DIABETES READMISSION PREDICTION

1. Summary of Problem Statement, Data, and Findings

The objective of this project was to predict hospital readmission for diabetic patients, using structured patient encounter data which has around 1 lakh rows and 50 columns extracted from **Diabetes 130 US hospitals for years 1999-2008(UCI repository).**
This helps healthcare providers proactively identify high-risk patients and reduce avoidable readmissions, improving patient outcomes and lowering costs.
The dataset contains patient demographics, hospital visit details, diagnoses, lab procedures, medications, and test results.
After cleaning and processing, multiple classification algorithms were applied, including Random Forest, LightGBM, Gradient Boosting, and a Stacking Classifier.

The Random Forest model achieved 95% test accuracy (Precision: 93%, Recall: 98%) and maintained strong performance in 5-fold CV with 93.29% accuracy, 91.09% precision, and 95.97% recall, indicating excellent generalization.

In the context of diabetes readmission prediction, minimizing Type II errors (false negatives) is critical, as failing to identify a patient at high risk of readmission may lead to preventable health deterioration and higher emergency care costs. The distribution of readmitted vs. not readmitted patients is skewed, which can bias model performance if not addressed.

2. Overview of the Final Process

Data Understanding → Data Preprocessing → Feature Engineering → Baseline Modeling → Advanced Modeling → Ensemble Learning → Model Evaluation → Interpretation

DATA UNDERSTANDING:

- **Prediction goal:** Binary classification — *Readmitted* vs *Not Readmitted* (sometimes a "<30 days" category is used for stricter readmission definition) .Studied readmission data with over 50 features, including categorical and numerical values.

  Feature Categories

  The dataset's columns can be grouped into several major categories:

  A. Demographic Features

  These describe the patient's basic characteristics:

- **Race** – e.g., Caucasian, African-American, Asian, Hispanic, Other
- **Gender** – Male / Female
- **Age** – Binned ranges (e.g., [0-10), [10-20), … [90-100)), representing approximate age groups without exact birthdates for privacy
- **Weight** – Patient body weight (often has many missing values)

  B. Encounter & Admission Details

  These describe the context of the hospital visit:

- **Admission type** – e.g., Emergency, Urgent, Elective, Newborn, Not Available
- **Discharge disposition** – e.g., Discharged to home, Transferred, Expired, Hospice
- **Admission source** – e.g., Physician referral, Emergency Room, Transfer from another facility
- **Time in hospital** – Length of stay in days
- **Number of previous inpatient, outpatient, and emergency visits** – Indicates historical healthcare utilization

C. Medical History & Diagnoses

- **Primary, Secondary, and Tertiary diagnoses** – ICD-9 codes representing main reasons for admission
- **Number of diagnoses** – Count of all recorded diagnoses during the stay

D. Laboratory & Test Results

- **Number of lab procedures** – Total number of lab tests performed during the encounter
- **Number of procedures** – Total medical procedures performed (excluding labs)
- **Number of medications** – Unique medications prescribed during the encounter

E. Diabetes-Specific Features

- **Diabetes medications** – Whether the patient is on diabetes medication
- **Change in medications** – Indicates whether the patient's diabetes medication regimen was changed during the hospital stay
- **Insulin usage** – Categorized as "No," "Steady," "Up," or "Down" depending on dosage changes

F. Administrative & Payment Details

- **Payer code** – Type of health insurance or payer (often has many missing values)

1.3 Target Variable

The target variable **readmitted** indicates:

- **No or >30** – Not readmitted within a given time frame or Readmitted after 30 days
- **<30** – Readmitted within 30 days (often considered the most critical outcome for hospital penalties)

Converted into binary classification (Readmitted vs Not Readmitted) for simplicity.

PREPROCESSING & FEATURE ENGINEERING:

- Handled missing values (especially in weight, payer_code, medical_specialty).

-Dropping insignificant like IDs, or columns with heavy null values.

