is comparable, the Random Forest model showed slightly different error patterns. It managed a similar balance between True Positives and True Negatives, though minor adjustments may further optimize its sensitivity.

Key Takeaways:

- **False Negatives:**

Both models have room for improvement in reducing false negatives, which is paramount in a healthcare application.

- **Precision-Recall Trade-off:**

The precision-recall curves indicate that while both models are reliable, further tuning (or combining models through ensemble methods) might improve performance, especially in capturing diabetic cases more accurately.

- **Future Focus:**

Addressing class imbalance through techniques like SMOTE, further feature engineering, and exploring advanced ensemble methods could enhance the model's ability to correctly classify diabetic patients.

GitHub Repository

All project files, including the Jupyter Notebook, Python scripts, and the Streamlit app code, are available on my GitHub repository:

https://github.com/shah-khush/Ds_assignment.git