-Performed datatype conversion, simple and logical imputation for categorical variables before encoding. Mapped ICD-9 diagnosis codes to broader disease categories for better interpretability.

- Encoded categorical variables using label encoding and one-hot encoding.

- Normalized numerical features and performed outlier detection and removal using IQR.

-Feature Engineering: Derived new features such as service_utilization (sum of outpatient, emergency, inpatient visits),target also has been converted to binary output.

-Performed univariate, bivariate and statical analysis to understand relationship between features.

-Used resample function to treat the heavy class imbalance in the target by oversampling the minority class.

MODELLING:
-**Baseline Models:** Logistic Regression, Random Forest, Decision Tree, Gradient Boosting.

Advanced Ensemble: Stacking Classifier combining AdaBoost + XGBoost.

Other Models: XGBoost, K-Nearest Neighbors (KNN), AdaBoost, CatBoost, LightGBM, Bagging Classifier, Tuned LightGBM, Voting Classifier (Random Forest + Gradient Boosting).

Evaluation: Accuracy, F1-score, Recall, Precision,ROC AUC, FP%, FN%.

Performed KFold cross-validation for Random Forest and LightGBM to verify good performance and robustness. Performed tuning and RFE to improve scores, but it did not contribute to the model.

3. Step-by-Step Walkthrough of the Solution

1. Data Loading & Exploration – Inspected distribution of readmission labels, feature distributions, and correlation matrix.
2. Missing Value Treatment – Dropped high-missing-value columns or imputed them.
3. Categorical Encoding –After required imputations, converted categorical variables into numerical format and applied dummy encoding which made the data increase to 84 columns.
4. Feature Scaling – Applied Standard scaling to continuous features.
5. Understanding data-Performed univariate, bivariate and statical analysis and plotted heatmaps. Overall, data didn't have much high correlations with the target so most features seemed to be equally important.
6. Feature Engineering-Created new columns like service utilization by summing up count of three columns to analyze better.
7. Class Imbalance treatment: Oversampled the minority class for the model to treat both target categories equally.
8. Baseline Modeling – Trained models and used a metrics function to update it with the evaluated scores.
9. Hyperparameter Tuning – Tried RandomizedCV but didn't improve the result much.
10. Stacking Ensemble – Used LightGBM and RandomForest as base learners, Logistic Regression as meta-learner.
11. Evaluation – Achieved high accuracy and ROC AUC, checked confusion matrix for FN and FP rates for Random Forest.

4. Model Evaluation

No	Model Name	Train Accuracy	Test Accuracy	f1-weighted	Recall	ROC AUC	FP %	FN %	Precision
2	Logistic Regression	0.61	0.60	0.59	0.54	0.63	17.19	23.24	0.61
3	RandomForest	0.98	0.95	0.95	0.97	0.99	3.77	1.28	0.93
4	Decision Tree	0.96	0.90	0.90	0.97	0.93	8.12	1.41	0.86
5	GradientBoost	0.60	0.59	0.59	0.56	0.63	18.62	21.90	0.60
6	XGBoost	0.70	0.68	0.68	0.67	0.75	15.76	16.59	0.68
7	KNN	0.90	0.85	0.85	0.99	0.94	14.02	0.53	0.78
8	AdaBoost	0.60	0.59	0.59	0.52	0.61	17.16	23.89	0.60
9	CatBoost	0.73	0.70	0.70	0.73	0.77	16.25	13.75	0.69
10	LighGBM	0.68	0.66	0.66	0.66	0.73	16.68	17.02	0.66
11	Bagging Classifier	0.60	0.60	0.59	0.54	0.63	17.23	23.15	0.61
12	LightGBM Tuned	0.98	0.93	0.93	0.98	0.98	5.66	1.09	0.90
13	Voting Classifier:RF+GB	0.95	0.91	0.91	0.95	0.98	5.90	2.69	0.89
14	Stacking Classifier: Ada+XGboost	0.80	0.76	0.76	0.77	0.84	12.86	11.30	0.75

RESULTS

Final Model: RandomForest Classifier

Objective: Maximize predictive accuracy while keeping FN (missed readmissions) low.

Key Parameters:
- rf1 = RandomForestClassifier(n_estimators=10, max_depth=25, criterion="gini", min_samples_split=10)

Performance:
- Train Accuracy: 98%
- Test Accuracy: 95%
- ROC AUC: 0.99
- FP%: 3-4% - FN%: 1.3%

The model performs well with a high test accuracy and ROC AUC, and very low false negatives — meaning it's effective at identifying patients likely to be readmitted. The small gap between train and test accuracy suggests mild overfitting but still strong generalization.

CROSS-VALIDATION:

It was performed for two models: RandomForest and LightGBM as a safety check to monitor their high performance.

Cross-Validation Results: Before vs After Model Tuning

To ensure the model's performance was not due to overfitting and was generalizable to unseen data, we applied **5-fold Stratified Cross-Validation**. This approach maintains class balance across folds while evaluating model performance.

Metric	Before Tuning (Train-Test Split)	After Cross-Validation (Mean ± Std)
Accuracy	0.9828 (Train) / 0.9506 (Test)	0.9329 ± *std*
Precision	0.93	0.9109 ± *std*
Recall	0.98	0.9597 ± *std*
F1-Score	0.9545	0.9347 ± *std*

Key Observations:

1. **Before Cross-Validation** (simple train-test split):
    - The model achieved **high accuracy** on both train (98.28%) and test sets (95.06%). ○ Precision (93%) and recall (98%) indicated the model was effective in detecting the positive class, with a slightly stronger bias towards recall.
2. **After Cross-Validation**:
    - The mean accuracy across folds was **93.29%**, with small variation (low standard deviation), suggesting the model is consistently performing well across different subsets of the training data.
    - Precision, recall, and F1-score slightly decreased compared to the single train-test split but remained high, indicating stable generalization.

Inference:
The slight drop in metrics after applying cross-validation is expected, as CV provides a more realistic estimate of model performance by testing on multiple different splits. The relatively small gap between the train-test performance and cross-validation results shows that the Random Forest model is **not overfitting significantly** and maintains **robust predictive power**. The results confirm that the tuned parameters yield a model that generalizes well to unseen data.

Model Performance Summary – LightGBM (Tuned)

Initial Single Train/Test Split Results (Optimistic):

- **Accuracy:** 93.25%
- **Precision:** 89.64%
- **Recall:** 97.82%
- **F1-score:** 93.24%
- **ROC AUC:** 98.18%
- **False Positive Rate:** 5.66%
- **False Negative Rate:** 1.09%

These scores were based on a single train/test split, which can lead to over-optimistic results due to potential desirable splits or data leakage.

Cross-Validation Safety Check (Realistic Generalization Performance):

- **Accuracy:** 82.09% ± 0.34%
- **Precision:** 79.42% ± 0.23%
- **Recall:** 86.60% ± 0.56%
- **F1-score:** 82.85% ± 0.36%
- **ROC AUC:** 90.86% ± 0.15%

These results are averaged over 5 StratifiedKFold cross-validation splits, giving a more reliable estimate of how the model will perform on truly unseen data. While the accuracy and recall are lower than the single split, they are more realistic and trustworthy.

Cross-Validation Results: Before vs After – LightGBM (Tuned)

To validate the robustness of the tuned LightGBM model, performance was compared between a **single train/test split** and **5-fold Stratified Cross-Validation**. The latter provides a more reliable estimate by testing the model on multiple balanced splits of the data.

Metric	Single Train/Test Split (Optimistic)	Cross-Validation (Mean ± Std)
Accuracy	93.25%	82.09% ± 0.34%
Precision	89.64%	79.42% ± 0.23%
Recall	97.82%	86.60% ± 0.56%
F1-Score	93.24%	82.85% ± 0.36%
ROC AUC	98.18%	90.86% ± 0.15%

Inference:
While the **single split** results appeared very strong, it slightly overestimated the performance and **cross-validation revealed a more conservative and trustworthy performance estimate**. The drop in metrics is expected, as CV tests the model's ability to generalize to truly unseen data. Despite this, the LightGBM model retains **strong classification ability and balanced performance** across folds, confirming its suitability for deployment with realistic expectations.But further tuning is required in the single split training,so random forest is considered the best model.

5. Comparison to Benchmark

1. **Random Forest(BENCHMARK)** ○ **Train Accuracy:** 98.23%, **Test**

   Accuracy: 94.95%

   - High **recall (97.44%)** and **precision (92.84%)** indicate it correctly identifies most readmitted patients with minimal false negatives (1.28%) and false positives (3.76%).
   - Overall, it is the **best performing model**, balancing accuracy, recall, and precision.

2. **Decision Tree**
    - Good **recall (97.18%)**, but lower test accuracy (90.47%) and higher false positives (8.12%) compared to Random Forest.
    - Performs reasonably well but prone to overfitting.
3. **LightGBM Tuned**
    - High **train accuracy (97.63%)** and strong **recall (97.82%)**, but slightly lower test accuracy (93.25%) than Random Forest.
    - Good choice for ensemble approaches; slightly more false positives (5.66%).
4. **Voting Classifier (RF + GBM)** ○ Combines strengths of Random Forest and Gradient Boosting. ○ Test accuracy 91.41%, recall 94.62%, precision 88.93%. Reliable but slightly behind Random Forest.
5. **Stacking Classifier (AdaBoost + XGBoost)** ○ Test accuracy 75.83%, recall 77.43%, precision 75%.
    - Lower performance than single Random Forest; better suited for scenarios emphasizing ensemble diversity rather than peak performance.
6. **Other Models (Logistic Regression, Gradient Boosting, XGBoost, KNN, AdaBoost, CatBoost, Bagging, LightGBM)** ○ Generally lower test accuracy (<85%) or lower recall, making them less reliable for identifying high-risk readmitted patients. ○ Logistic Regression and Gradient Boosting show high false negatives (>20%), which is risky in healthcare settings.

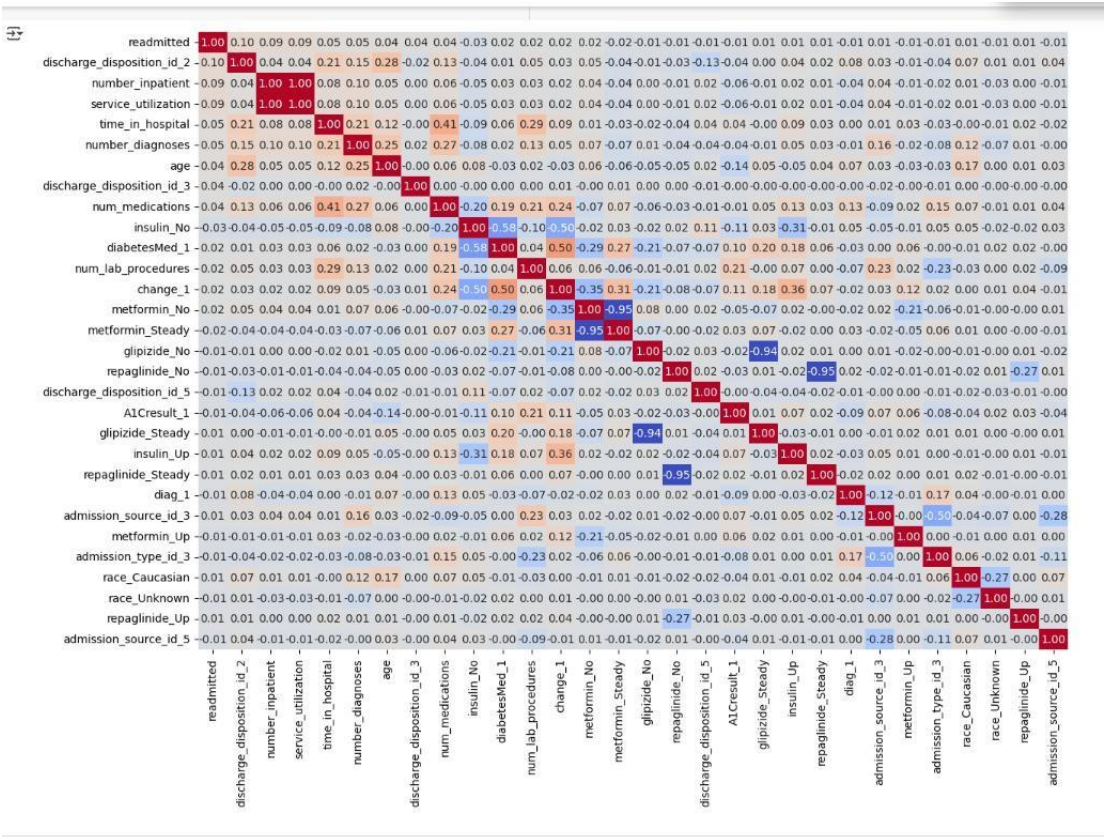
Overall Inference:

- **Random Forest** is the most reliable model for predicting diabetes readmissions based on the current dataset, with **high accuracy, recall, and precision**.
- Ensemble approaches like **Tuned LightGBM** and **Voting Classifier** also perform well but slightly underperform Random Forest.
- Simpler models (Logistic Regression, Gradient Boosting) may miss too many at-risk patients and are less suitable for clinical deployment.

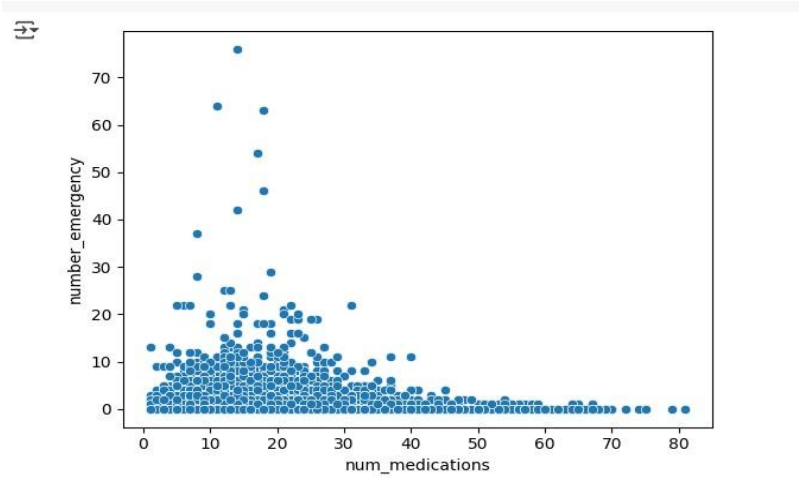
6. Visualizations

Each visualization includes insights and inferences noted during EDA and model evaluation.

Feature correlation heatmap



- Bar plots, scatter plots for EDA-Univariate and Bivariate analysis.

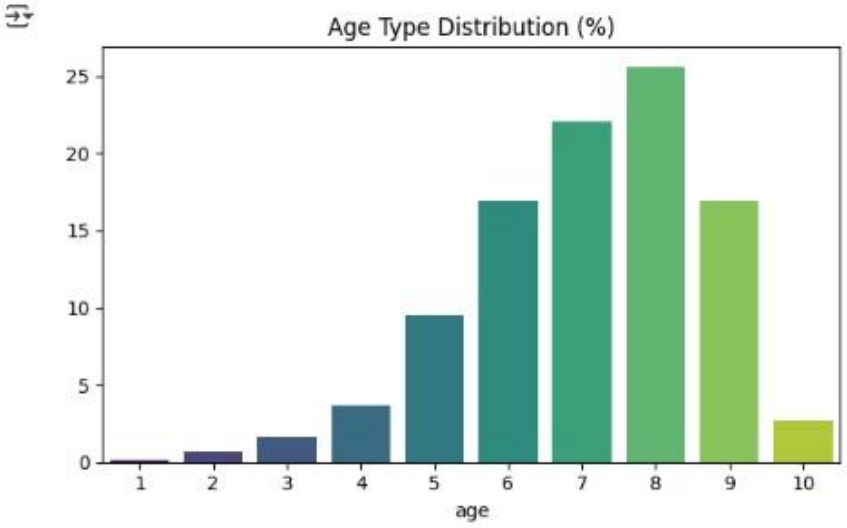



```
df['num_medications'].corr(df['number_emergency'])
```

```
np.float64(0.013179927892209937)
```

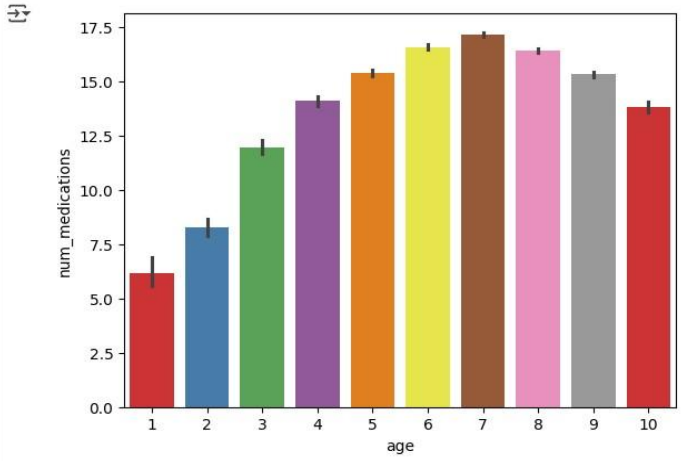
Number of emergency cases have no relation with number of medications they take.

```
ag= df['age'].value_counts(normalize=True) * 100
plt.figure(figsize=(6,4))
sns.barplot(x=ag.index, y=ag.values, palette='viridis')
plt.title('Age Type Distribution (%)')
plt.tight_layout()
plt.show()
```



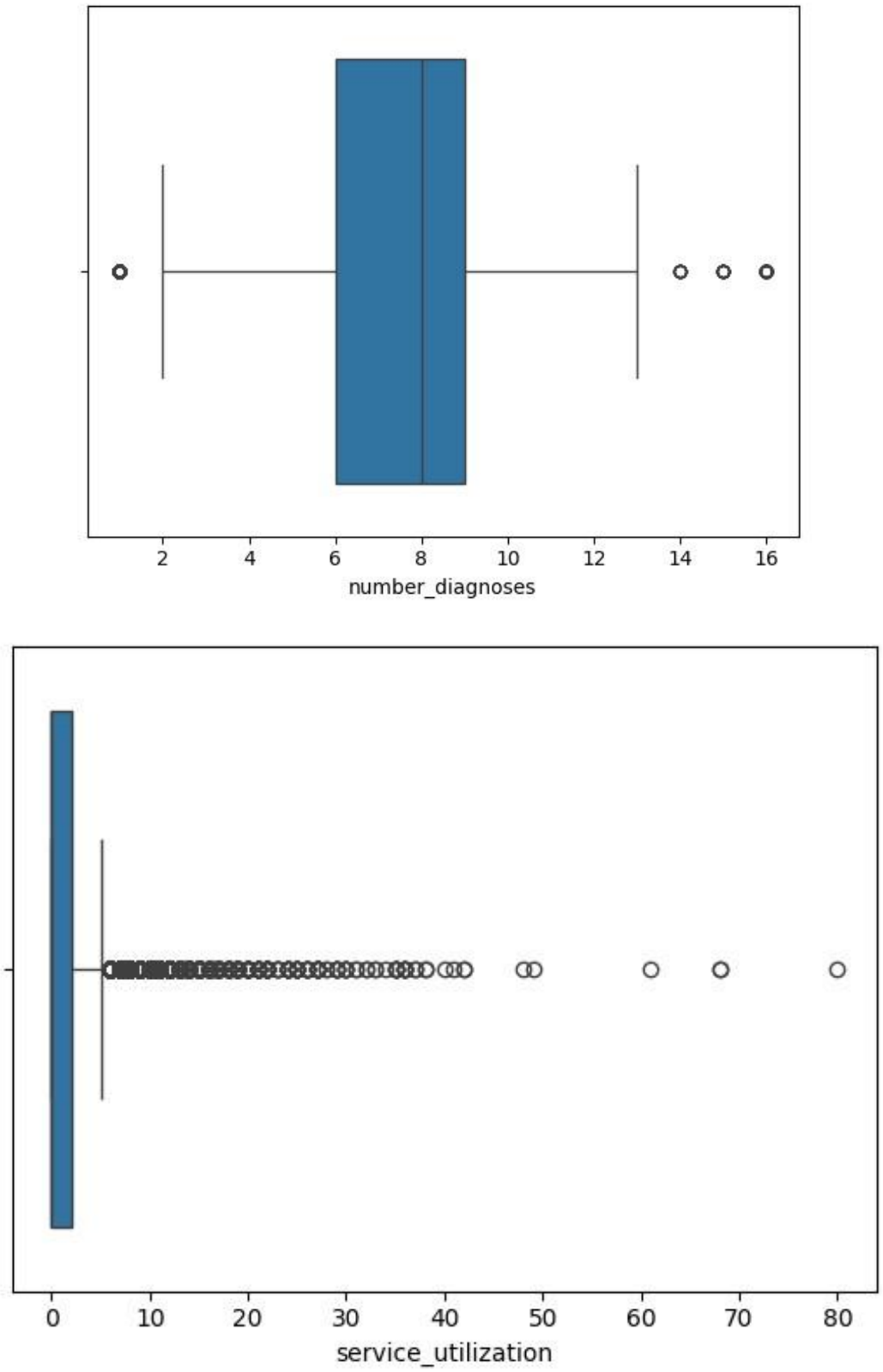
Most admissions for diabetes are aged from 70 to 80 years old.

```
#CAT VS NUM
sns.barplot(data=df,x='age', y='num_medications',palette='Set1')
plt.show()
```

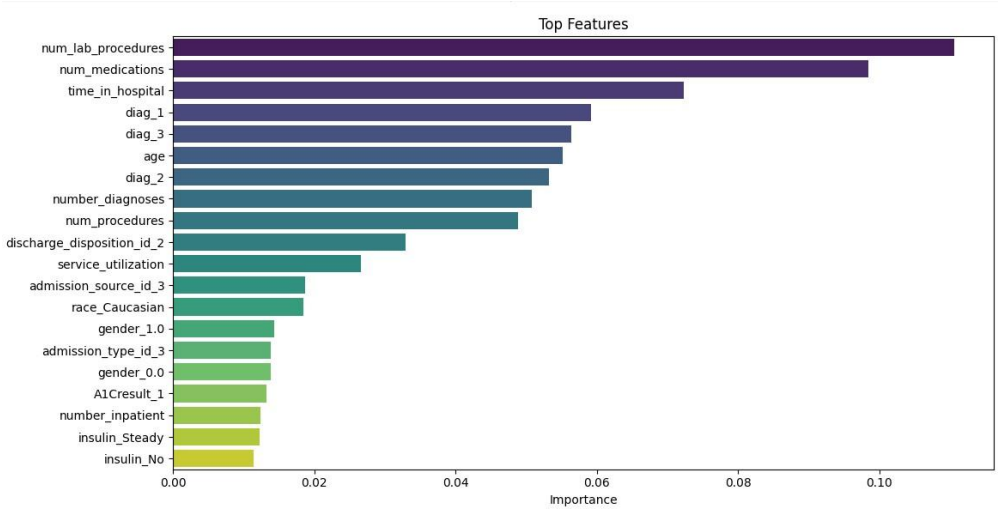


Age 70 to 80 are prescribed most amount of medicines

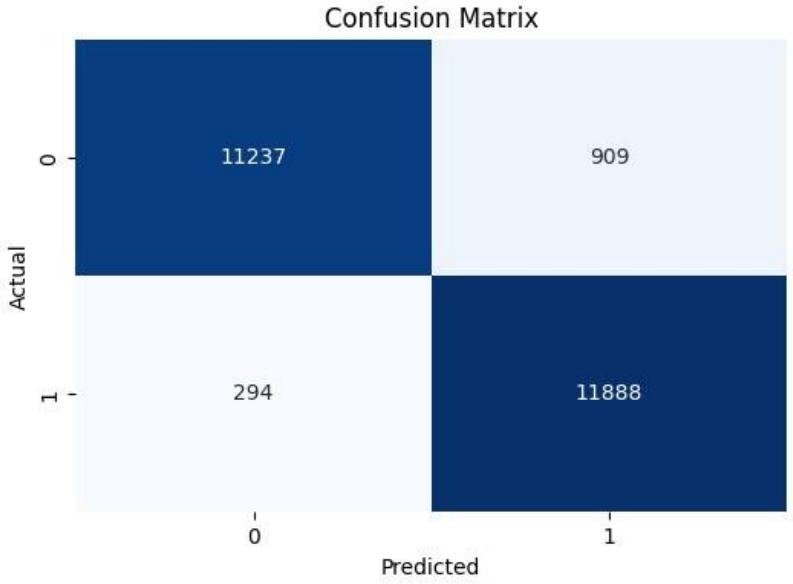
Outliers before treatment



Feature importance plot for RandomForest



- Confusion matrix for final RF model.



Note: other visualizations are performed in the code file.

7. Implications

Impact on Domain/Business:

The Random Forest model, with **cross-validated accuracy of 93.3%**, precision of 91.1%, and recall of 95.97%, provides a highly reliable tool for identifying high-risk diabetic patients. Its strong recall ensures that most patients likely to be readmitted are correctly flagged, which is critical for reducing preventable readmissions and improving patient outcomes.

Recommendations:

- Deploy the Random Forest model in hospital **EHR systems** to support real-time risk assessment.
- Prioritize **post-discharge interventions** for patients flagged as high risk, leveraging the model's high recall to minimize missed cases.
- Use the model in conjunction with **clinical judgment** to guide targeted follow-ups and care plans.
- Continuously monitor and retrain the model as more patient data becomes available to maintain accuracy.

Level of Confidence:
Given the **stable cross-validation performance** (CV scores: 0.9386, 0.9293, 0.9273, 0.9315, 0.9380; mean ± std: 0.9329 ± 0.0046), we can be **highly confident** in the model's ability to generalize across similar patient populations, making it suitable for real-world deployment.

The Random Forest model demonstrates **exceptionally low false positive (3.77%) and false negative (1.28%) rates** compared to other models in the study.This balance of low false positives and false negatives makes Random Forest particularly **safe and reliable for clinical decision support**, providing hospitals with a trustworthy tool to prioritize patients while minimizing risk and resource wastage.
If validated, this model could help hospitals identify high-risk diabetic patients for targeted follow-ups, reducing readmission rates. This could improve patient care and reduce hospital costs.

8. Limitations

- Dataset may not represent all populations or hospital systems.

- Some key medical details (e.g., weight, payer_code) were missing for many patients.

-There is a chance of around 5% that the person who had diabetes fails to be detected based one the recall score.

9. Closing Reflections

The process reinforced the importance of data preprocessing and model validation.
Next time, we would focus on:
- Using nested cross-validation for all models for robust evaluation.
- Adding external validation datasets.
- Trying imbalanced test data to adapt to more real-world examples.
- Performing explainability analysis (e.g., SHAP values) to increase model interpretability in a healthcare context.